PHILIPS

Data handbook

Electronic components and materials

Integrated circuits

Book IC14 1987

Microcontrollers and peripherals

Bipolar, MOS

Elcoma – Philips Electronic Components and Materials Division – embraces a world-wide group of companies operating under the following names:

IBRAPE
Miniwatt
Mullard
PHILIPS
RTC
Signetics
VALVO

Elcoma offers you a technological partnership in developing your systems to the full. A partnership to which we can bring

- world-wide production and marketing
- know-how
- systems approach
- continuity
- broad product line
- fundamental research
- leading technologies
- applications support
- quality

MICROCONTROLLERS AND PERIPHERALS

*

User Manual 1986
"Single-chip 8-bit Microcontrollers"
Available on request

DATA HANDBOOK SYSTEM

Our Data Handbook System comprises more than 60 books with specifications on electronic components, subassemblies and materials. It is made up of four series of handbooks:

ELECTRON TUBES	BLUE
SEMICONDUCTORS	RED
INTEGRATED CIRCUITS	PURPLE
COMPONENTS AND MATERIALS	GREEN

The contents of each series are listed on pages iv to vii.

The data handbooks contain all pertinent data available at the time of publication, and each is revised and reissued periodically.

When ratings or specifications differ from those published in the preceding edition they are indicated with arrows in the page margin. Where application information is given it is advisory and does not form part of the product specification.

Condensed data on the preferred products of Philips Electronic Components and Materials Division is given in our Preferred Type Range catalogue (issued annually).

Information on current Data Handbooks and on how to obtain a subscription for future issues is available from any of the Organizations listed on the back cover.
Product specialists are at your service and enquiries will be answered promptly.

ELECTRON TUBES (BLUE SERIES)

The blue series of data handbooks comprises:

T1 **Tubes for r.f. heating**

T2a **Transmitting tubes for communications, glass types**

T2b **Transmitting tubes for communications, ceramic types**

T3 **Klystrons**

T4 **Magnetrons for microwave heating**

T5 **Cathode-ray tubes**
Instrument tubes, monitor and display tubes, C.R. tubes for special applications

T6 **Geiger-Müller tubes**

T8 **Colour display systems**
Colour TV picture tubes, colour data graphic display tube assemblies, deflection units

T9 **Photo and electron multipliers**

T10 **Plumbicon camera tubes and accessories**

T11 **Microwave semiconductors and components**

T12 **Vidicon and Newvicon camera tubes**

T13 **Image intensifiers and infrared detectors**

T15 **Dry reed switches**

T16 **Monochrome tubes and deflection units**
Black and white TV picture tubes, monochrome data graphic display tubes, deflection units

SEMICONDUCTORS (RED SERIES)

The red series of data handbooks comprises:

S1 **Diodes**
Small-signal silicon diodes, voltage regulator diodes (< 1,5 W), voltage reference diodes, tuner diodes, rectifier diodes

S2a **Power diodes**

S2b **Thyristors and triacs**

S3 **Small-signal transistors**

S4a **Low-frequency power transistors and hybrid modules**

S4b **High-voltage and switching power transistors**

S5 **Field-effect transistors**

S6 **R.F. power transistors and modules**

S7 **Surface mounted semiconductors**

S8a **Light-emitting diodes**

S8b **Devices for optoelectronics**
Optocouplers, photosensitive diodes and transistors, infrared light-emitting diodes and infrared sensitive devices, laser and fibre-optic components

S9 **Power MOS transistors**

S10 **Wideband transistors and wideband hybrid IC modules**

S11 **Microwave transistors**

S12 **Surface acoustic wave devices**

S13 **Semiconductor sensors**

***S14** **Liquid Crystal Displays**

*To be issued shortly.

INTEGRATED CIRCUITS (PURPLE SERIES)

The NEW SERIES of handbooks is now completed. With effect from the publication date of this handbook the "N" in the handbook code number will be deleted.
Handbooks to be replaced during 1986 are shown below.

The purple series of handbooks comprises:

IC01	**Radio, audio and associated systems** Bipolar, MOS	new issue 1986 IC01N 1985
IC02a/b	**Video and associated systems** Bipolar, MOS	new issue 1986 IC02Na/b 1985
IC03	**Integrated circuits for telephony** Bipolar, MOS	new issue 1986 IC03N 1985
IC04	**HE4000B logic family** CMOS	new issue 1986 IC4 1983
IC05N	**HE4000B logic family — uncased ICs** CMOS	published 1984
IC06N	**High-speed CMOS; PC74HC/HCT/HCU** Logic family	published 1986
IC08	**ECL 10K and 100K logic families**	New issue 1986 IC08N 1984
IC09N	**TTL logic series**	published 1986
IC10	**Memories** MOS, TTL, ECL	new issue 1986 IC7 1982
IC11N	**Linear LSI**	published 1985
Supplement to IC11N	**Linear LSI**	published 1986
IC12	**I^2C-bus compatible ICs**	not yet issued
IC13	**Semi-custom** Programmable Logic Devices (PLD)	new issue 1986 IC13N 1985
IC14	**Microcontrollers and peripherals** Bipolar, MOS	published 1986
IC15	**FAST TTL logic series**	new issue 1986 IC15N 1985
IC16	**CMOS integrated circuits for clocks and watches**	first issue 1986
IC17	**Integrated Services Digital Networks (ISDN)**	not yet issued
IC18	**Microprocessors and peripherals**	new issue 1986

COMPONENTS AND MATERIALS (GREEN SERIES)

The green series of data handbooks comprises:

C2 Television tuners, coaxial aerial input assemblies, surface acoustic wave filters

C3 Loudspeakers

C4 Ferroxcube potcores, square cores and cross cores

C5 Ferroxcube for power, audio/video and accelerators

C6 Synchronous motors and gearboxes

C7 Variable capacitors

C8 Variable mains transformers

C9 Piezoelectric quartz devices

C11 Varistors, thermistors and sensors

C12 Potentiometers, encoders and switches

C13 Fixed resistors

C14 Electrolytic and solid capacitors

C15 Ceramic capacitors

C16 Permanent magnet materials

C17 Stepping motors and associated electronics

C18 Direct current motors

C19 Piezoelectric ceramics

C20 Wire-wound components for TVs and monitors

C22 Film capacitors

SELECTION GUIDE

FUNCTIONAL INDEX

Operating temperature range: 0 to 70 °C.
Extended temperature range: −40 to + 85 °C.
Automotive temperature range: −40 to + 110 °C;
except MAF80A31/51AH only −40 to + 100 °C.

Operating temperature range: 0 to 70 °C.
Extended temperature range: –40 to + 85 °C.
Automotive temperature range: –40 to + 110 °C
except MAF80A31/51AH only –40 to + 100 °C.

Operating temperature range: 0 to 70 °C.
Extended temperature range: −40 to + 85 °C.
Automotive temperature range: −40 to + 110 °C.

NUMERICAL INDEX

type number	description			package code	page
SINGLE-CHIP 8-BIT MICROCONTROLLERS					
	RAM	ROM			
MAB8031AHP	128	–	ROM-less version of MAB8051AH	DIL-40; SOT-129	31
MAB8031AHWP	128	–	ROM-less version of MAB8051AH	44-PLCC; SOT-187A	31
MAB8032AHP	256	–	ROM-less version of MAB8052AH	DIL-40; SOT-129	65
MAB8035HLP	64	–	ROM-less version of MAB8048H	DIL-40; SOT-129	97
MAB8035HLWP	64	–	ROM-less version of MAB8048H	44-PLCC; SOT-187A	97
MAB8039HLP	128	–	ROM-less version of MAB8049H	DIL-40; SOT-129	97
MAB8039HLWP	128	–	ROM-less version of MAB8049H	44-PLCC; SOT-187A	97
MAB8040HLP	256	–	ROM-less version of MAB8050H	DIL-40; SOT-129	97
MAB8040HLWP	256	–	ROM-less version of MAB8050H	44-PLCC; SOT-187A	97
MAB8048HP	64	1K		DIL-40; SOT-129	97
MAB8048HWP	64	1K		44-PLCC; SOT-187A	97
MAB8049HP	128	2K		DIL-40; SOT-129	97
MAB8049HWP	128	2K		44-PLCC; SOT-187A	97
MAB8050HP	256	4K		DIL-40; SOT-129	97
MAB8050HWP	256	4K		44-PLCC; SOT-187A	97
MAB8051AHP	128	4K	mask-programmable ROM	DIL-40; SOT-129	31
MAB8051AHWP	128	4K	mask-programmable ROM	44-PLCC; SOT-187A	31
MAB8052AHP	256	8K	mask-programmable ROM	DIL-40; SOT-129	65
MAB8401B	128	–	bond-out version for MAB84X1 family plus 8-bit LED-driver	28/28 Piggy-back	123
MAB8401WP	128	–	bond-out version for MAB84X1 family plus 8-bit LED-driver	68-PLCC; SOT-188A	123
MAB8411P	64	1K	plus 8-bit LED driver	DIL-28; SOT-117	123
MAB8411T	64	1K	plus 8-bit LED driver	SO-28; SOT-136A	123
MAB8421P	64	2K	plus 8-bit LED driver	DIL-28; SOT-117	123
MAB8421T	64	2K	plus 8-bit LED driver	SO-28; SOT-136A	123
MAB8422P	64	2K	plus 8-bit LED driver	DIL-20; SOT-146	155
MAB8441P	128	4K	plus 8-bit LED driver	DIL-28; SOT-117	123
MAB8441T	128	4K	plus 8-bit LED driver	SO-28; SOT-136A	123
MAB8442P	128	4K	plus 8-bit LED driver	DIL-20; SOT-146	155
MAB8461P	128	6K	plus 8-bit LED driver	DIL-28; SOT-117	123
MAB8461T	128	6K	plus 8-bit LED driver	SO-28; SOT-136A	123

type number	description			package code	page
	RAM	ROM			
MAF8031AHP	128	–	ROM-less version of MAB8051AH; extended temperature	DIL-40; SOT-129	31
MAF8031AHWP	128	–	ROM-less version of MAB8051AH; extended temperature	44-PLCC; SOT-187A	31
MAF80A31AHP	128	–	ROM-less version of MAB8051AH; automotive temperature; reduced frequency	DIL-40; SOT-129	31
MAF80A31AHWP	128	–	ROM-less version of MAB8051AH; automotive temperature; reduced frequency	44-PLCC; SOT-187A	31
MAF8035HLP	64	–	ROM-less version of MAB8048H; extended temperature	DIL-40; SOT-129	97
MAF80A35HLP	64	–	ROM-less version of MAB8048H; automotive temperature; reduced frequency	DIL-40; SOT-129	97
MAF8039HLP	128	–	ROM-less version of MAB8049H; extended temperature	DIL-40; SOT-129	97
MAF80A39HLP	128	–	ROM-less version of MAB8049H; automotive temperature; reduced frequency	DIL-40; SOT-129	97
MAF8040HLP	256	–	ROM-less version of MAB8050H; extended temperature	DIL-40; SOT-129	97
MAF80A40HLP	256	–	ROM-less version of MAB8050H; automotive temperature; reduced frequency	DIL-40; SOT-129	97
MAF8048HP	64	1K	like MAB8048H; extended temperature	DIL-40; SOT-129	97
MAF80A48HP	64	1K	like MAB8048H; automotive temperature; reduced frequency	DIL-40; SOT-129	97
MAF8049HP	128	2K	like MAB8049H; extended temperature	DIL-40; SOT-129	97
MAF80A49HP	128	2K	like MAB8049H; automotive temperature; reduced frequency	DIL-40; SOT-129	97
MAF8050HP	256	4K	like MAB8050H; extended temperature	DIL-40; SOT-129	97
MAF80A50HP	256	4K	like MAB8050H; automotive temperature; reduced frequency	DIL-40; SOT-129	97

Operating temperature range: 0 to 70 °C.
Extended temperature range: −40 to + 85 °C.
Automotive temperature range: −40 to + 110 °C;
except MAF80A31/51AH only −40 to + 100 °C.

type number	description			package code	page
	RAM	ROM			
MAF8051AHP	128	4K	like MAB8051AH; extended temperature	DIL-40; SOT-129	31
MAF8051AHWP	128	4K	like MAB8051AH; extended temperature	44-PLCC; SOT-187A	31
MAF80A51AHP	128	4K	like MAB8051AH; automotive temperature; reduced frequency	DIL-40; SOT-129	31
MAF80A51AHWP	128	4K	like MAB8051AH; automotive temperature; reduced frequency	44-PLCC; SOT-187A	31
MAF8411P	64	1K	plus 8-bit LED driver; extended temperature	DIL-28; SOT-117	123
MAF84A11P	64	1K	plus 8-bit LED driver; automotive temperature; reduced frequency	DIL-28; SOT-117	123
MAF8421P	64	2K	plus 8-bit LED driver; extended temperature	DIL-28; SOT-117	123
MAF84A21P	64	2K	plus 8-bit LED driver; automotive temperature; reduced frequency	DIL-28; SOT-117	123
MAF8422P	64	2K	plus 8-bit LED driver; extended temperature	DIL-20; SOT-146	155
MAF84A22P	64	2K	plus 8-bit LED driver; automotive temperature; reduced frequency	DIL-20; SOT-146	123
MAF8441P	128	4K	plus 8-bit LED driver; extended temperature	DIL-28; SOT-117	123
MAF84A41P	128	4K	plus 8-bit LED driver; automotive temperature; reduced frequency	DIL-28; SOT-117	123
MAF8442P	128	4K	plus 8-bit LED driver; extended temperature	DIL-20; SOT-146	155
MAF84A42P	128	4K	plus 8-bit LED driver; automotive temperature; reduced frequency	DIL-20; SOT-146	155
MAF8461P	128	6K	plus 8-bit LED driver; extended temperature	DIL-28; SOT-117	123
MAF84A61P	128	6K	plus 8-bit LED driver; automotive temperature; reduced frequency	DIL-28; SOT-117	123

Operating temperature range: 0 to 70 °C.
Extended temperature range: −40 to + 85 °C.
Automotive temperature range: −40 to + 110 °C;
except MAF80A31/51AH only −40 to + 100 °C.

type number	description	package code	page
MEA8000	voice synthesizer	DIL-24; SOT-101A	1005
OM8200	speech demonstration board (for PCF8200)	standard Eurocard	1019
OM8201	speech demonstration box (for OM8200)	special pack	1023
OM8210	speech analysis/editing system (for PCF8200)	special pack	1025
PCB5010	digital signal processor woth on-chip ROM	68-PLCC; SOT-188A	855
PCB5011	ROM-less version of PCB5010	144-PGA	855

SINGLE-CHIP 8-BIT MICROCONTROLLERS (continued)

	RAM	ROM			
PCB80C31BHP	128	–	ROM-less version of PCB80C51	DIL-40; SOT-129	175
PCB80C31BHWP	128	–	ROM-less version of PCB80C51	44-PLCC; SOT-187A	175
PCB80C39P	128	–	ROM-less version of PCB80C49	DIL-40; SOT-129	203
PCB80C39WP	128	–	ROM-less version of PCB80C49	44-PLCC; SOT-187A	203
PCB80C49P	128	2K		DIL-40; SOT-129	203
PCB80C49WP	128	2K		44-PLCC; SOT-187A	203
PCB80C51BHP	128	4K	mask-programmable ROM	DIL-40; SOT-129	175
PCB80C51BHWP	128	4K	mask-programmable ROM	44-PLCC; SOT-187A	175
PCB80C552WP	256	–	ROM-less version of PCB83C552	68-PLCC; SOT-188A	233
PCB80C652WP	256	–	ROM-less version of PCB83C652	44-PLCC; SOT-187A	281
PCB83C552WP	256	8K	PCB80C51 plus additional functions	68-PLCC; SOT-188A	233
PCB83C652WP	256	8K	PCB80C51 plus additional functions	44-PLCC; SOT-187A	281
PCB85C51B	128	–	software development version of PCB80C51	DIL-40; SOT-215	319
PCB8582	256 x 8 EEPROM; I^2C bus			DIL-8; SOT-97AE	1061

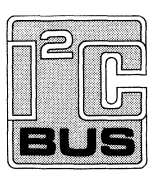
SINGLE-CHIP 8-BIT MICROCONTROLLERS (continued)

	RAM	ROM			
PCD3315CP	160	1,2K	telephony microcontroller	DIL-28; SOT-117	351
PCD3315CT	160	1,2K	telephony microcontroller	SO-28; SOT-136A	351
PCD3343D	224	3K	telephony microcontroller	DIL-28; SOT-135A	355
PCD3343P	224	3K	telephony microcontroller	DIL-28; SOT-117	355
PCD3343T	224	3K	telephony microcontroller	SO-28; SOT-136A	355
PCF80C31BHP	128	–	ROM-less version of PCB80C51	DIL-40; SOT-129	175
PCF80C31BHWP	128	–	ROM-less version of PCB80C51	44-PLCC; SOT-187A	175
PCF80C39P	128	–	ROM-less version of PCF80C49; extended temperature	DIL-40; SOT-129	203
PCF80C49P	128	2K	like PCB80C49; extended temperature	DIL-40; SOT-129	203
PCF80C51BHP	128	4K	mask-programmable ROM	DIL-40; SOT-129	175
PCB80C51BHWP	128	4K	mask-programmable ROM	44-PLCC; SOT-187A	175
PCF8200	voice synthesizer (CMOS)			DIL-24; SOT-101A	1029

Operating temperature range: 0 to 70 °C.
Extended temperature range: −40 to + 85 °C.
Automotive temperature range: −40 to + 110 °C.

type number	description			package code	page
SINGLE-CHIP 8-BIT MICROCONTROLLERS (continued)					
	RAM	ROM			
PCF84C00B	256	–	bond-out version PCF84CXX family	28/28 Piggy-back	423
PCF84C00T	256	–	bond-out version PCF84CXX family	VSO-56; SOT-190	423
PCF84C12P	64	1K		DIL-20; SOT-146	395
PCF84C12T	64	1K		SO-20; SOT-163A	395
PCF84C21P	64	2K	plus 8-bit LED driver; extended temperature	DIL-28; SOT-117	423
PCF84C21T	64	2K	plus 8-bit LED driver; extended temperature	SO-28; SOT-136A	423
PCF84C41P	128	4K	plus 8-bit LED driver; extended temperature	DIL-28; SOT-117	423
PCF84C41T	128	4K	plus 8-bit LED driver; extended temperature	SO-28; SOT-136A	423
PCF84C81P	256	8K	plus 8-bit LED driver; extended temperature	DIL-28; SOT-117	423
PCF84C81T	256	8K	plus 8-bit LED driver; extended temperature	SO-28; SOT-136A	423
PCF84C85P	256	8K	32 I/O; extended temperature	DIL-40; SOT-129	471
PCF84C85T	256	8K	32 I/O; extended temperature	VSO-40; SOT-158A	471
PCF8566P	universal LCD driver for low multiplex rates (1:1 to 1:4); max. 96 segments			DIL-40; SOT-129	1069
PCF8566T	universal LCD driver for low multiplex rates (1:1 to 1:4); max. 96 segments			VSO-40; SOT-158A	1069
PCF8570P	256 x 8 static RAM; I^2C bus			DIL-8; SOT-97AE	1099
PCF8570T	256 x 8 static RAM; I^2C bus			SO-8L; SOT-176	1099
PCF8571D	128 x 8 static RAM; I^2C bus			DIL-8; SOT-151A	1111
PCF8571P	128 x 8 static RAM; I^2C bus			DIL-8; SOT-97AE	1111
PCF8571T	128 x 8 static RAM; I^2C bus			SO-8L; SOT-176	1111
PCF8573P	clock calendar with serial I/O			DIL-16; SOT-38	1123
PCF8573T	clock calendar with serial I/O			SO-16L; SOT-162A	1123
PCF8574P	remote 8-bit I/O expander			DIL-16; SOT-38	1141
PCF8574T	remote 8-bit I/O expender			SO-16L; SOT-162A	1141
PCF8576T	universal LCD driver for low multiplex rates (1:1 to 1:4); max. 160 segments			VSO-56; SOT-190	1155
PCF8576U	universal LCD driver for low multiplex rates (1:1 to 1:4); max. 160 segments			uncased chip	1155

Operating temperature range: 0 to 70 °C.
Extended temperature range: −40 to + 85 °C.
Automotive temperature range: −40 to + 110 °C.



Purchase of Philips' I²C components conveys a license under the Philips' I²C patent to use the components in the I²C system provided the system conforms to the I²C specification defined by Philips.

MAINTENANCE TYPE LIST

The types listed below are not included in this handbook. Detailed information will be supplied on request.

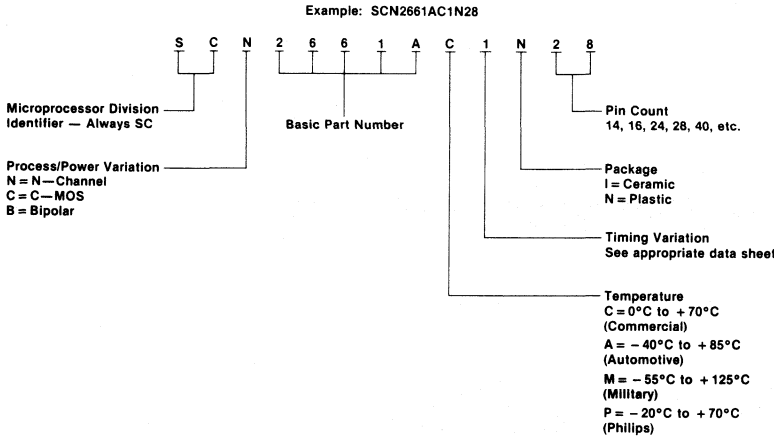
MAB8021 MAB8041A	single-chip 8-bit microcomputer single-chip 8-bit microcontroller	
MAB84XX MAF84XX MAF84AX1 FAMILY	single-chip 8-bit microcontroller	successor type: MAB84X1 MAF84X1 MAF84AX1 FAMILY
PCB80C31 PCB80C51	single-chip 8-bit microcontroller	successor type: PCB80C31B PCB80C51BH
TEA1002	PAL colour encoder and video summer	successor type: TEA2000

ORDERING INFORMATION

TYPE DESIGNATION

PRODUCT STATUS DEFINITIONS

PART NUMBERING SYSTEM



BIPOLAR LSI ORDERING INFORMATION

3X300 I/O Peripheral Components

Various 8X300/8X305 MicroController I/O parts and bus expanders can be ordered with an address preprogrammed at the factory or unprogrammed to permit feild address assignment. Addresses in range indicated as STOCK in the table below may be ordered in any quantity. Addresses outside of the STOCK range but indicated as AVAILABLE require a minimum order of 250 pieces per line item per part type. To order, use the part number indicated in the table, substituting the desired address of xx or xxx when ordering preprogrammed parts.

PART NUMBER	ADDRESSES		ORDER NUMBER	
	AVAILABLE	STOCK	UNPROGRAMMED	PREPROGRAMMED
N8T32F	000-255	None	N8T32F	N8T32F-xxx
N8T32N	000-255	000-015	N8T32N	N8T32N-xxx
N8T36F	000-255	None	N8T36F	N8T36F-xxx
N8T36N	000-255	000-015	N8T36N	N8T36N-xxx
N8X372N	000-255	000-015	N8X372N	N8X372N-xxx
N8X374N	000-255	000-015	N8X374N	N8X374-xxx
N8X376N	000-255	000-015	N8X376N	N8X376N-xxx
N8X382N	000-255	000-015	N8X382N	N8X382N-xxx

MCCAP 8X300/8X305 CROSSASSEMBLER PROGRAM

MCCAP, the crossassembler program for the 8X300 and 8X305 Micro-Controllers, is supplied as a 9-track magnetic tape containing FORTRAN IV source code for the crossassembler program. For compatibility with various computer systems, the tape is available in various combinations of density and data encoding. To order, use the following part numbers.

NUMBER	DENSITY	ENCODING
8X300 AS1-1 SS	800	ASCII
8X300 AS1-2 SS	800	EBCDIC
8X300 AS1-3 SS	1600	ASCII
8X300 AS1-4 SS	1600	EBCDIC
8X300 AS2SS	SINGLE/ DOUBLE	FLOPPY DISK

PRO ELECTRON TYPE DESIGNATION CODE FOR INTEGRATED CIRCUITS

This type nomenclature applies to semiconductor monolithic, semiconductor multi-chip, thin-film, thick-film and hybrid integrated circuits.

A basic number consists of:

THREE LETTERS FOLLOWED BY A SERIAL NUMBER

FIRST AND SECOND LETTER

1. DIGITAL FAMILY CIRCUITS

   The FIRST TWO LETTERS identify the FAMILY (see note 1).

2. SOLITARY CIRCUITS

   The FIRST LETTER divides the solitary circuits into:

   S : Solitary digital circuits
   T : Analogue circuits
   U : Mixed analogue/digital circuits

   The SECOND LETTER is a serial letter without any further significance except 'H' which stands for hybrid circuits.

3. MICROPROCESSORS

   The FIRST TWO LETTERS identify microprocessors and correlated circuits as follows:

   MA : Microcomputer / Central processing unit
   MB : Slice processor (see note 2)
   MD : Correlated memories
   ME : Other correlated circuits (interface, clock, peripheral controller, etc.)

4. CHARGE-TRANSFER DEVICES AND SWITCHED CAPACITORS

   The FIRST TWO LETTERS identify the following:

   NH : Hybrid circuits
   NL : Logic circuits
   NM : Memories
   NS : Analogue signal processing, using switched capacitors
   NT : Analogue signal processing, using CTDs
   NX : Imaging devices
   NY : Other correlated circuits

Notes

1. A logic family is an assembly of digital circuits designed to be interconnected and defined by its basic electrical characteristics (such as: supply voltage, power consumption, propagation delay, noise immunity).
2. By 'slice processor' is meant: a functional slice of microprocessor.

THIRD LETTER

It indicates the operating ambient temperature range.
The letters A to G give information about the temperature:

A : temperature range not specified
B : 0 to + 70 °C
C : −55 to + 125 °C
D : −25 to + 70 °C
E : −25 to + 85 °C
F : −40 to + 85 °C
G : −55 to + 85 °C

If a circuit is published for another temperature range, the letter indicating a narrower temperature range may be used or the letter 'A'.

Example: the range 0 to + 75 °C can be indicated by 'B' or 'A'.

SERIAL NUMBER

This may be either a 4-digit number assigned by Pro Electron, or the serial number (which may be a combination of figures and letters) of an existing company type designation of the manufacturer.

To the basic type number may be added:

A VERSION LETTER

Indicates a minor variant of the basic type or the package. Except for 'Z', which means customized wiring, the letter has no fixed meaning. The following letters are recommended for package variants:

C : for cylindrical
D : for ceramic DIL
F : for flat pack
L : for chip on tape
P : for plastic DIL
Q : for QIL
T : for miniature plastic (mini-pack)
U : for uncased chip

Alternatively a TWO LETTER SUFFIX may be used instead of a single package version letter, if the manufacturer (sponsor) wishes to give more information.

FIRST LETTER: General shape

C : Cylindrical
D : Dual-in-line (DIL)
E : Power DIL (with external heatsink)
F : Flat (leads on 2 sides)
G : Flat (leads on 4 sides)
K : Diamond (TO-3 family)
M : Multiple-in-line (except Dual-, Triple-, Quadruple-in-line)
Q : Quadruple-in-line (QIL)
R : Power QIL (with external heatsink)
S : Single-in-line
T : Triple-in-line

SECOND LETTER: Material

C : Metal-ceramic
G : Glass-ceramic (cerdip)
M : Metal
P : Plastic

A hyphen precedes the suffix to avoid confusion with a version letter.

DEFINITION OF TERMS

Data Sheet Identification	Product Status	Definition
Preview	Formative or In Design	This data sheet contains the design specifications for product development. Specifications may change in any manner without notice.
Advance Information	Sampling or Pre-Production	This data sheet contains advance information and specifications are subject to change without notice.
Preliminary	First Production	This data sheet contains preliminary data and supplementary data will be published at a later date. Signetics reserves the right to make changes at any time without notice in order to improve design and supply the best possible product.
No Identification Noted	Full Production	This data sheet contains final specifications. Signetics reserves the right to make changes at any time without notice in order to improve design and supply the best possible product.

RATING SYSTEMS

RATING SYSTEMS

The rating systems described are those recommended by the International Electrotechnical Commission (IEC) in its Publication 134.

DEFINITIONS OF TERMS USED

Electronic device. An electronic tube or valve, transistor or other semiconductor device.

Note
This definition excludes inductors, capacitors, resistors and similar components.

Characteristic. A characteristic is an inherent and measurable property of a device. Such a property may be electrical, mechanical, thermal, hydraulic, electro-magnetic, or nuclear, and can be expressed as a value for stated or recognized conditions. A characteristic may also be a set of related values, usually shown in graphical form.

Bogey electronic device. An electronic device whose characteristics have the published nominal values for the type. A bogey electronic device for any particular application can be obtained by considering only those characteristics which are directly related to the application.

Rating. A value which establishes either a limiting capability or a limiting condition for an electronic device. It is determined for specified values of environment and operation, and may be stated in any suitable terms.

Note
Limiting conditions may be either maxima or minima.

Rating system. The set of principles upon which ratings are established and which determine their interpretation.

Note
The rating system indicates the division of responsibility between the device manufacturer and the circuit designer, with the object of ensuring that the working conditions do not exceed the ratings.

ABSOLUTE MAXIMUM RATING SYSTEM

Absolute maximum ratings are limiting values of operating and environmental conditions applicable to any electronic device of a specified type as defined by its published data, which should not be exceeded under the worst probable conditions.

These values are chosen by the device manufacturer to provide acceptable serviceability of the device, taking no responsibility for equipment variations, environmental variations, and the effects of changes in operating conditions due to variations in the characteristics of the device under consideration and of all other electronic devices in the equipment.

The equipment manufacturer should design so that, initially and throughout life, no absolute maximum value for the intended service is exceeded with any device under the worst probable operating conditions with respect to supply voltage variation, equipment component variation, equipment control adjustment, load variations, signal variation, environmental conditions, and variations in characteristics of the device under consideration and of all other electronic devices in the equipment.

DESIGN MAXIMUM RATING SYSTEM

Design maximum ratings are limiting values of operating and environmental conditions applicable to a bogey electronic device of a specified type as defined by its published data, and should not be exceeded under the worst probable conditions.

These values are chosen by the device manufacturer to provide acceptable serviceability of the device, taking responsibility for the effects of changes in operating conditions due to variations in the characteristics of the electronic device under consideration.

The equipment manufacturer should design so that, initially and throughout life, no design maximum value for the intended service is exceeded with a bogey device under the worst probable operating conditions with respect to supply voltage variation, equipment component variation, variation in characteristics of all other devices in the equipment, equipment control adjustment, load variation, signal variation and environmental conditions.

DESIGN CENTRE RATING SYSTEM

Design centre ratings are limiting values of operating and environmental conditions applicable to a bogey electronic device of a specified type as defined by its published data, and should not be exceeded under normal conditions.

These values are chosen by the device manufacturer to provide acceptable serviceability of the device in average applications, taking responsibility for normal changes in operating conditions due to rated supply voltage variation, equipment component variation, equipment control adjustment, load variation, signal variation, environmental conditions, and variations in the characteristics of all electronic devices.

The equipment manufacturer should design so that, initially, no design centre value for the intended service is exceeded with a bogey electronic device in equipment operating at the stated normal supply voltage.

HANDLING MOS DEVICES

HANDLING MOS DEVICES

Though all our MOS integrated circuits incorporate protection against electrostatic discharges, they can nevertheless be damaged by accidental over-voltages. In storing and handling them, the following precautions are recommended.

Caution

Testing or handling and mounting call for special attention to personal safety. Personnel handling MOS devices should normally be connected to ground via a resistor.

Storage and transport

Store and transport the circuits in their original packing. Alternatively, use may be made of a conductive material or special IC carrier that either short-circuits all leads or insulates them from external contact.

Testing or handling

Work on a conductive surface (e.g. metal table top) when testing the circuits or transferring them from one carrier to another. Electrically connect the person doing the testing or handling to the conductive surface, for example by a metal bracelet and a conductive cord or chain. Connect all testing and handling equipment to the same surface.
Signals should not be applied to the inputs while the device power supply is off. All unused input leads should be connected to either the supply voltage or ground.

Mounting

Mount MOS integrated circuits on printed circuit boards *after* all other components have been mounted. Take care that the circuits themselves, metal parts of the board, mounting tools, and the person doing the mounting are kept at the same electric (ground) potential. If it is impossible to ground the printed-circuit board the person mounting the circuits should touch the board before bringing MOS circuits into contact with it.

Soldering

Soldering iron tips, including those of low-voltage irons, or soldering baths should also be kept at the same potential as the MOS circuits and the board.

Static charges

Dress personnel in clothing of non-electrostatic material (no wool, silk or synthetic fibres). After the MOS circuits have been mounted on the board proper handling precautions should still be observed. Until the sub-assemblies are inserted into a complete system in which the proper voltages are supplied, the board is no more than an extension of the leads of the devices mounted on the board. To prevent static charges from being transmitted through the board wiring to the device it is recommended that conductive clips or conductive tape be put on the circuit board terminals.

Transient voltages

To prevent permanent damage due to transient voltages, do not insert or remove MOS devices, or printed-circuit boards with MOS devices, from test sockets or systems with power on.

Voltage surges

Beware of voltage surges due to switching electrical equipment on or off, relays and d.c. lines.

SINGLE-CHIP 8-BIT MICROCONTROLLERS (NMOS, CMOS)*

* User manual 1986
"Single-chip 8-bit Microcontrollers"
Available on request

SINGLE-CHIP 8-BIT MICROCONTROLLER

DESCRIPTION

The MAB8051AH family of single-chip 8-bit microcontrollers is manufactured in an advanced 2 μ NMOS process. The family consists of the following members:

- MAB8031AH: ROM-less version of the MAB8051AH
- MAB8051AH: 4 K bytes mask-programmable ROM, 128 bytes RAM

Both types are available in 8, 10 and 12 MHz versions and 15 MHz for the MAB8031AH. In the following, the generic term "MAB8051AH" is used to refer to both family members.

The device provides hardware features, architectural enhancements and new instructions to function as a controller for applications requiring up to 64 K bytes of program memory and/or up to 64 K bytes of data storage.

The MAB8051AH contains a non-volatile 4 K x 8 read-only program memory (not ROM-less version); a volatile 128 x 8 read/write data memory; 32 I/O lines; two 16-bit timer/event counters; a five-source, two-priority-level, nested interrupt structure; a serial I/O power for either multi-processor communications, I/O expansion, or full duplex UART; and on-chip oscillator and timing circuits. For systems that require extra capability, the MAB8051AH can be expanded using standard TTL compatible memories and logic.

The device also functions as an arithmetic processor having facilities for both binary and BCD arithmetic plus bit-handling capabilities. The instruction set consists of 255 instructions; 44% one-byte, 41% two-byte and 15% three-byte. With a 12 MHz crystal, 58% of the instructions are executed in 1 μs and 40% in 2 μs. Multiply and divide instructions require 4 μs. Multiply, divide, subtract and compare are among the many instructions added to the standard MAB8048H instruction set.
For further detailed information see users manual 'Single-chip 8-bit microcontrollers'.

Features

- 4 K x 8 ROM (8051AH only), 128 x 8 RAM
- Four 8-bit ports, 32 I/O lines
- Two 16-bit timer/event counters
- Full duplex serial port
- External memory expandable to 128 K
- Boolean processing
- 218 bit-addressable locations
- On-chip oscillator
- Five-source interrupt structure with two priority levels
- 58% of instructions executed in 1 μs; multiply and divide in 4 μs (at 12 MHz clock)
- Enhanced architecture with:
  - non-page-oreinted instructions
  - direct addressing
  - four 8-bit register banks
  - stack depth up to 128-bytes
  - multiply, divide, subtract and compare
- Available with extended temperature range: −40 to + 85 °C (MAF8031/51AH)
- Available with automotive temperature range: −40 to + 100 °C (MAF80A31/51AH)

PACKAGE OUTLINES

MAB8031/51AHP; MAF8031/51AHP; MAF80A31/51AHP: 40-lead DIL; plastic (SOT-129).
MAB8031/51AHWP; MAF8031/51AHWP; MAF80A31/51AHWP: 44-lead, plastic leaded-chip-carrier (PLCC); SOT-187A.

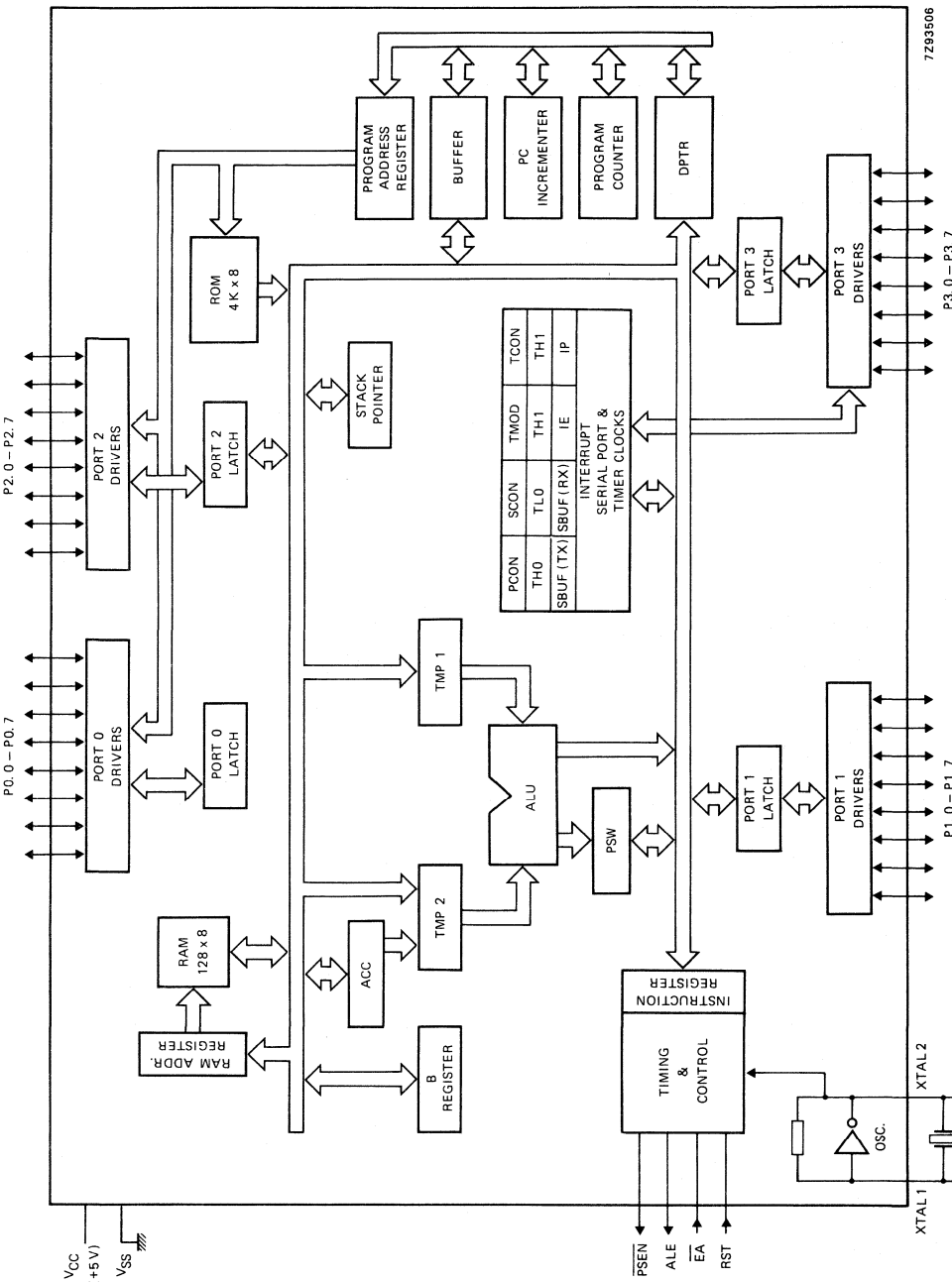


Fig. 1 Block diagram.

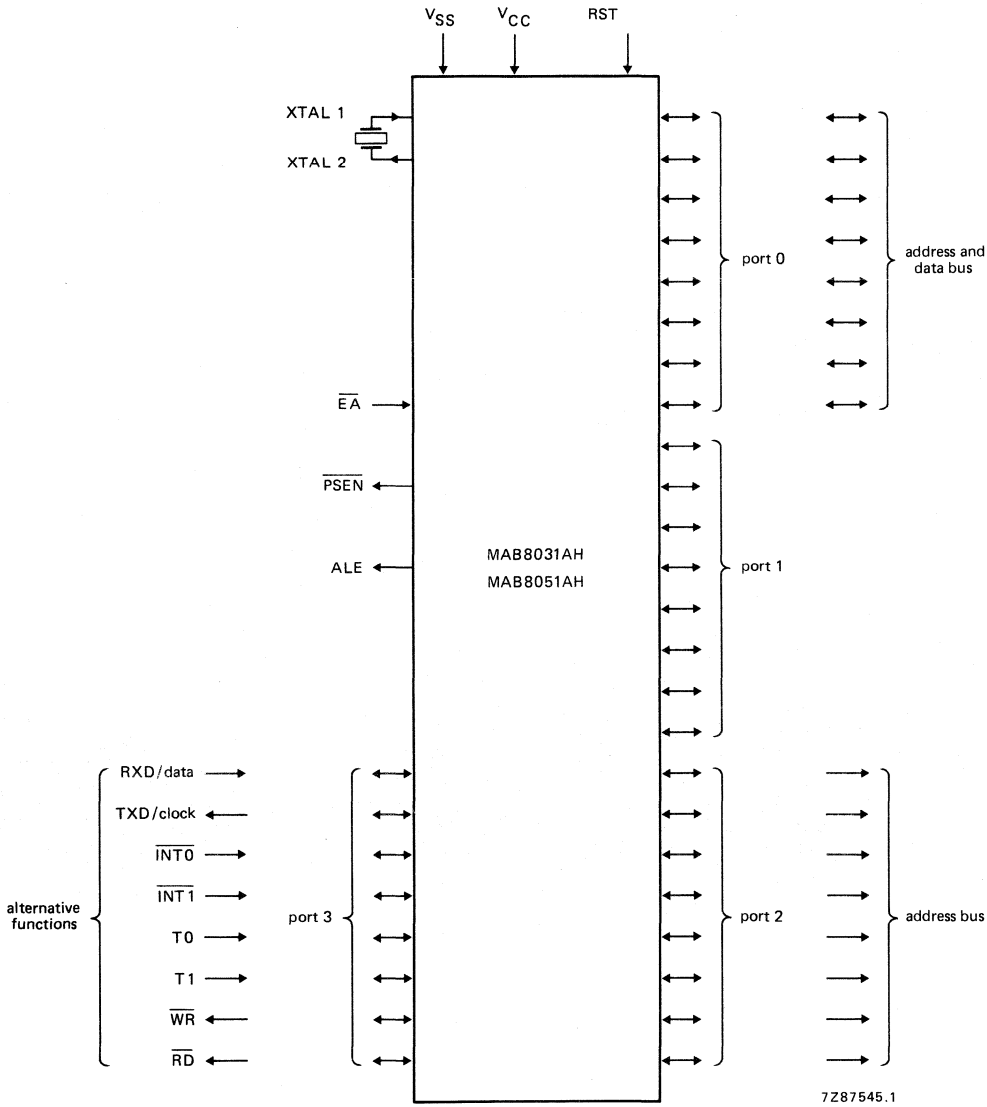


Fig. 2 Functional diagram.

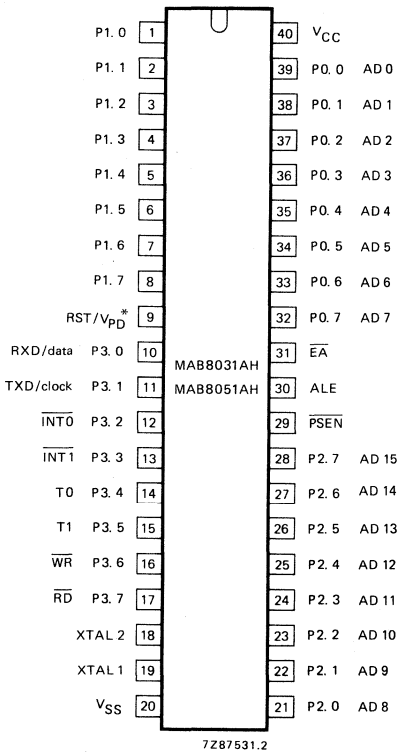


Fig. 3a Pinning diagram for MAB8031/51AHP; MAF8031/51AHP; MAF80A31/51AHP.

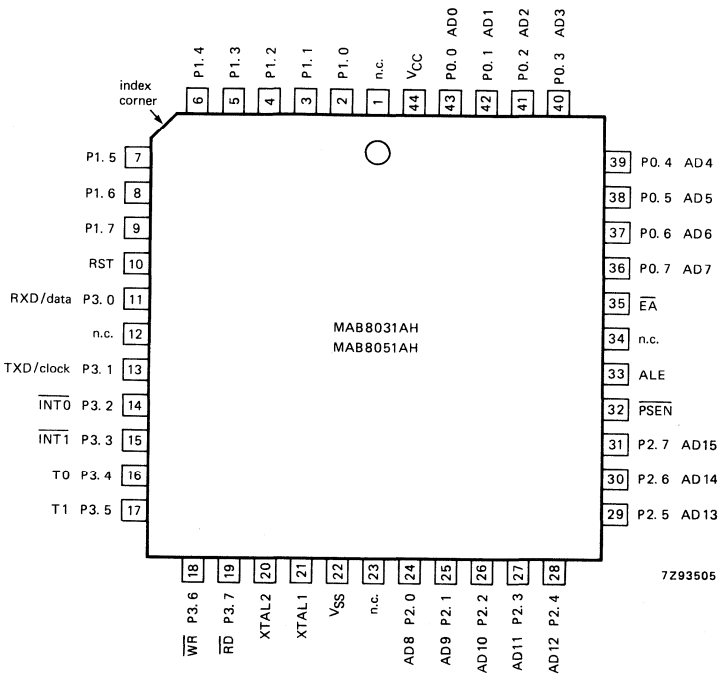


Fig. 3b Pinning diagram for MAB8031/51AHWP; MAF8031/51AHWP; MAF80A31/51AHWP.

* V_{PD} option available on request.

PINNING (DIL package)

1-8	P1.0-P1.7	**Port 1:** 8-bit quasi-bidirectional I/O port. It receives the low-order address byte during program verification. Port 1 can sink/source one TTL (= 4 LS TTL) input. It can drive MOS inputs without external pull-ups.
9		**RST/V_{PD} RESET/POWER DOWN:** a high level on this pin for two machine cycles while the oscillator is running resets the device. An internal pulldown permits Power-On reset using only a capacitor connected to V_{CC}. As an available option, this pin also supplies standby power to the RAM: V_{PD} should be held within its specified limit while V_{CC} drops below its specified limit. When V_{PD} is LOW the RAM current is drawn from V_{CC}.
10-17	P3.0-P3.7	**Port 3:** 8-bit quasi-bidirectional I/O port with internal pull-ups. It also serves the following alternative functions:

Port pin	*Alternative function*
P3.0	**RXD/data:** serial port receiver data input (asynchronous) or data input/output (synchronous)
P3.1	**TXD/clock:** serial port transmitter data output (asynchronous) or clock output (synchronous)
P3.2	**$\overline{\text{INT0}}$:** external interrupt 0 or gate control input for timer/event counter 0
P3.3	**$\overline{\text{INT1}}$** : external interrupt 1 or gate control input for timer/event counter 1
P3.4	**T0** : external input for timer/event counter 0
P3.5	**T1** : external input for timer/event counter 1
P3.6	**$\overline{\text{WR}}$**: external data memory write strobe
P3.7	**$\overline{\text{RD}}$** : external data memory read strobe.

Operation of an alternative function is determined by the relevant output latch programmed to logic 1. Port 3 can sink/source one TTL input. It can drive MOS inputs without external pull-ups.

18	XTAL 2	**Crystal input 2:** output of the inverting amplifier that forms the oscillator, and input to the internal clock generator. Receives the external oscillator signal when an external oscillator is used (see figures 6 and 7).
19	XTAL 1	**Crystal input 1:** input to the inverting amplifier that forms the oscillator. Connected to V_{SS} when an external oscillator is used (see figures 6 and 7).
20	V_{SS}	**Ground:** circuit ground potential.
21-28	P2.0-P2.7	**Port 2:** 8-bit quasi-bidirectional I/O port with internal pull-ups. It emits the high-order address byte when accessing external memory. It also receives the high-order address bits and control signals during program verification. Port 2 can sink/source one TTL input. It can drive MOS inputs without external pull-ups.
29	$\overline{\text{PSEN}}$	**Program Store Enable output:** read strobe to the external Program Memory. It is activated twice each machine cycle during fetches from external Program Memory. When executing out of external Program Memory two activations of $\overline{\text{PSEN}}$ are skipped during each access to external Data Memory. $\overline{\text{PSEN}}$ is not activated (remains HIGH) during fetches from internal Program Memory.
30	ALE	**Address Latch Enable output:** latches the low byte of the address during accesses to external memory in normal operation. It is activated every six oscillator periods except during an external data memory access.

PINNING (continued)

31	$\overline{EA}$	When $\overline{EA}$ is held at a TTL high level the CPU executes out of the internal Program Memory (ROM), provided the Program Counter is less than 4096. When $\overline{EA}$ is held at a TTL low level the CPU executes out of external Program Memory. $\overline{EA}$ does not float.
32-39	P0.7-P0.0	**Port 0**: 8-bit open drain bidirectional I/O port. It is also the multiplexed low-order address and data bus during accesses to external memory (during these accesses it activates internal pull-ups). It also outputs instruction bytes during program verification. (External pull-ups are required during program verification). Port 0 can sink/source two TTL inputs.
40	V_{CC}	**Power Supply**: + 5 V power supply pin during normal operation.

FUNCTIONAL DESCRIPTION

General

The MAB8051AH is a stand-alone high-performance microcontroller designed for use in real-time applications such as instrumentation, industrial control and intelligent computer peripherals.

The device provides hardware features, architectural enhancements and new instructions to function as a controller for applications requiring up to 64 K bytes of program memory and/or up to 64 K bytes of data storage.

The MAB8031AH is a control-oriented CPU without on-chip program memory. It can address 64 K bytes of external program memory in addition to 64 K bytes of external data memory. The MAB8051AH is a MAB8031AH with the lower 4 K bytes of program memory filled with on-chip mask programmable ROM. For systems requiring extra capability, the MAB8051AH can be expanded using standard TTL memories and peripherals.

The two pin-compatible versions of this component reduce development problems to a minimum and provide maximum flexibility. The MAB8051AH is for low-cost, high volume production; and the MAB8031AH for applications requiring the flexibility of external program memory which can be easily modified and updated in the field.

The MAB8051AH contains a non-volatile 4 K x 8 read-only program memory; a volatile 128 x 8 read/write data memory; 32 I/O lines; two 16-bit timer/event counters; a five-source, two-priority-level, nested interrupt structure; a serial I/O port for either multi-processor communications, I/O expansion, or full duplex UART; and on-chip oscillator and timing circuits.

Central processing unit

The central processing unit (CPU) manipulates operands in four memory spaces. These are the 64 K-byte external data memory, 256-byte internal data memory and 16-bit program counter spaces. The internal data memory address space is sub-divided into the 128-byte internal data RAM and 128-byte special function register (SFR) address spaces, as shown in Fig. 4a. See also Figures 4b to 4f.

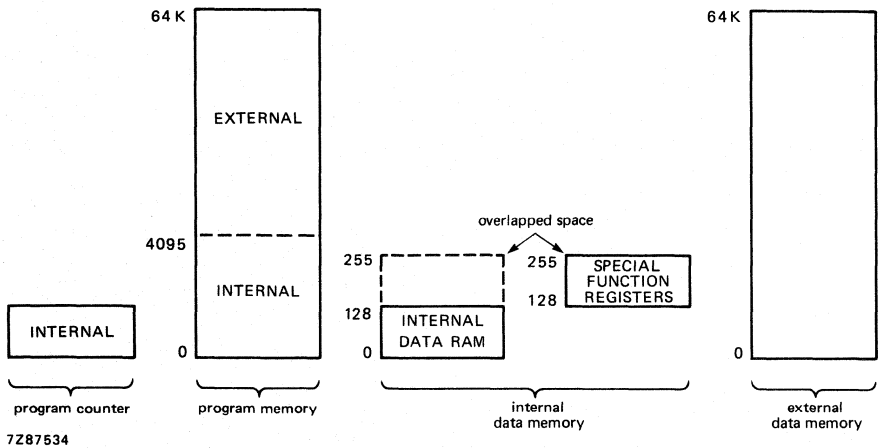


Fig. 4a Memory map.

Where:

The internal data memory locations are addressable direct/indirect as follows:

location	addressed
RAM 0 to 127	direct and indirect
SFR 128 to 255	direct only

FUNCTIONAL DESCRIPTION (continued)

The internal data RAM contains four register banks (each with eight registers), 128 addressable bits, and the stack. The stack depth is limited by the available internal data RAM and its location is determined by the 8-bit stack pointer. All registers except the program counter and the four 8-register banks reside in the special function register (SFR) address space. These memory mapped registers include arithmetic registers, pointers, I/O ports, interrupt system registers, timers and serial port. There are 128 addressable bit locations in the SFR address space.

The MAB8051AH contains 128 bytes of internal data RAM and 20 special function registers. It provides a non-paged program memory address space to accommodate relocatable code. Conditional branches are performed relative to the program counter. The register-indirect jump permits branching relative to a 16-bit base register with an offset provided by an 8-bit index register. 16-bit jumps and calls permit branching to any location in the contiguous 64 K program memory address space.

The MAB8051AH has five methods for addressing source operands:

- Register
- Direct
- Register-Indirect
- Immediate
- Base-Register-plus Index-Register-Indirect

The first three methods can be used for addressing destination operands. Most instructions have a "destination source" field that specifies the data type, addressing methods and operands involved. For operations other than moves, the destination operand is also a source operand.

Access addressing is as follows:

- Registers in the four 8-register banks through Register, Direct, or Register-Indirect.
- 128 bytes of internal data RAM through Direct or Register-Indirect.
- Special function registers through Direct.
- External data memory through Register-Indirect.
- Program memory look-up tables through Base-Register-plus Index-Register-Indirect.

The MAB8051AH is classified as an 8-bit device since the internal ROM, RAM, Special Function Registers (SFR), Arithmetic Logic Unit (ALU), and external data bus are each 8-bits wide. It performs operations on bit, nibble, byte and double-byte data types.

Facilities are available for byte transfer, logic, and integer arithmetic operations. Data transfer, logic, and conditional branch operations can be performed directly on Boolean variables to provide excellent bit handling.

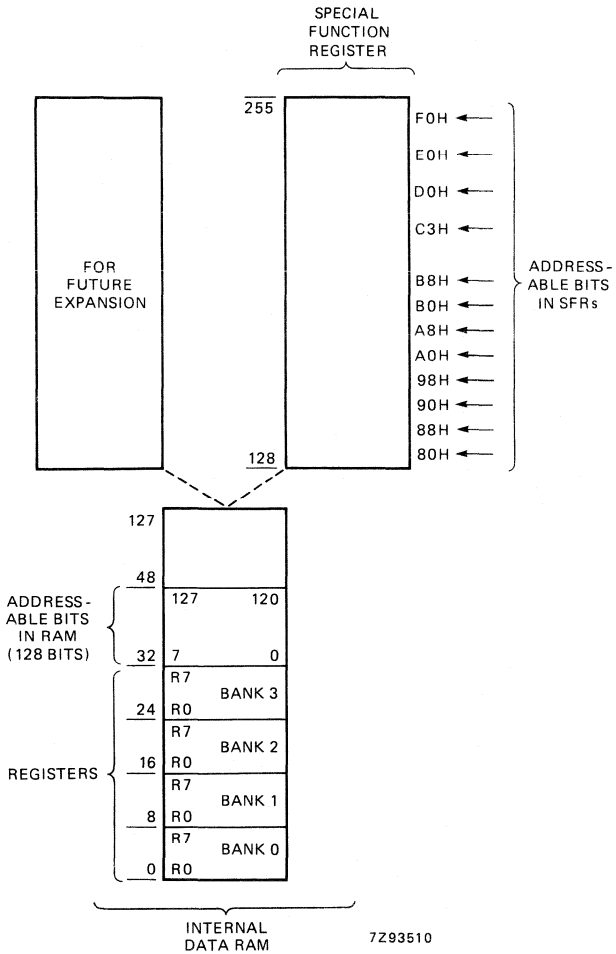


Fig. 4b Internal data memory address space.

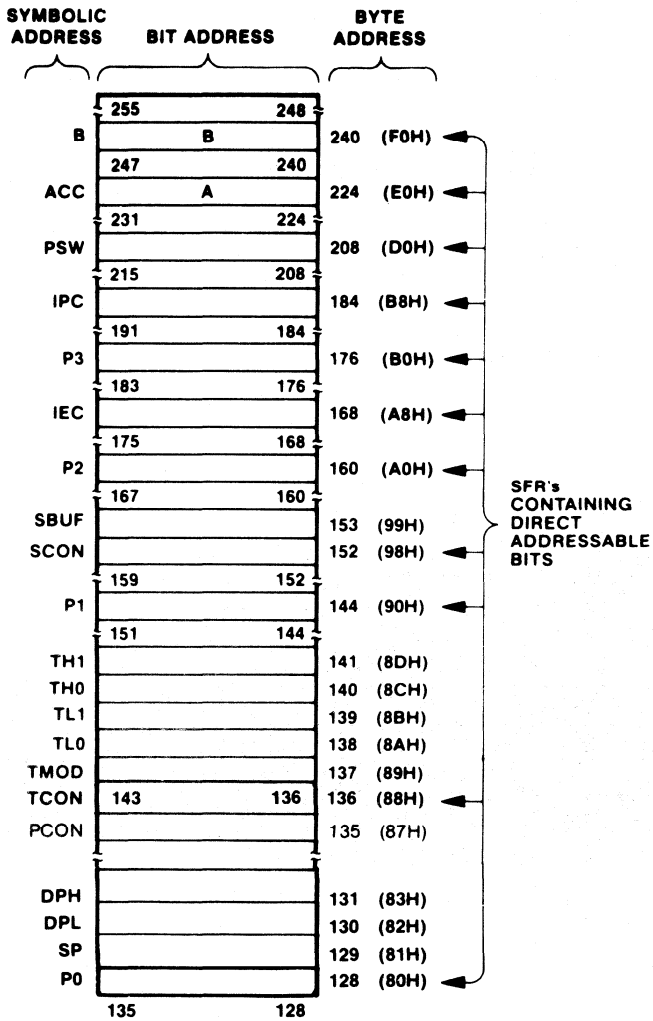


Fig. 4c Mapping of special function registers.

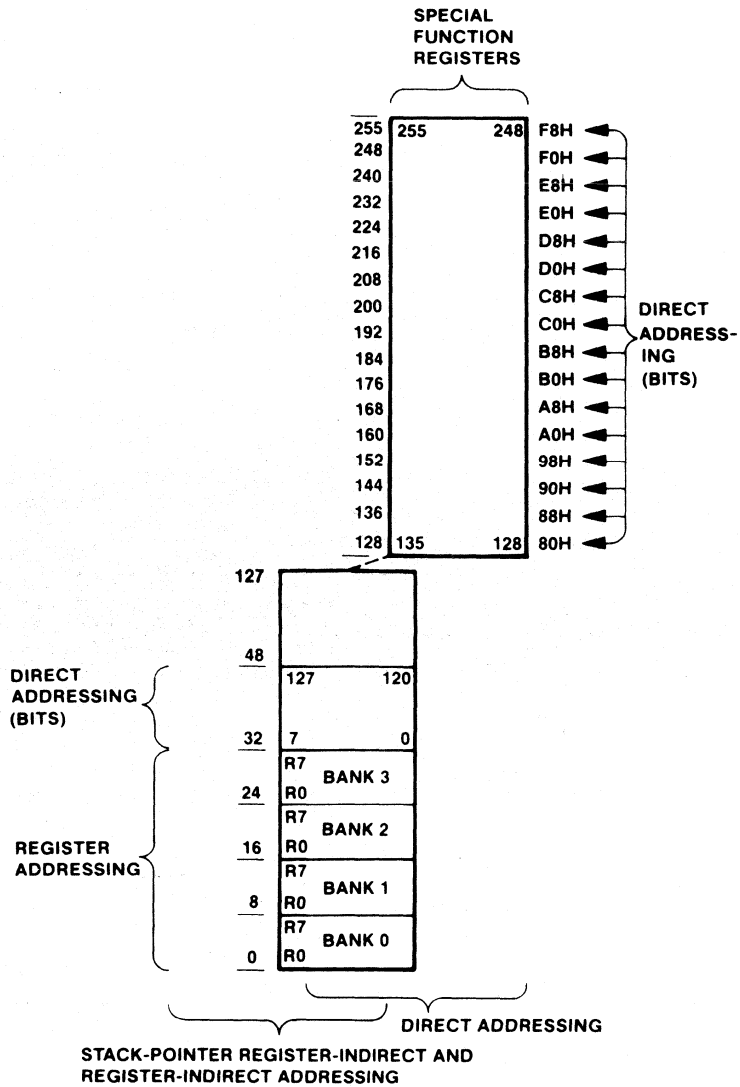


Fig. 4d Special function register bit addresses.

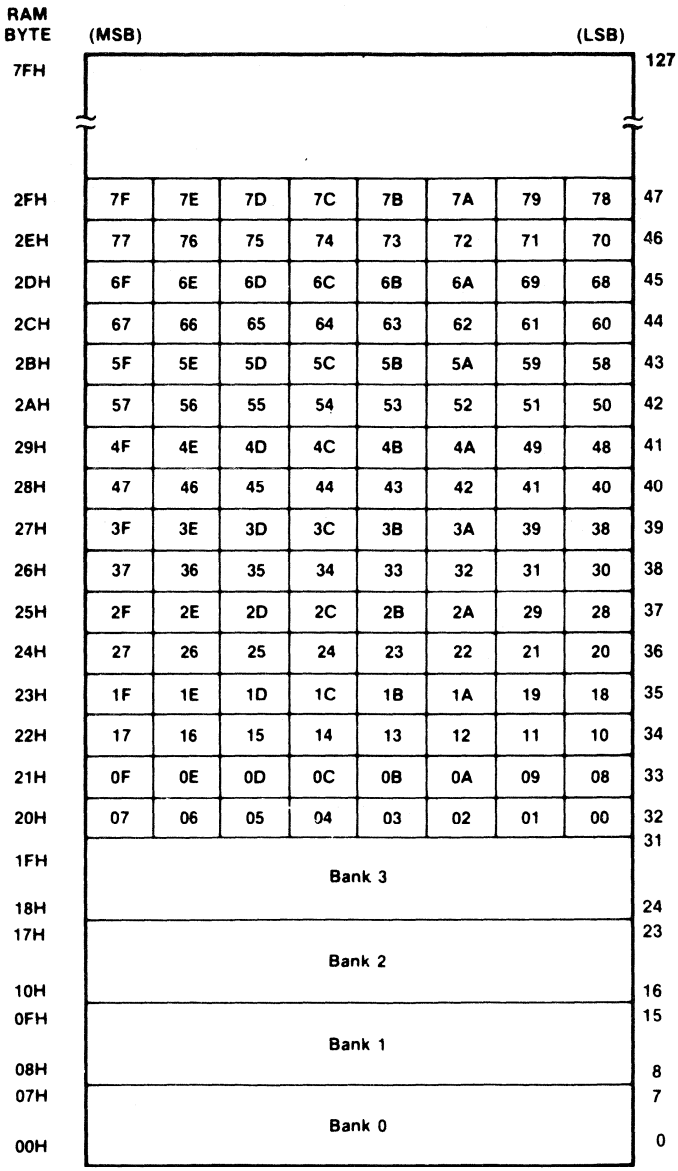


Fig. 4e RAM bit addresses.

Direct Byte Address	Bit Addresses (MSB)							(LSB)	Hardware Register Symbol
240	F7	F6	F5	F4	F3	F2	F1	F0	B
224	E7	E6	E5	E4	E3	E2	E1	E0	ACC
	CY	AC	FO	RS1	RS0	OV		P	
208	D7	D6	D5	D4	D3	D2	D1	D0	PSW
				PS	PT1	PX1	PT0	PX0	
184	—	—	—	BC	BB	BA	B9	B8	IP
176	B7	B6	B5	B4	B3	B2	B1	B0	P3
	EA			ES	ET1	EX1	ET0	EX0	
168	AF	—	—	AC	AB	AA	A9	A8	IE
160	A7	A6	A5	A4	A3	A2	A1	A0	P2
	SMO	SM1	SM2	REN	TB8	RB8	TI	RI	
152	9F	9E	9D	9C	9B	9A	99	98	SCON
144	97	96	95	94	93	92	91	90	P1
	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	
136	8F	8E	8D	8C	8B	8A	89	88	TCON
128	87	86	85	84	83	82	81	80	P0

Fig. 4f Addressing operands in internal data memory.

FUNCTIONAL DESCRIPTION (continued)

I/O facilities

The MAB8051AH has 32 I/O lines treated as 32 individual addressable bits and as four parallel 8-bit addressable ports. Ports 0, 1, 2 and 3 perform the following alternate functions:

- Port 0; provides the multiplexed low-order address and data bus used for expanding the MAB8051AH with standard memories and peripherals
- Port 2; provides the high-order address bus when expanding the MAB8051 with external program memory or more than 256 bytes of external data memory
- Port 3; pins can be configured individually to provide:-

  external interrupt requests inputs
  counter inputs
  serial port receiver input and transmitter output
  control signals to READ and WRITE to external data memory

The generation or use of a Port 3 pin as an alternate function is carried out automatically by the MAB8051AH provided the pin is configured as an output.

Timer/event counters

The MAB8051AH contains two 16-bit registers, Timer 0 and Timer 1, that can be used as timers or event counters to carry out the following functions:

- Measure time intervals and pulse durations
- Count events
- Generate interrupt requests

Each timer/event counter can be programmed independently to operate in three modes:

- Mode 0; 8-bit timer or 8-bit counter each with divide by 32 prescaler
- Mode 1; 16-bit time-interval or event counter
- Mode 2; 8-bit time-interval or event counter with automatic reload upon overflow

Counter 0 can be programmed to operate in an additional mode as follows:

- Mode 3; one 8-bit time-interval or event counter and one 8-bit time-interval counter

When counter 0 is in Mode 3, counter 1 can be programmed to operate in Modes 0, 1 or 2 but cannot set an interrupt request flag or generate an interrupt. However the overflow from counter 1 can be used to pulse the serial Port transmission-rate generator.

The frequency handling range of these counters with a 3,5 to 12 MHz crystal is as follows:

- 0,3 to 1 MHz when programmed for an input that is a division by 12 of the oscillator frequency
- 0 Hz to an upper limit of 150 kHz to 0,5 MHz when programmed for external inputs

Both internal and external inputs can be gated to the counter by a second external source for directly measuring pulse durations.

The counters are started and stopped under software control. Each one sets its interrupt request flag when it overflows from all 1's to all 0's (or automatic reload value).

On-chip peripheral functions

In addition to the CPU and memories, an interrupt system, extensive I/O facilities, and several peripheral functions are integrated on-chip to relieve the CPU of repetitious, complicated or time-critical tasks and to permit stringent real-time control of external system interfaces. The I/O facilities include the I/O pins, parallel ports, bidirectional address/data bus and the serial port for I/O expansion. The CPU peripheral functions integrated on-chip are the two 16-bit timer/event counters and the serial port.

Interrupt system (see Fig. 5)

External events and the real-time-driven on-chip peripherals require service by the CPU asynchronous to the execution of any particular section of code. To tie the asynchronous activities of these functions to normal program execution a multiple-source, two-priority-level, nested interrupt system is provided. Interrupt response latency is from 3 μs to 7 μs when using a 12 MHz crystal.

The MAB8051AH acknowledge interrupt requests from five sources as follows:

- $\overline{\text{INT0}}$ and $\overline{\text{INT1}}$; externally via pins 12 and 13 respectively
- Timer 0 and Timer 1; from the two internal counters
- Serial Port; from the internal serial I/O port

Each interrupt vectors to a separate location in program memory for its service program.
Each source can be individually enabled or disabled and can be programmed to a high or low priority level. Also all enabled sources can be globally disabled or enabled. Both external interrupts can be programmed to be level-activated or transition-activated and is active LOW to allow "wire-ORing" of several interrupt sources to the input pin.

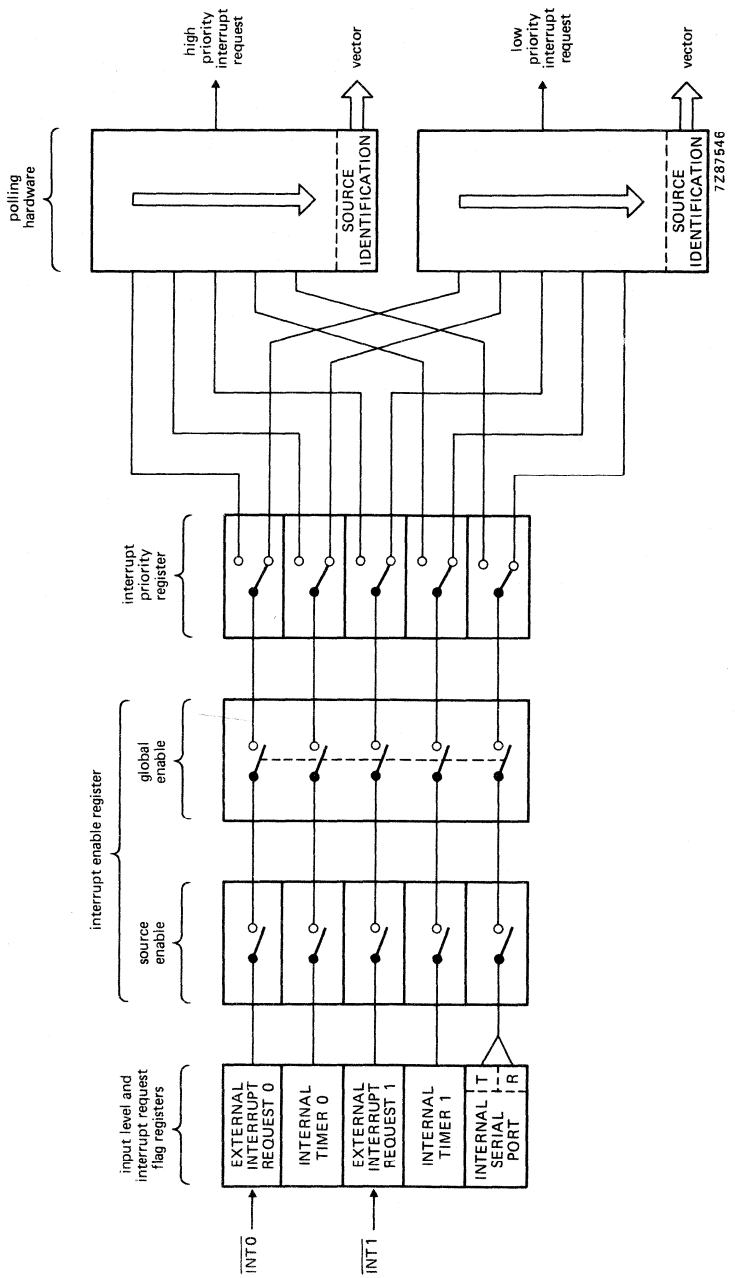


Fig. 5 Interrupt system.

OSCILLATOR CIRCUITRY

The oscillator circuitry of the MAB8051AH is a single-stage inverting amplifier in a Pierce oscillator configuration. The circuitry has a combination of depletion and enhancement mode MOS FETs to produce the inverting characteristics, and not passive components. Either a crystal or ceramic resonator can be used as the feedback element to complete the oscillator circuitry. XTAL1, pin 19, is the high gain amplifier input, and XTAL2, pin 18, is the output (see Fig. 6).

To drive the MAB8051AH externally, XTAL1 should be connected to ground and XTAL2 driven from an external source (see Fig. 7).

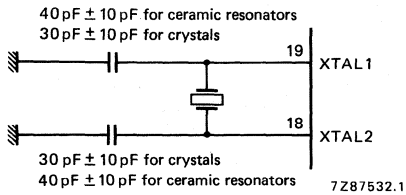


Fig. 6 MAB8051AH oscillator circuit.

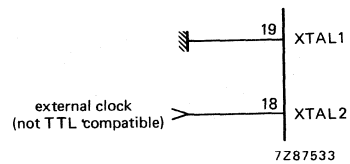


Fig. 7 Driving the MAB8051AH from an external source.

RESET CIRCUITRY

The reset circuitry for the MAB8051AH is connected to the reset pin, RST, as shown in Fig. 8. A Schmitt trigger is used th the input for noise rejection. The output of the Schmitt trigger is sampled by the reset circuitry every machine cycle.

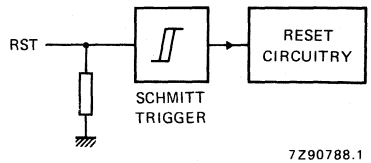


Fig. 8 Reset configuration at RST.

A reset is accomplished by holding the RST pin HIGH for at least two machine cycles (24 oscillator periods), while the oscillator is running. The CPU responds by executing an internal reset. It also configures the ALE and $\overline{PSEN}$ pins as inputs. (They are quasi-bidirectional.) The internal reset is executed during the second cycle in which RST is high and is repeated every cycle until RST goes LOW. It leaves the internal registers as follows:

REGISTER	CONTENT
PC	0000H
ACC	00H
B	00H
PSW	00H
SP	07H
DPTR	0000H
P0 -- P3	0FFH
IP	XXX00000B
IE	XXX00000B
TMOD	00H
TCON	00H
TH0	00H
TL0	00H
TH1	00H
TL1	00H
SCON	00H
SBUF	Indeterminate
PCON	0XXXXXXXB

Where

H = Hexadecimal
B = Binary

The internal RAM is not affected by reset. When VCC is turned on, the RAM content is determinate.

RESET CIRCUITRY (continued)

Power-on reset

An automatic reset when V_{CC} is turned on can be obtained by connecting the RST pin to V_{CC} through a 10 μF capacitor, as long as the V_{CC} rise-time does not exceed 10 milliseconds. This power-on reset circuit is shown in Fig. 9. When the power is switched on, the current drawn by RST is the difference between V_{CC} and the capacitor voltage, and decreases from V_{CC} as the capacitor charges. The larger the capacitor, the more slowly V_{RST} decreases. V_{RST} must remain above the lower threshold of the Schmitt trigger long enough to effect a complete reset. The time required is the oscillator start-up time, plus 2 machine cycles.

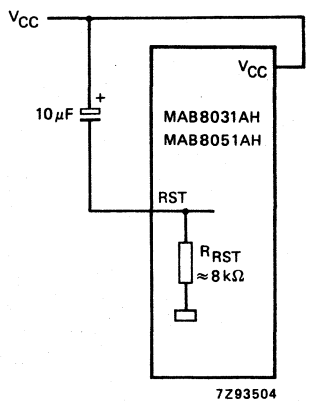


Fig. 9 Power-on reset.

INSTRUCTION SET

The MAB8051AH uses a powerful instruction set to allow expansion of on-chip CPU peripherals and to optimize byte efficiency and execution speed. Reassigned opcodes add new high-power operations and permit new addressing modes to make old operations more orthogonal. The instruction set consists of 49 single-byte, 45 two-byte and 17 three-byte instructions. When using a 12 MHz oscillator, 64 instructions execute in 1 μs and 45 instructions execute in 2 μs. Multiply and divide instructions execute in 4 μs.

Table 1 Instruction set description

mnemonic		description	bytes/ cycles	opcode (hex.)
Arithmetic operation				
ADD	A,Rr	Add register to A	1 1	2*
ADD	A,direct	Add direct byte to A	2 1	25
ADD	A,@Ri	Add indirect RAM to A	1 1	26, 27
ADD	A,#data	Add immediate data to A	2 1	24
ADDC	A,Rr	Add register to A with carry flag	1 1	3*
ADDC	A,direct	Add direct byte to A with carry flag	2 1	35
ADDC	A,@Ri	Add indirect RAM to A with carry flag	1 1	36, 37
ADDC	A,#data	Add immediate data to A with carry flag	2 1	34
SUBB	A,Rr	Subtract register from A with borrow	1 1	9*
SUBB	A,direct	Subtract direct byte from A with borrow	2 1	95
SUBB	A,@Ri	Subtract indirect RAM from A with borrow	1 1	96, 97
SUBB	A,#data	Subtract immediate data from A with borrow	2 1	94
INC	A	Increment A	1 1	04
INC	Rr	Increment register	1 1	0*
INC	direct	Increment direct byte	2 1	05
INC	@Ri	Increment indirect RAM	1 1	06, 07
DEC	A	Decrement A	1 1	14
DEC	Rr	Decrement register	1 1	1*
DEC	direct	Decrement direct byte	2 1	15
DEC	@Ri	Decrement indirect RAM	1 1	16, 17
INC	DPTR	Increment data pointer	1 2	A3
MUL	AB	Multiply A & B	1 4	A4
DIV	AB	Divide A by B	1 4	84
DA	A	Decimal adjust A	1 1	D4

mnemonic		description	bytes/ cycles	opcode (hex.)
Logic operations				
ANL	A,Rr	AND register to A	1 1	5*
ANL	A,direct	AND direct byte to A	2 1	55
ANL	A,@Ri	AND indirect RAM to A	1 1	56, 57
ANL	A,#data	AND immediate data to A	2 1	54
ANL	direct,A	AND A to direct byte	2 1	52
ANL	direct,#data	AND immediate data to direct byte	3 2	53
ORL	A,Rr	OR register to A	1 1	4*
ORL	A,direct	OR direct byte to A	2 1	45
ORL	A,@Ri	OR indirect RAM to A	1 1	46, 47
ORL	A,#data	OR immediate data to A	2 1	44
ORL	direct,A	OR A to direct byte	2 1	42
ORL	direct,#data	OR immediate data to direct byte	3 2	43
XRL	A,Rr	Exclusive-OR register to A	1 1	6*
XRL	A,direct	Exclusive-OR direct byte to A	2 1	65
XRL	A,@Ri	Exclusive-OR indirect RAM to A	1 1	66, 67
XRL	A,#data	Exclusive-OR immediate data to A	2 1	64
XRL	direct, A	Exclusive-OR to direct byte	2 1	62
XRL	direct,#data	Exclusive-OR immediate data to direct byte	3 2	63
CLR	A	Clear A	1 1	E4
CPL	A	Complement A	1 1	F4
RL	A	Rotate A left	1 1	23
RLC	A	Rotate A left through the carry flag	1 1	33
RR	A	Rotate A right	1 1	03
RRC	A	Rotate A right through the carry flag	1 1	13
SWAP	A	Swap nibbles within A	1 1	C4

INSTRUCTION SET (continued)

mnemonic		description	bytes/ cycles	opcode (hex.)
Data transfer				
MOV**	A,Rr	Move register to A	1 1	E*
MOV	A,direct	Move direct byte to A	2 1	E5
MOV	A,@Ri	Move indirect RAM to A	1 1	E6, E7
MOV	A,#data	Move immediate data to A	2 1	74
MOV	Rr,A	Move A to register	1 1	F*
MOV	Rr,direct	Move direct byte to register	2 2	A*
MOV	Rr,#data	Move immediate data to register	2 1	7*
MOV	direct,A	Move A to direct byte	2 1	F5
MOV	direct,Rr	Move register to direct byte	2 2	8*
MOV	direct,direct	Move direct byte to direct	3 2	85
MOV	direct,@Ri	Move indirect RAM to direct byte	2 2	86, 87
MOV	direct,#data	Move immediate data to direct byte	3 2	75
MOV	@Ri,A	Move A to indirect RAM	1 1	F6, F7
MOV	@Ri,direct	Move direct byte to indirect RAM	2 2	A6, A7
MOV	@Ri,#data	Move immediate data to indirect RAM	2 1	76, 77
MOV	DPTR,#data16	Load data pointer with a 16-bit constant	3 2	90
MOVC	A,@A+DPTR	Move code byte relative to DPTR to A	1 2	93
MOVC	A,@A+PC	Move code byte relative to PC to A	1 2	83
MOVX	A,@Ri	Move external RAM (8-bit address) to A	1 2	E2, E3
MOVX	A,@DPTR	Move external RAM (16-bit address) to A	1 2	E0
MOVX	@Ri,A	Move A to external RAM (8-bit address)	1 2	F2, F3
MOVX	@DPTR,A	Move A to external RAM (16-bit address)	1 2	F0
PUSH	direct	Push direct byte onto stack	2 2	C0
POP	direct	Pop direct byte from stack	2 2	D0
XCH	A,Rr	Exchange register with A	1 1	C*
XCH	A,direct	Exchange direct byte with A	2 1	C5
XCH	A,@Ri	Exchange indirect RAM with A	1 1	C6, C7
XCHD	A,@Ri	Exchange LOW-order digit indirect RAM with A	1 1	D6, D7

** MOV A,ACC not valid instruction.

mnemonic		description	bytes/ cycles	opcode (hex.)
Boolean variable manipulation				
CLR	C	Clear carry flag	1 1	C3
CLR	bit	Clear direct bit	2 1	C2
SETB	C	Set carry flag	1 1	D3
SETB	bit	Set direct bit	2 1	D2
CPL	C	Complement carry flag	1 1	B3
CPL	bit	Complement direct bit	2 1	B2
ANL	C,bit	AND direct bit to carry flag	2 2	82
ANL	C,/bit	AND complement of direct bit to carry flag	2 2	B0
ORL	C,bit	OR direct bit to carry flag	2 2	72
ORL	C,/bit	OR complement of direct bit to carry flag	2 2	A0
MOV	C,bit	Move direct bit to carry flag	2 1	A2
MOV	bit,C	Move carry flag to direct bit	2 2	92
Program and machine control				
ACALL	addr11	Absolute subroutine call	2 2	●1addr
LCALL	addr16	Long subroutine call	3 2	12
RET		Return from subroutine	1 2	22
RET1		Return from interrupt	1 2	32
AJMP	addr11	Absolute jump	2 2	▲1addr
LJMP	addr16	Long jump	3 2	02
SJMP	rel	Short jump (relative address)	2 2	80
JMP	@A+DPTR	Jump indirect relative to the DPTR	1 2	73
JZ	rel	Jump if A is zero	2 2	60
JNZ	rel	Jump if A is not zero	2 2	70
JC	rel	Jump if carry flag is set	2 2	40
JNC	rel	Jump if no carry flag	2 2	50
JB	bit,rel	Jump if direct bit is set	3 2	20
JNB	bit,rel	Jump if direct bit is not set	3 2	30
JBC	bit,rel	Jump if direct bit is set and clear bit	3 2	10
CJNE	A,direct,rel	Compare direct to A and jump if not equal	3 2	B5
CJNE	A,#data,rel	Compare immediate to A and jump if not equal	3 2	B4
CJNE	Rr,#data,rel	Compare immed. to reg. and jump if not equal	3 2	B*
CJNE	@Ri,#data,rel	Compare immed. to ind. and jump if not equal	3 2	B6, B7
DJNZ	Rr,rel	Decrement register and jump if not zero	2 2	D*
DJNZ	direct,rel	Decrement direct and jump if not zero	3 2	D5
NOP		No operation	1 1	00

Notes to Table 1

Data addressing modes

Rr	Working register R0-R7.
direct	128 internal RAM locations and any special function register (SFR).
@Ri	Indirect internal RAM location addressed by register R0 or R1.
#data	8-bit constant included in instruction.
#data16	16-bit constant included as bytes 2 and 3 of instruction.
bit	Direct addressed bit in internal RAM or SFR.
addr16	16-bit destination address. Used by LCALL and LJMP. The branch will be anywhere within the 64 K-byte program memory address space.
addr11	11-bit destination address. Used by ACALL and AJMP. The branch will be within the same 2 K-byte page of program memory as the first byte of the following instruction.
rel	Signed (two's complement) 8-bit offset byte. Used by SJMP and all conditional jumps. Range is −128 to + 127 bytes relative to first byte of the following instruction.

Hexadecimal opcode cross-reference to Table 2

*: 8, 9, A, B, C, D, E, F.

●: 11, 31, 51, 71, 91, B1, D1, F1.

▲: 01, 21, 41, 61, 81, A1, C1, E1.

Table 2 Instruction map

first hexadecimal chacter of opcode ↓ | second hexadecimal character of opcode →

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NOP	AJMP page 0	LJMP addr16	RR A	INC A	INC dir	INC @Ri 0	1	INC Rr 0	1	2	3	4	5	6	7
1	JBC bit, addr8	ACALL page 0	LCALL addr16	RRC A	DEC A	DEC dir	DEC @Ri 0	1	DEC Rr 0	1	2	3	4	5	6	7
2	JB bit, addr8	AJMP page 1	RET	RL A	ADD A,#data	ADD A,dir	ADD A,@Ri 0	1	ADD A,Rr 0	1	2	3	4	5	6	7
3	JNB bit, addr8	ACALL page 1	RET1	RLC A	ADDC A,#data	ADDC A,dir	ADDC A,@Ri 0	1	ADDC A,Rr 0	1	2	3	4	5	6	7
4	JC addr8	AJMP page 2	ORL dir,A	ORL dir,#data	ORL A,#data	ORL A,dir	ORL A,@Ri 0	1	ORL A,Rr 0	1	2	3	4	5	6	7
5	JNC addr8	ACALL page 2	ANL dir,A	ANL dir,#data	ANL A,#data	ANL A,dir	ANL A,@Ri 0	1	ANL A,Rr 0	1	2	3	4	5	6	7
6	JZ addr8	AJMP page 3	XRL dir,A	XRL dir,#data	XRL A,#data	XRL A,dir	XRL A,@Ri 0	1	XRL A,Rr 0	1	2	3	4	5	6	7
7	JNZ addr8	ACALL page 3	ORL C,bit	JMP @A+DPTR	MOV A,#data	MOV dir,#data	MOV @Ri,#data 0	1	MOV Rr,#data 0	1	2	3	4	5	6	7
8	SJMP addr8	AJMP page 4	ANL C,bit	MOVC A,@A+PC	DIV AB	MOV dir,dir	MOV dir,@Ri 0	1	MOV dir,Rr 0	1	2	3	4	5	6	7
9	MOV DPTR, #data 16	ACALL page 4	MOV bit,C	MOVC A,@A+DPTR	SUBB A,#data	SUBB A,dir	SUBB A,@Ri 0	1	SUBB A,Rr 0	1	2	3	4	5	6	7
A	ORL C,/bit	AJMP page 5	MOV C,bit	INC DPTR	MUL AB		MOV @Ri,dir 0	1	MOV Rr,dir 0	1	2	3	4	5	6	7
B	ANL C,/bit	ACALL page 5	CPL bit	CPL C	CJNE A, #data, addr8	CJNE A,dir,addr8	CJNE @Ri,#data, addr8 0	1	CJNE Rr,#data, addr8 0	1	2	3	4	5	6	7
C	PUSH dir	AJMP page 6	CLR bit	CLR C	SWAP A	XCH A,dir	XCH A,@Ri 0	1	XCH A,Rr 0	1	2	3	4	5	6	7
D	POP dir	ACALL page 6	SETB bit	SETB C	DA A	DJNZ dir, addr8	XCHD A,@Ri 0	1	DJNZ Rr, addr8 0	1	2	3	4	5	6	7
E	MOVX A,@DPTR	AJMP page 7	MOVX A,@Ri 0	1	CLR A	MOV A,dir	MOV A,@Ri 0	1	MOV A,Rr 0	1	2	3	4	5	6	7
F	MOVX @DPTR,A	ACALL page 7	MOVX @Ri,A 0	1	CPL A	MOV dir,A	MOV @Ri,A 0	1	MOV Rr,A 0	1	2	3	4	5	6	7

RATINGS

Limiting values in accordance with the Absolute Maximum System (IEC 134)

Input voltage on any pin with respect to ground (V_{SS})	V_I	−0,5 to + 7	V
Total power dissipation	P_{tot}	max. 1	W
Input, output current	$\pm I_I, I_O$	max. 10	mA
Storage temperature range	T_{stg}	−65 to + 150	°C
Operating ambient temperature range			
MAB8031/51AH	T_{amb}	0 to + 70	°C
MAF8031/51AH	T_{amb}	−40 to + 85	°C
MAF80A31/51AH	T_{amb}	−40 to + 100	°C

D.C. CHARACTERISTICS (MAB8031/51AH)

V_{CC} = 5 V (± 10%); V_{SS} = 0 V; T_{amb} = 0 to + 70 °C; all voltages with respect to V_{SS} unless otherwise specified

parameter	symbol	min.	max.	unit	conditions
Supply current	I_{CC}	–	160	mA	
Supply current (optional)	I_{PD}	–	20	mA	
Inputs					
Input voltage LOW	V_{IL}	−0,5	0,8	V	
Input voltage HIGH all inputs except RST and XTAL 2	V_{IH}	2	V_{CC} + 0,5	V	
Input voltage HIGH to RST and XTAL 2	V_{IH1}	2,5		V	XTAL 1 to V_{SS}
Outputs					
Output voltage LOW (Ports 1, 2, 3) (note 1)	V_{OL}		0,45	V	I_{OL} = 1,6 mA
Output voltage LOW (Port 0, ALE, $\overline{PSEN}$) (note 1)	V_{OL1}		0,45	V	I_{OL1} = 3,2 mA
Output voltage HIGH (Port 1, 2, 3)	V_{OH}	2,4		V	I_{OH} = −80 μA
Output voltage HIGH (Port 0, ALE, $\overline{PSEN}$)	V_{OH1}	2,4		V	I_{OH1} = −400 μA
Input leakage current (Port 0, $\overline{EA}$)	$\pm I_{LI}$		10	μA	$V_{SS} < V_I < V_{CC}$
Input current HIGH (RST)	I_{IH1}		500	μA	$V_I = V_{CC}$−1,5 V
current logic 0 (Ports 1, 2, 3)	I_{IL}		−800	μA	V_{IL} = 0,45 V
Input current logic 0 (XTAL 2)	I_{IL2}		−2,5	mA	V_{IL} = 0,45 V; XTAL 1 to V_{SS}
Capacitance of I/O buffer	$C_{I/O}$		10	pF	f_c = 1 MHz; T_{amb} = 25 °C

D.C. CHARACTERISTICS (MAF8031/51AH; MAF80A31/51AH)

V_{CC} = 5 V (± 10%); V_{SS} = 0 V; T_{amb} = −40 to + 85 °C (MAF8031/51AH), −40 to + 100 °C (MAF80A31/51AH); all voltages with respect to V_{SS} unless otherwise specified

parameter	symbol	min.	max.	unit	conditions
Supply current	I_{CC}	–	175	mA	
Inputs					
Input voltage LOW	V_{IL}	−0,5	0,8	V	
Input voltage HIGH all inputs except RST and XTAL 2	V_{IH}	2,2	V_{CC} + 0,5	V	
Input voltage HIGH to RST and XTAL 2	V_{IH1}	2,7		V	XTAL 1 to V_{SS}
Outputs					
Output voltage LOW (Ports 1, 2, 3) (note 1)	V_{OL}		0,45	V	I_{OL} = 1,2 mA
Output voltage LOW (Port 0, ALE, $\overline{PSEN}$) (note 1)	V_{OL1}		0,45	V	I_{OL1} = 2,4 mA
Output voltage HIGH (Port 1, 2, 3)	V_{OH}	2,4		V	I_{OH} = −50 μA
Output voltage HIGH (Port 0, ALE $\overline{PSEN}$)	V_{OH1}	2,4		V	I_{OH1} = −360 μA
Input leakage current (Port 0, $\overline{EA}$)	$\pm I_{LI}$		10	μA	$V_{SS} < V_I < V_{CC}$
Input current HIGH (RST)	I_{IH1}		500	μA	$V_I = V_{CC}-1,5$ V
current logic 0 (Ports 1, 2, 3)	I_{IL}		−800	μA	V_{IL} = 0,45 V
Input current logic 0 (XTAL 2)	I_{IL2}		−3,0	mA	V_{IL} = 0,45 V; XTAL 1 to V_{SS}
Capacitance of I/O buffer	$C_{I/O}$		10	pF	f_c = 1 MHz; T_{amb} = 25 °C

Note 1

V_{OL} is degraded when the MAB8051AH rapidly discharges external capacitance.
This a.c. noise is most pronounced during emission of address data. When using external memory, locate the latch or buffer as close to the MAB8051AH as possible.

datum	emitting ports	time slot interval	degraded I/O lines	V_{OL} (max.)
address	P2, P0	TS3, TS9	P1, P3	0,8 V
write data	P0	TS6	P1, P3, ALE	0,8 V

A.C. CHARACTERISTICS (note 1)

V_{CC} = 5 V ± 10%; V_{SS} = 0 V; T_{amb} = 0 to + 70 °C; C_L = 100 pF (Port 0, ALE and $\overline{PSEN}$); C_L = 80 pF all other outputs unless otherwise specified (see waveforms Figs 7, 8 and 9).

parameter	symbol	8 MHz		10 MHz		12 MHz		variable clock (note 2)		unit
		min.	max.	min.	max.	min.	max.	min.	max.	
Program memory										
ALE pulse duration	t_{LL}	127	–	160	–	127	–	$2t_{CK}-40$	–	ns
Address set-up time to ALE	t_{AL}	85	–	60	–	43	–	$t_{CK}-40$	–	ns
Address hold time after ALE	t_{LA}	90	–	65	–	48	–	$t_{CK}-35$	–	ns
Time from ALE to valid instruction input	t_{LIV}	–	400	–	300	–	233	–	$4t_{CK}-100$	ns
Time from ALE to control pulse $\overline{PSEN}$	t_{LC}	100	–	75	–	58	–	$t_{CK}-25$	–	ns
Control pulse duration $\overline{PSEN}$	t_{CC}	340	–	265	–	215	–	$3t_{CK}-35$	–	ns
Time from $\overline{PSEN}$ to valid instruction input	t_{CIV}	–	250	–	175	–	125	–	$3t_{CK}-125$	ns
Input instruction hold time after $\overline{PSEN}$	t_{CI}	0	–	0	–	0	–	0	–	ns
Input instruction float delay after $\overline{PSEN}$ (note 3)	t_{CIF}	–	105	–	80	–	63	–	$t_{CK}-20$	ns
Address valid after $\overline{PSEN}$ (note 3)	t_{AC}	117	–	92	–	75	–	$t_{CK}-8$	–	ns
Address to valid instruction input	t_{AIV}	–	510	–	385	–	302	–	$5t_{CK}-115$	ns
Address float time to $\overline{PSEN}$	t_{AFC}	−12	–	−12	–	−12	–	−12	–	ns

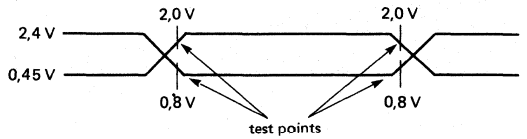
parameter	symbol	8 MHz		10 MHz		12 MHz		variable clock (note 2)		unit
		min.	max.	min.	max.	min.	max.	min.	max.	
External data memory										
$\overline{RD}$ pulse duration	t_{RR}	650	–	500	–	400	–	$6t_{CK}-100$	–	ns
$\overline{WR}$ pulse duration	t_{WW}	650	–	500	–	400	–	$6t_{CK}-100$	–	ns
Address hold time after ALE	t_{LA}	90	–	65	–	48	–	$t_{CK}-35$	–	ns
RD to valid data input	t_{RD}	–	460	–	335	–	250	–	$5t_{CK}-165$	ns
Data hold time after $\overline{RD}$	t_{DR}	0	–	0	–	0	–	0	–	ns
Data float delay after $\overline{RD}$	t_{DFR}	–	180	–	130	–	97	–	$2t_{CK}-70$	ns
Time from ALE to valid data input	t_{LD}	–	850	–	650	–	517	–	$8t_{CK}-150$	ns
Address to valid data input	t_{AD}	–	960	–	735	–	585	–	$9t_{CK}-165$	ns
Time from ALE to $\overline{RD}$ or $\overline{WR}$	t_{LW}	325	425	250	350	200	300	$3t_{CK}-50$	$3t_{CK}+50$	ns
Time from address to $\overline{RD}$ or $\overline{WR}$	t_{AW}	370	–	270	–	203	–	$4t_{CK}-130$	–	ns
Time from $\overline{RD}$ or $\overline{WR}$ HIGH to ALE HIGH	t_{WHLH}	85	165	60	140	43	123	$t_{CK}-40$	$t_{CK}+40$	ns
Data valid to $\overline{WR}$ transition	t_{DWX}	65	–	40	–	23	–	$t_{CK}-60$	–	ns
Data set-up time before $\overline{WR}$	t_{DW}	725	–	550	–	433	–	$7t_{CK}-150$	–	ns
Data hold time after $\overline{WR}$	t_{WD}	75	–	50	–	33	–	$t_{CK}-50$	–	ns
Address float delay after $\overline{RD}$	t_{AFR}	–	12	–	12	–	12	–	12	ns

Notes to the a.c. characteristics

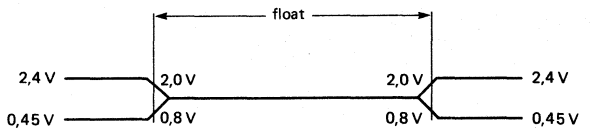
1. T_{amb} for MAF8031AH/MAF8051AH is –40 to 85 °C;
   for MAF80A31AH/MAF80A51AH is –40 to 100 °C.
   The clock frequencies applicable to each device are as follows:

device	clock frequency (MHz)			
	8	10	12	15
MAB8031AH	yes	yes	yes	yes
MAB8051AH	yes	yes	yes	no
MAF8031AH	yes	yes	yes	no
MAF8051AH	yes	yes	yes	no
MAF80A31AH	yes	yes	no	no
MAF80A51AH	yes	yes	no	no

2. $1/t_{CK}$ = 3,5 to 15 MHz (see Fig. 11 and Table 3).
3. Interfacing the MAB8051AH to devices with float times up to 75 ns is permitted. This limited bus contention will not cause damage to port 0 drivers.



(a)



(b)

A.C. testing inputs are driven at 2,4 V for a logic 1 and 0,45 V for a logic 0. Timing measurements are taken at 2,0 V for a logic 1 and 0,8 V for logic 0. The float state is defined as the point at which a Port 0 pin sinks 3,2 mA or sources 400 μA at the voltage test levels.

Fig. 10 A.C. testing input, output waveform (a) and float waveform (b).

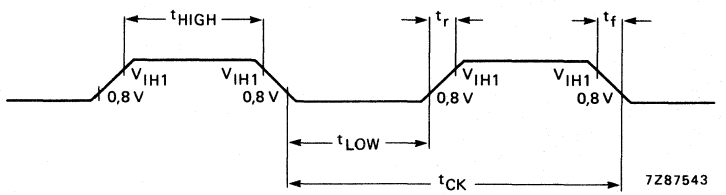


Fig. 11 External clock drive XTAL 2 (see Table 3).

Table 3 External clock drive XTAL 2 (see Fig. 11)

parameter	symbol	variable clock (f = 3,5 to 15 MHz)		unit
		min.	max.	
oscillator clock period	t_{CK}	66,6	286	ns
HIGH time	t_{HIGH}	20	$t_{CK}-t_{LOW}$	ns
LOW time	t_{LOW}	20	$t_{CK}-t_{HIGH}$	ns
rise time	t_r	–	20	ns
fall time	t_f	–	20	ns

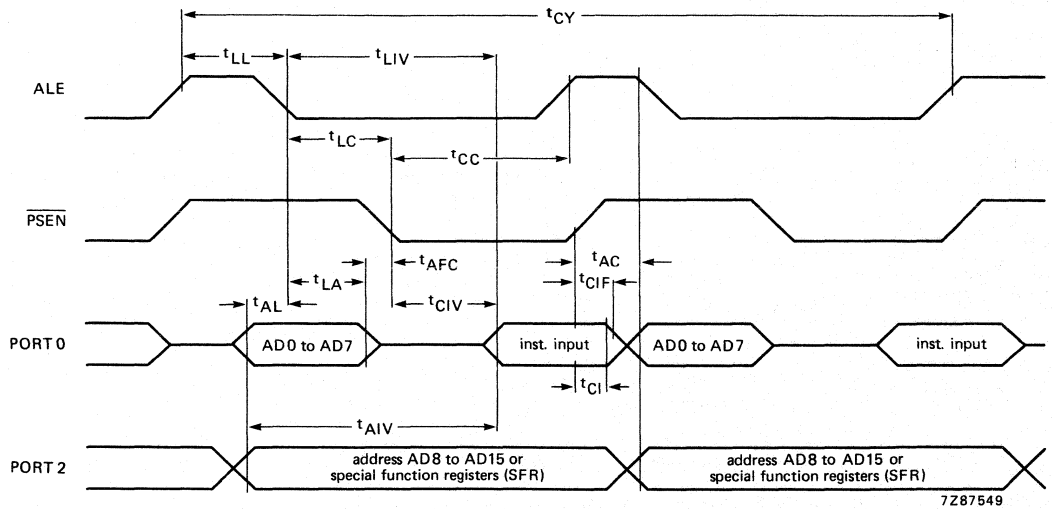


Fig. 12 Read from program memory.

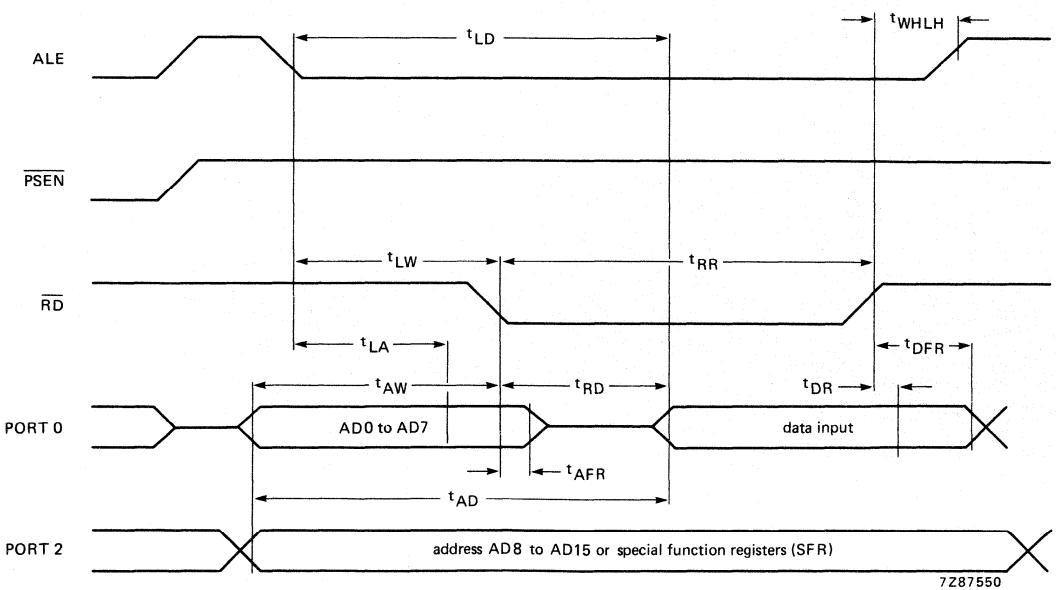


Fig. 13 Read from data memory.

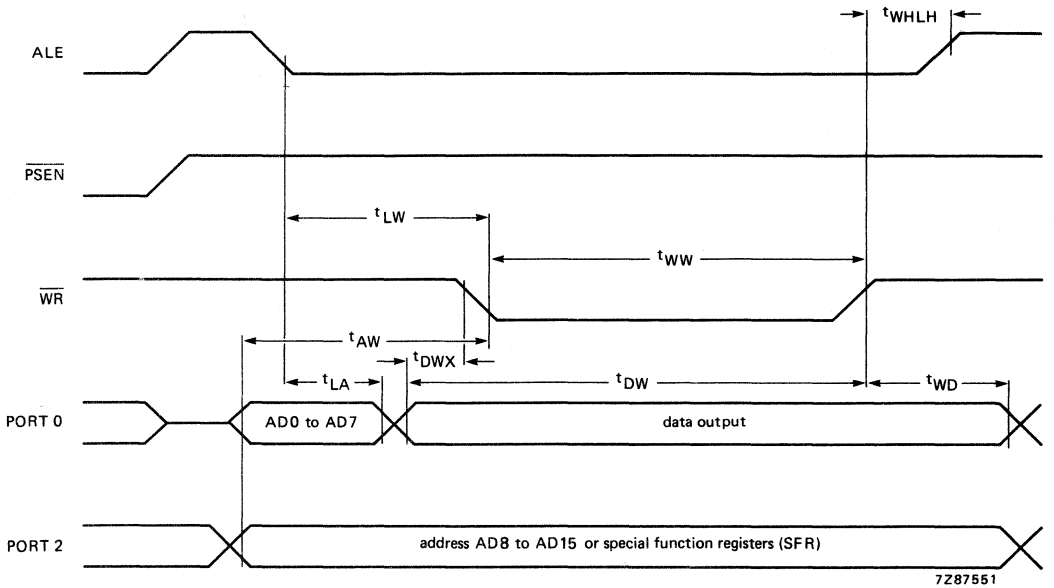


Fig. 14 Write to data memory.

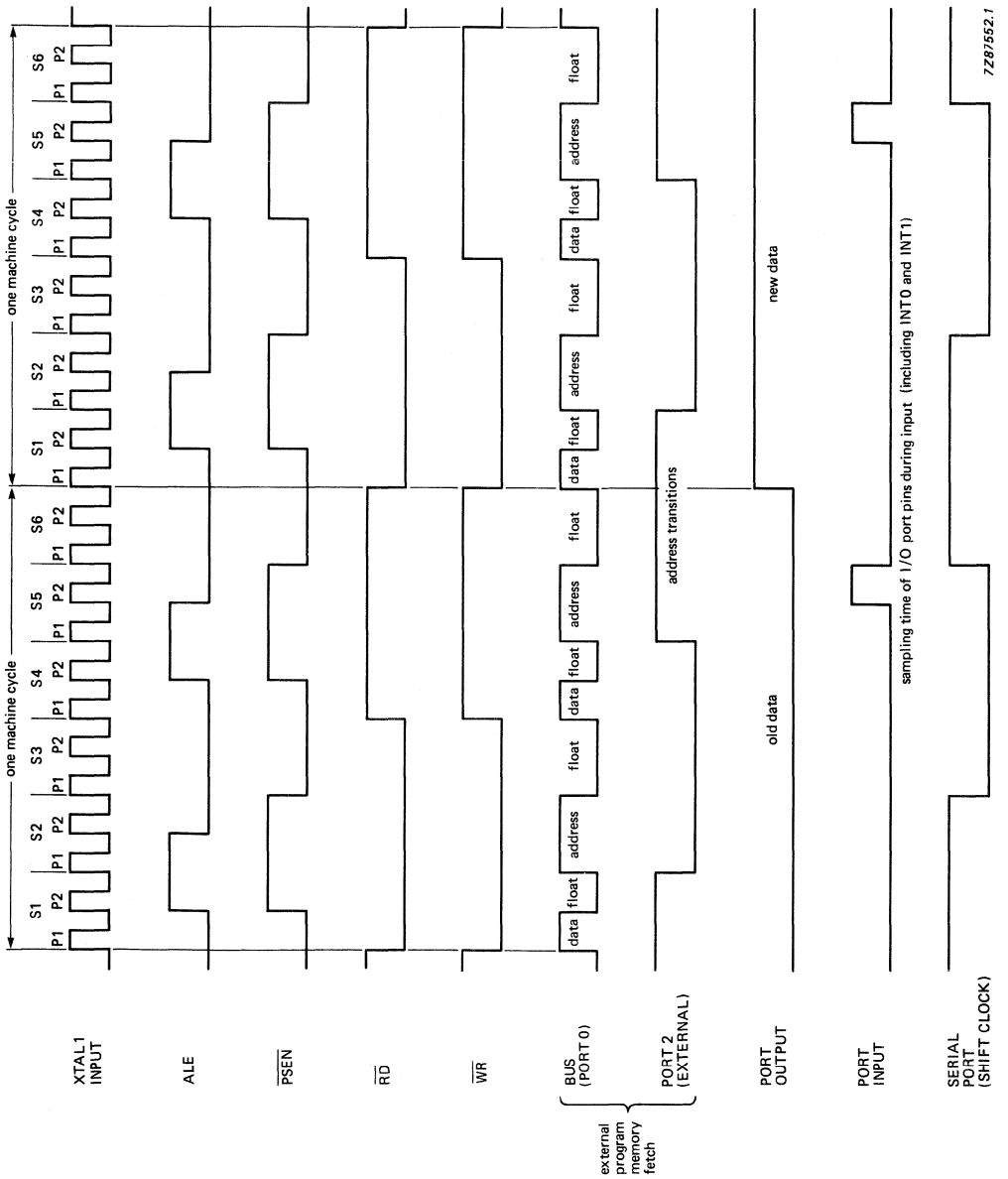


Fig. 15 Instruction cycle timing.

SINGLE-CHIP 8-BIT MICROCONTROLLER

DESCRIPTION

The MAB8052AH is a member of the MAB8051AH family with a higher performance. This single-chip 8-bit microcontroller is manufactured in an advanced 2 μ NMOS process. For this version the following members exist:

- MAB8032AH: ROM-less version of the MAB8052AH
- MAB8052AH: 8 K bytes mask programmable ROM, 256 bytes RAM

Both types are available in 12 MHz versions. In the following, the generic term "MAB8052AH" is used to refer to both family members.

The device provides hardware features, architectural enhancements and new instructions to function as a controller for applications requiring up to 64 K bytes of program memory and/or up to 64 K bytes of data storage.

The MAB8052AH contains a non-volatile 8 K x 8 read-only program memory (not ROM-less version); a volatile 256 x 8 read/write data memory; 32 I/O lines; three 16-bit timer/event counters; a six-source, two-priority-level, nested interrupt structure; a serial I/O port for either multi-processor communications, I/O expansion, or full duplex UART; and on-chip oscillator and timing circuits. For systems that require extra capability, the MAB8052AH can be expanded using standard TTL compatible memories and logic.

The device also functions as an arithmetic processor having facilities for both binary and BCD arithmetic plus bit-handling capabilities. The instruction set consists of 255 instructions; 44% one-byte, 41% two-byte and 15% three-byte. With a 12 MHz crystal, 58% of the instructions are executed in 1 μs and 40% in 2 μs. Multiply and divide instructions require 4 μs.
For further detailed information see users manual 'single-chip 8-bit microcontrollers'.

Features

- 8 K x 8 ROM (8052AH only), 256 x 8 RAM
- Four 8-bit ports, 32 I/O lines
- Three 16-bit timer/event counters
- Full-duplex serial port
- External memory expandable to 128 K
- Boolean processing
- 218 bit-addressable locations
- On-chip oscillator
- Six-source interrupt structure with two priority levels
- 58% of instructions executed in 1 μs; multiply and divide in 4 μs
- Enhanced architecture with:
  - non-page-oriented instructions
  - direct addressing
  - four 8-bit register banks
  - stack depth up to 128-bytes
  - multiply, divide, substract and compare
- Upward compatible with MAB8031AH/8051AH
- Extended temperature range (−40 to +100 °C) in preparation

PACKAGE OUTLINE

MAB8032/52AHP: 40-lead DIL, plastic (SOT-129).

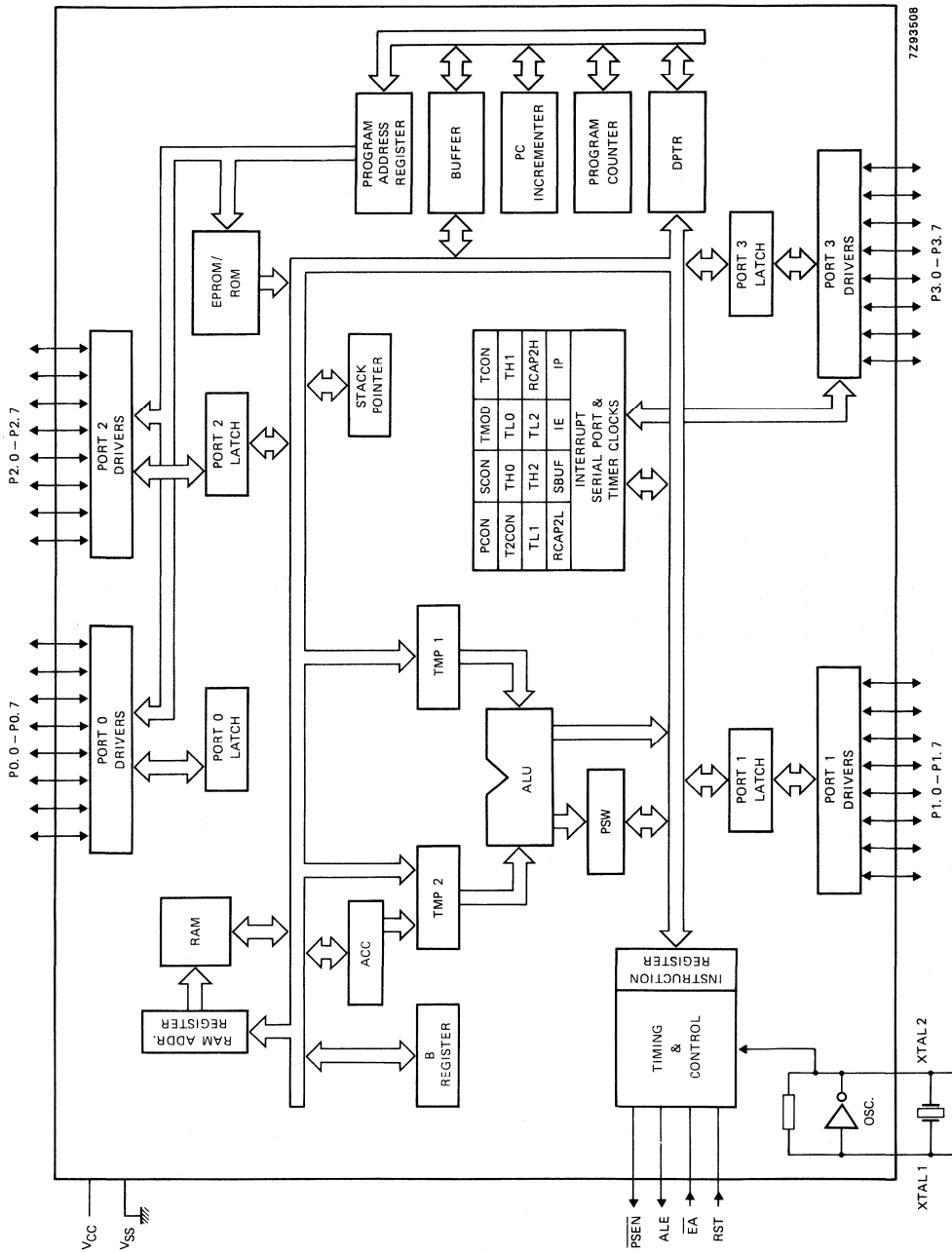


Fig. 1 Block diagram.

DEVELOPMENT DATA

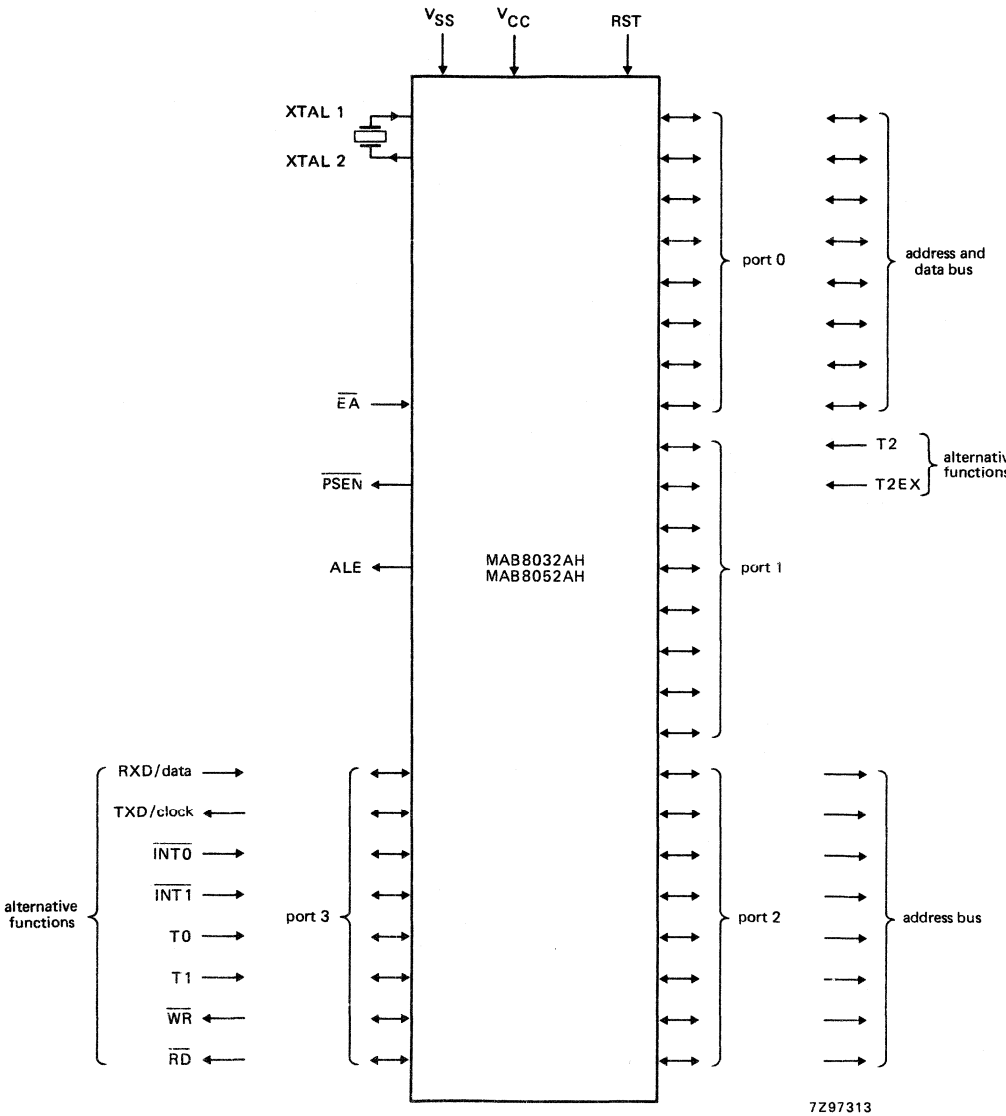


Fig. 2 Functional diagram.

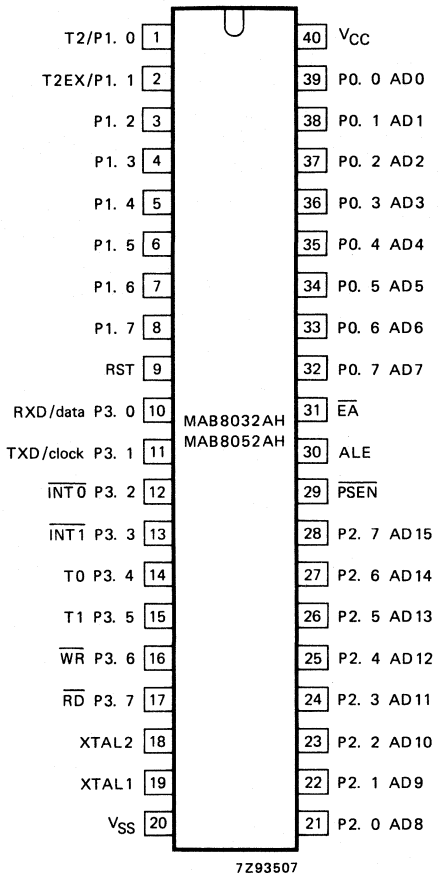


Fig. 3 Pinning diagram.

DEVELOPMENT DATA

PINNING

1-8	P1.0-P1.7	**Port 1:** 8-bit quasi-bidirectional I/O port. It receives the low-order address byte during program verification. Port 1 can sink/source four LS TTL (= 1TTL) inputs. It can drive MOS inputs without external pull-ups. Pins 1 and 2 also supply alternative functions T2 and T2EX. T2 is the counter trigger input for timer 2; T2EX the external input to timer 2. Operation of the alternative functions is determined by the relevant output latch programmed to logic 1.
9		**RST:** a high level on this pin for two machine cycles while the oscillator is running resets the device. An internal pulldown permits Power-On reset using only a capacitor connected to V_{CC} (see Fig. 18).
10-17	P3.0-P3.7	**Port 3**: 8-bit quasi-bidirectional I/O port with internal pull-ups. It also serves the following alternative functions:

Port pin	*Alternative function*
P3.0	**RXD/data**: serial port receiver data input (asynchronous) or data input/output (synchronous)
P3.1	**TXD/clock**: serial port transmitter data output (asynchronous) or clock output (synchronous)
P3.2	**$\overline{\text{INT0}}$**: external interrupt 0 or gate control input for timer/event counter 0
P3.3	**$\overline{\text{INT1}}$**: external interrupt 1 or gate control input for timer/event counter 1
P3.4	**T0** : external input for timer/event counter 0
P3.5	**T1** : external input for timer/event counter 1
P3.6	**$\overline{\text{WR}}$**: external data memory write strobe
P3.7	**$\overline{\text{RD}}$** : external data memory read strobe

Operation of an alternative function is determined by the relevant output latch programmed to logic 1. Port 3 can sink/source four LS TTL inputs. It can drive MOS inputs without external pull-ups.

18	XTAL 2	**Crystal input 2**: output of the inverting amplifier that forms the oscillator, and input to the internal clock generator. Receives the external oscillator signal when an external oscillator is used (see figures 15 and 16).
19	XTAL 1	**Crystal input 1**: input to the inverting amplifier that forms the oscillator. Connected to V_{SS} when an external oscillator is used.
20	V_{SS}	**Ground**: circuit earth potential.
21-28	P2.0-P2.7	**Port 2**: 8-bit quais-bidirectional I/O port with internal pull-ups. It emits the high-order address byte when accessing external memory. It also receives the high-order address bits and control signals during program verification. Port 2 can sink/source four LS TTL inputs. It can drive MOS inputs without external pull-ups.
29	$\overline{\text{PSEN}}$	**Program Store Enable output**: read strobe to the external Program Memory. It is activated twice each machine cycle during fetches from external Program Memory. When executing out of external Program Memory two activations of $\overline{\text{PSEN}}$ are skipped during each access to external Data Memory. $\overline{\text{PSEN}}$ is not activated (remains HIGH) during fetches from internal Program Memory.

PINNING (continued)

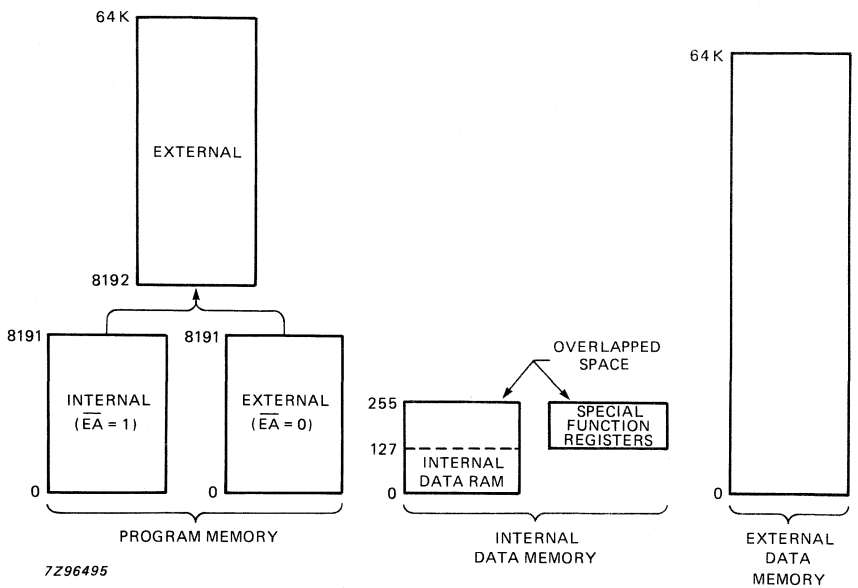
30	ALE	**Address Latch Enable output**: latches the low byte of the address during accesses to external memory in normal operation. It is activated every six oscillator periods except during an external data memory addess.
31	$\overline{EA}$	When $\overline{EA}$ is held at a TTL high level the CPU executes out of the internal Program Memory (ROM), provided the Program Counter is less than 8192. When $\overline{EA}$ is held at a TTL low level the CPU executes out of external Program Memory. $\overline{EA}$ does not float.
32-39	P0.7-P0.0	**Port 0**: 8-bit open drain bidirectional I/O port. It is also the multiplexed low-order address and data bus during accesses to external memory (during these accesses it activates internall pull-ups). It also outputs instruction bytes during program verification. (External pull-ups are required during program verification). Port 0 can sink (and in bus operations can source) eight LS TTL inputs.
40	V_{CC}	**Power Supply**: + 5 V power supply pin during normal operation.

As the MAB8052AH is based on the MAB8051 the following pages are a description of the additional features.

MEMORIES (see Fig. 4)

ROM: 8 K bytes mask programmable. An external ROM is accessed up to 64 K to accommodate an address higher than 8191.

RAM: 256 bytes (on-chip) plus special function registers detailed in Table 1.



Where:

The internal data memory locations are addressable direct/indirect as follows:

location	addressed
RAM 0 to 127	direct and indirect
RAM 128 to 255	indirect only
SFR 128 to 255	direct only

Fig. 4 Memory map.

SPECIAL FUNCTION REGISTERS

Table 1 Special Function Registers (SFR)

symbol	name	address	contents after reset
ACC*	accumulator	0E0H	00H
B*	B register	0F0H	00H
PSW*	program status word	0D0H	00H
SP	stack pointer	81H	07H
DPTR	data pointer		
	DPH	83H	0000H
	DPL	82H	0000H
P0*	port 0	80H	0FFH
P1*	port 1	90H	0FFH
P2*	port 2	0A0H	0FFH
P3*	port 3	0B0H	0FFH
IP*	interrupt priority control	0B8H	XX000000B
IE*	interrupt enable control	0A8H	0X000000B
TMOD	timer/counter mode control	89H	00H
T2CON*	timer/counter 2 control	0C8H	00H
TCON	timer/counter control	88H	00H
TH0	timer/counter 0 (high byte)	8CH	00H
TL0	timer/counter 0 (low byte)	8AH	00H
TH1	timer/counter 1 (high byte)	8DH	00H
TL1	timer/counter 1 (low byte)	8BH	00H
TH2	timer/counter 2 (high byte)	0CDH	00H
TL2	timer/counter 2 (low byte)	0CCH	00H
RCAP2H	timer/counter 2 capture register (high byte)	0CBH	00H
RCAP2L	timer/counter 2 capture register (low byte)	0CAH	00H
SCON*	serial control	98H	00H
SBUF	serial data buffer	99H	indeterminate
PCON	power control	87H	0XXXXXXXB

Where

H = Hexadecimal
B = Binary

* Registers are both byte and bit addressable.

T2CON

Special Function Register T2CON is the timer/counter 2 control register. T2CON contains the control and status bits shown in Fig. 5 and described in Table 2.

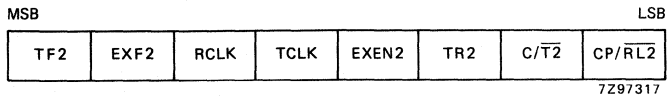


Fig. 5 Timer/counter 2 control register (T2CON).

Table 2 T2CON: Control and status bits

symbol	position	name	operation
TF2	T2CON.7	Timer 2 overflow flag	Set by a Timer 2 overflow and must be cleared by software. TF2 will not be set when either RCLK = 1 or TCLK = 1.
EXF2	T2CON.6	Timer 2 external flag	Set when either a capture or reload is caused by a negative transition on T2EX and EXEN2 = 1. When Timer 2 interrupt is enabled, EXF2 = 1 will cause the CPU to vector to the Timer 2 interrupt routine. EXF2 must be cleared by software.
RCLK	T2CON.5	Receive clock flag	When set, causes the serial port to use Timer 2 overflow pulses for its receive clock in modes 1 and 3. RCLK = 0 causes Timer 1 overflow to be used for the receive clock.
TCLK	T2CON.4	Transmit clock flag	When set causes the serial port to use Timer 2 overflow pulses for its transmit clock in modes 1 and 3. TCLK = 0 causes Timer 1 overflows to be used for the transmit clock.
EXEN2	T2CON.3	Timer 2 external enable flag	When set allows a capture or reload to occur as a result of a negative transition on T2EX if Timer 2 is not being used to clock the serial port. EXEN2 = 0 causes Timer 2 to ignore events at T2EX.
TR2	T2CON.2	Start/stop control for timer 2	A logic 1 starts the timer.
C/$\overline{T2}$	T2CON.1	Time or counter select (timer 2)	0 = internal timer (OSC/12) 1 = external event counter (falling edge triggered)
CP/RL2	T2CON.0	Capture/reload flag	When set capture will occur on negative transitions at T2EX if EXEN2 = 1. When cleared automatic reloads will occur either with Timer 2 overflows or negative transitions at T2EX when EXEN2 = 1. When either RCLK = 1 or TCLK = 1, this bit is ignored and the timer is forced to automatic-reload on Timer 2 overflow.

SPECIAL FUNCTION REGISTERS (continued)

TH2; TL2

Timer/counter 2 (high byte); Timer/counter 2 (low byte) are the register pair for the 16-bit counting of timer/counter 2.

RCAP2H; RCAP2L

Timer/counter 2 capture (high byte); Timer/counter 2 capture (low byte) are the register pair for the "capture mode" of Timer 2. In this mode, a signal at T2EX initiates a copy of TH2 and TL2 into RCAP2H and RCAP2L. Timer 2 also has an automatic-reload mode. RCAP2H and RCAP2L hold the reload value for this mode.

T2CON

Timer 2 is 16-bit and operates the same as Timer 0 and Timer 1. The operating mode is selected by the T2CON register. Table 3 shows the operating modes of Timer 2.

Table 3 Timer 2 operating modes.

RCLK + TCLK	CP/$\overline{RL2}$	TR2	mode
0	0	1	16-bit automatic-reload
0	1	1	16-bit capture
1	X	1	baud rate generator
X	X	0	off

Where

X = don't care.

Capture mode (see Fig. 6)

In the capture mode there are two options which are selected by bit EXEN2 in T2CON. If EXEN2 = 0, then Timer 2 is a 16-bit timer or counter which on overflow will set bit TF2 (Timer 2 overflow bit). TF2 can be used to generate an interrupt. If EXEN2 = 1, Timer 2 operates as for EXEN2 = 0 with the added feature that a 1-to-0 transition at the external input T2EX causes the current value in the Timer 2 registers (TL2 and TH2) to be captured into registers RCAP2L and RCAP2H respectively. The 1-to-0 transition of T2EX also causes bit EXF2 in T2CON to be set. EXF2 can be used to generate an interrupt.

Automatic-reload mode (see Fig. 7)

In the automatic-reload mode there are two options which are selected by bit EXEN2 in T2CON. If EXEN2 = 0, when Timer 2 overflows it sets TF2 and causes the Timer 2 registers to be reloaded with the 16-bit value in registers RCAP2L and RCAP2H, which are preset by software. If EXEN2 = 1, Timer 2 operates as for EXEN2 = 0 with the added feature that a 1-to-0 transition at the external input T2EX triggers the 16-bit reload and sets EXF2.

Baud rate generator mode (see Fig. 8)

The baud rate generator mode is selected by RCLK = 1 and/or TCLK = 1 in T2CON. Thus baud rates for transmit and receive can be simultaneously different. This mode is described in conjunction with the serial port.

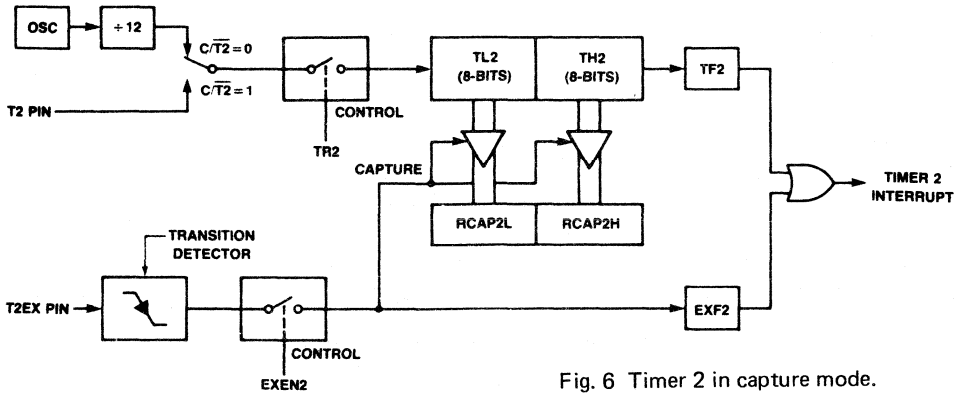


Fig. 6 Timer 2 in capture mode.

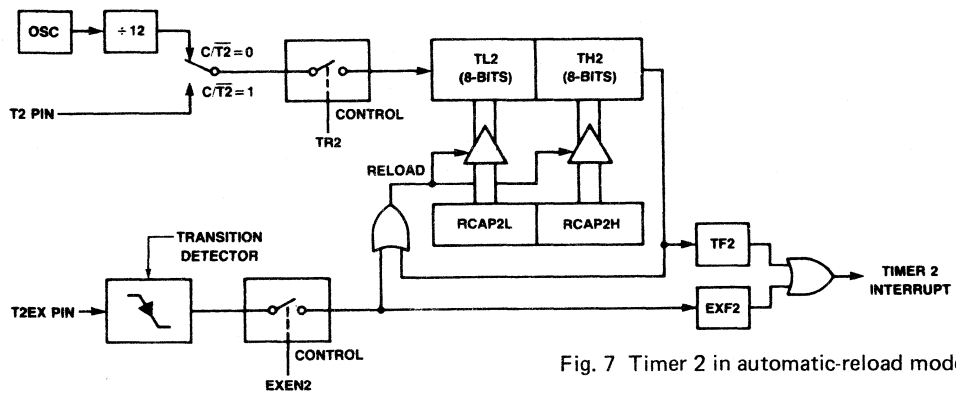


Fig. 7 Timer 2 in automatic-reload mode.

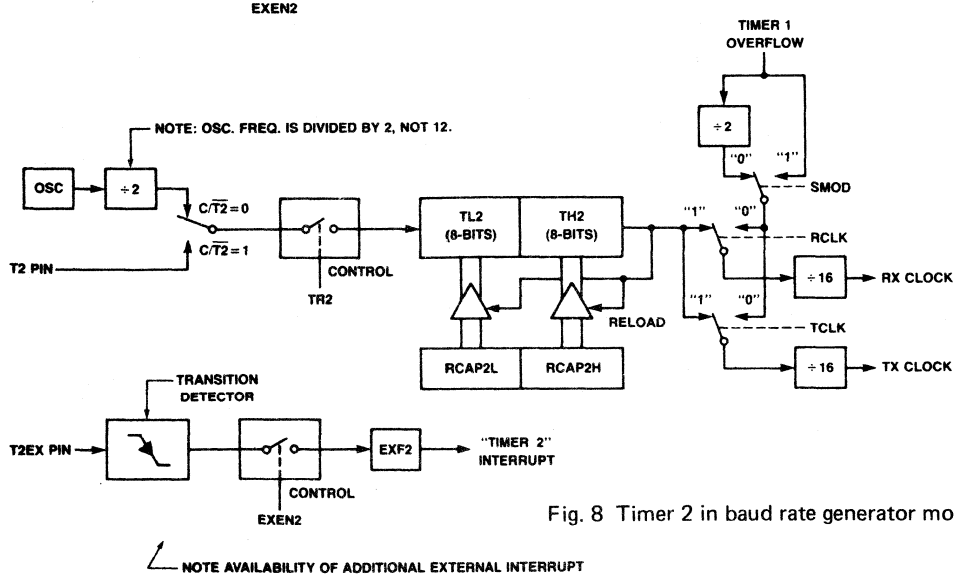


Fig. 8 Timer 2 in baud rate generator mode.

DEVELOPMENT DATA

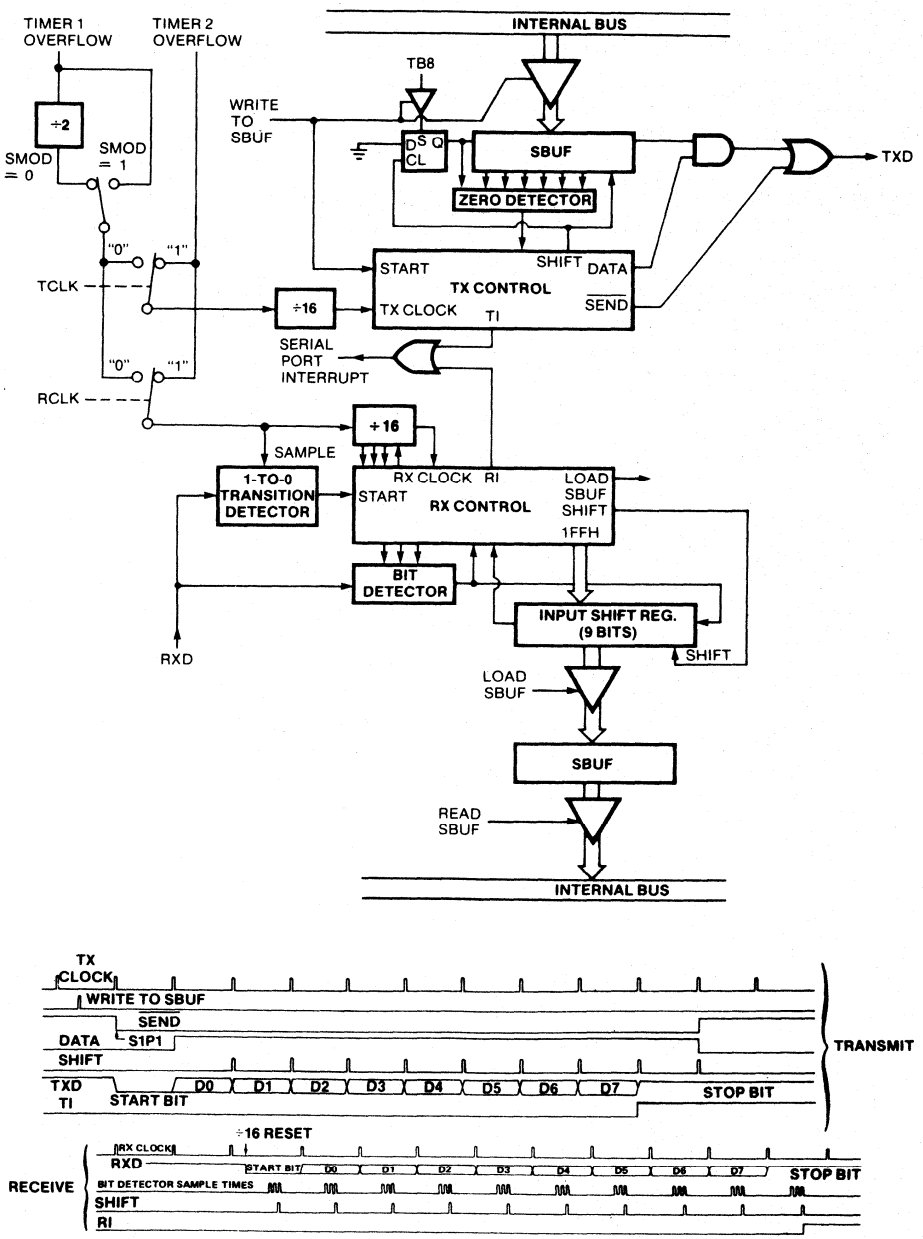


Fig. 9 The serial port in mode 1.

DEVELOPMENT DATA

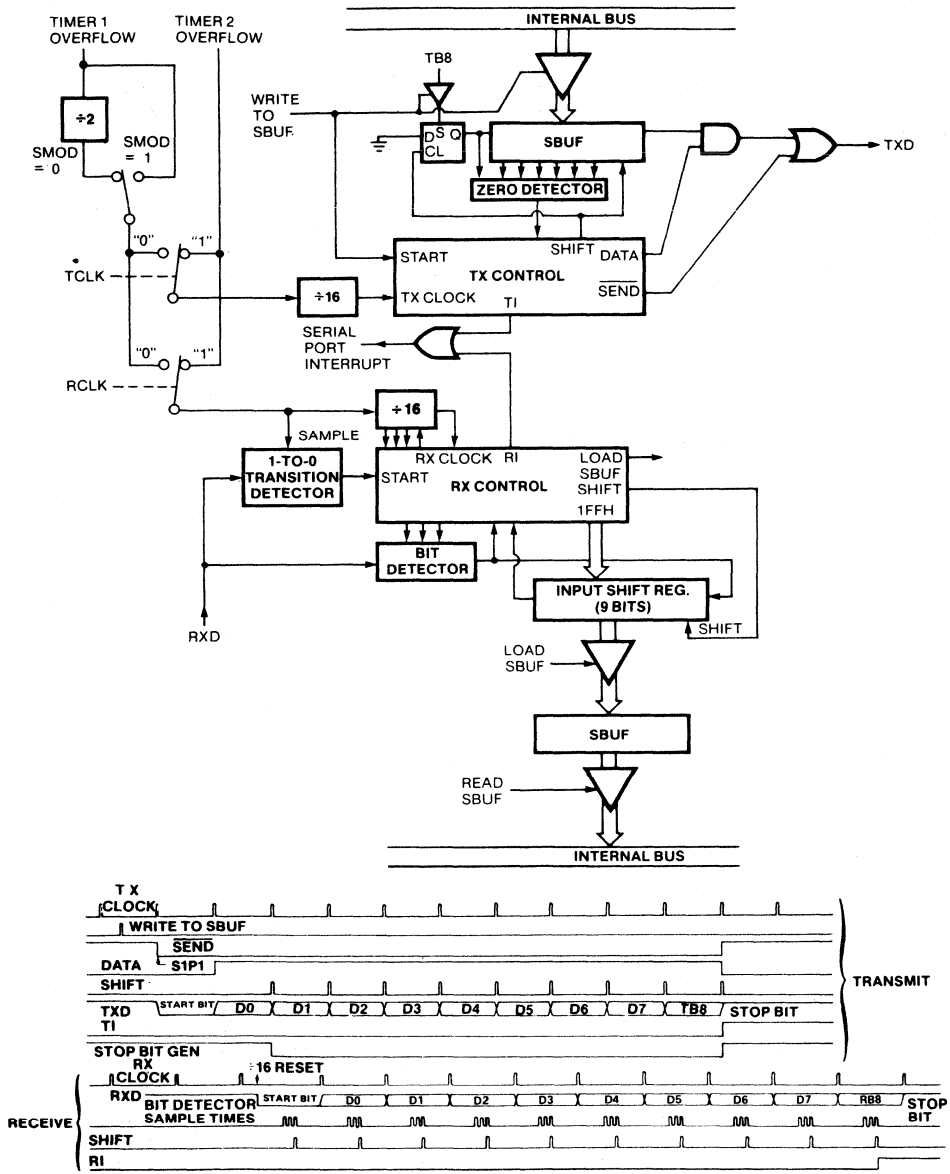


Fig. 10 The serial port in mode 3.

INTERRUPTS

The MAB8052AH has 6 interrupts as shown in Fig. 11.

$\overline{INT0}$; $\overline{INT1}$

The external interrupts $\overline{INT0}$ and $\overline{INT1}$ can be level-activated or transition-activated dependent on bits IT0 and IT1 in register TCON. The flags that generate these interrupts are bits IE0 and IE1.

When a transition-activated external interrupt is generated, the interrupt flag is cleared by the on-chip hardware when the CPU transfers control to the service routine. If the interrupt is level-activated, the request flag is controlled by the external requesting source.

Timer 0; Timer 1

The Timer 0 and Timer 1 interrupts are generated by TF0 and TF1, which are set by an overflow in their respective timer/counter registers. This is not applicable for Timer 0 in mode 3. When a timer interrupt is generated, the interrupt flag is cleared by the on-chip hardware when the CPU transfers control to the service routine.

Timer 2

The interrupt of Timer 2 is generated by the logical OR of TF2 and EXF2. The service routine determines whether it was TF2 or EXF2 that generated the interrupt, and the bits are cleared by software.

Serial port interrupt

The serial port interrupt is generated by the logical OR of R1 and T1 flags. These flags are not cleared by the on-chip hardware when the CPU transfers control to the service routine. The service routine determines whether it was R1 or T1 that generated the interrupt, and the bit is cleared by software.

All of the bits that generate interrupts can be set or cleared by software, with the same result as being set or cleared by hardware. Thus interrupts can be generated, or pending interrupts cancelled, by software.

Each of the interrupt sources can be individually enabled or disabled by setting or clearing a bit in Special Function Register IE (see Fig. 12 and Table 4).

Register IE contains a global disable bit (EA) which disables all interrupts simultaneously.

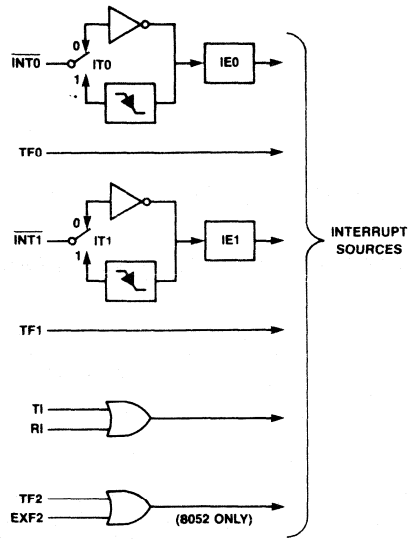


Fig. 11 Interrupt sources.

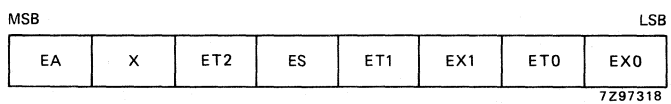


Fig. 12 Interrupt enable register (IE).

Table 4 IE: Control and status bits

symbol	position	function
EA	IE.7	Disables all interrupts. If EA = 0 no interrupt will be acknowledged. If EA = 1 each interrupt source is individually enabled or disabled by setting or clearing its enable bit.
–	IE.6	Reserved.
ET2	IE.5	Enables or disables the Timer 2 overflow or capture interrupt. If ET2 = 0 the Timer 2 interrupt is disabled.
ES	IE.4	Enables or disables the Serial Port interrupt. If ES = 0 the Serial Port interrupt is disabled.
ET1	IE.3	Enables or disables the Timer 1 overflow interrupt. If ET1 = 0 the Timer 1 interrupt is disabled.
EX1	IE.2	Enables or disables external interrupt 1. If EX1 = 0 the external interrupt is disabled.
ET0	IE.1	Enables or disables the Timer 0 overflow interrupt. If ET0 = 0 the Timer 0 interrupt is disabled.
EX0	IE.0	Enables or disables external interrupt.0. If EX0 = 0 the external interrupt is disabled.

DEVELOPMENT DATA

INTERRUPTS (continued)

Priority level structure (see Fig. 13 and Table 5)

Each interrupt source can be programmed to one of two priority levels by setting or clearing a bit in Special Function Register IP. A low-priority interrupt can be interrupted by a high-priority interrupt, but not by another low-priority interrupt. A high-priority interrupt cannot be interrupted by any other interrupt source.

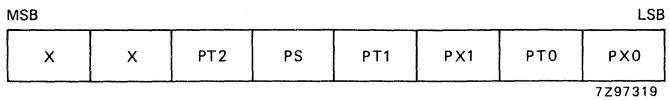


Fig. 13 Interrupt priority register (IP).

Table 5 IP: Control and status bits

symbol	position	function
–	IP.7	Reserved.
–	IP.6	Reserved.
PT2	IP.5	Defines the Timer 2 interrupt priority level. PT2 = 1 programs it to the higher priority level.
PS	IP.4	Defines the Serial Port interrupt priority level. PS1 = 1 programs it to the higher priority level.
PT1	IP.3	Defines the Timer 1 interrupt priority level. PT1 = 1 programs it to the higher priority level.
PX1	IP.2	Defines the external interrupt 1 priority level. PX1 = 1 programs it to the higher priority level.
PT0	IP.1	Defines the Timer 0 interrupt priority level. PT0 = 1 programs it to the higher priority level.
PX0	IP.0	Defines the external interrupt 0 priority level. PX0 = 1 programs it to the higher priority level.

The interrupt response timing is shown in Fig. 14. The fastest possible response occurs when C2 is the final cycle of an instruction other than "return from interrupt" instruction (RET1), or an access to registers IE or IP.

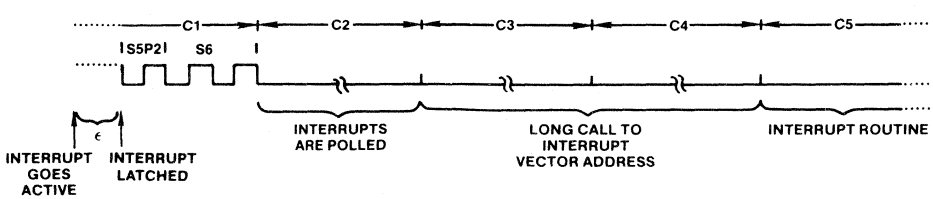


Fig. 14 Interrupt response timing diagram.

Priority level structure (continued)

If two requests of different priority levels are received simultaneously, the higher priority request is serviced. If requests of the *same* priority level are received simultaneously, an internal polling sequence determines which request is serviced. This second priority structure determined by the polling sequence is detailed in Table 6.

Table 6 Vector addresses by interrupt depending on interrupt source and priority

number	source	priority within level	vector address
1	IE0	(highest)	0003H
2	TF0		000BH
3	IE1		0013H
4	TF1		001BH
5	R1 + T1		0023H
6	TF2 + EXF2	(lowest)	002BH

OSCILLATOR CIRCUITRY

The oscillator circuitry of the MAB8052AH is a single-stage inverting amplifier in a Pierce oscillator configuration. The circuitry has a combination of depletion and enhancement mode MOS FETs to produce the inverting characteristics, and passive components. Either a crystal or ceramic resonator can be used as the feedback element to complete the oscillator circuitry. XTAL1, pin 19, is the high gain amplifier input, and XTAL2, pin 18, is the output (see Fig. 15).

To drive the MAB8052AH externally, XTAL1 should be connected to ground and XTAL2 driven from an external source (see Fig. 16).

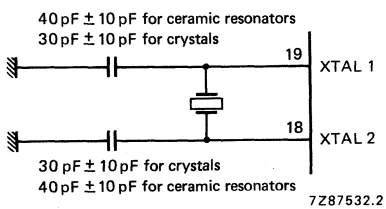


Fig. 15 MAB8052AH oscillator circuit.

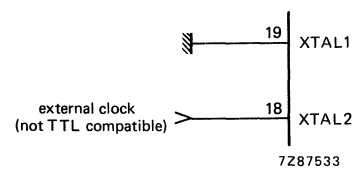


Fig. 16 Driving the MAB8052AH from an external source.

RESET CIRCUITRY

The reset circuitry for the MAB8052AH is connected to the reset pin, RST, as shown in Fig. 17. A Schmitt trigger is used at the input for noise rejection. The output of the Schmitt trigger is sampled by the reset circuitry every machine cycle.

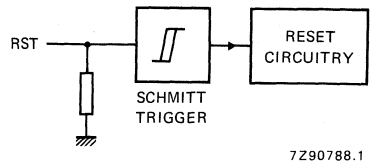


Fig. 17 Reset configuration at RST.

A reset is accomplished by holding the RST pin HIGH for at least two machine cycles (24 oscillator periods), while the oscillator is running. The CPU responds by executing an internal reset. It also configures the ALE and $\overline{PSEN}$ pins as inputs. (They are quasi-bidirectional.) The internal reset is executed during the second cycle in which RST is HIGH and is repeated every cycle until RST goes LOW. It leaves the internal registers as shown in Table 1.

The internal RAM is not affected by reset. When V_{CC} is turned on, the RAM content is indeterminate.

Power-on reset (see Fig. 18)

When V_{CC} is turned on, and provided its rise-time does not exceed 10 ms, an automatic reset can be obtained by connecting the RST pin to V_{CC} via a 10 μF capacitor. When the power is switched on, the current drawn by RST is the difference between V_{CC} and the capacitor voltage, and decreases from V_{CC} as the capacitor charges through the internal resistor (R_{RST}) to ground. The larger the capacitor, the more slowly V_{RST} decreases. V_{RST} must remain above the lower threshold of the Schmitt trigger long enough to effect a complete reset. The time required is the oscillator start-up time, plus 2 machine cycles.

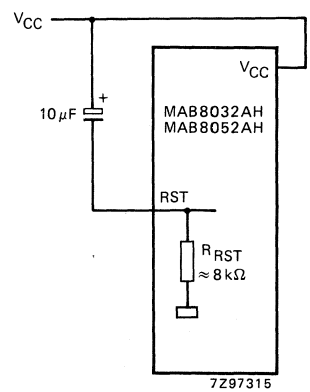


Fig. 18 Power-on reset.

INSTRUCTION SET

The MAB8052AH uses a powerful instruction set to allow expansion of on-chip CPU peripherals and to optimize byte efficiency and execution speed. Reassigned opcodes add new high-power operations and permit new addressing modes to make old operations more orthogonal. The instruction set consists of 49 single-byte, 45 two-byte and 17 three-byte instructions. When using a 12 MHz oscillator, 64 instructions execute in 1 μs and 45 instructions execute in 2 μs. Multiply and divide instructions execute in 4 μs.

Table 7 Instruction set description

mnemonic		description	bytes/ cycles	opcode (hex.)
Arithmetic operation				
ADD	A,Rr	Add register to A	1 1	2*
ADD	A,direct	Add direct byte to A	2 1	25
ADD	A,@Ri	Add indirect RAM to A	1 1	26, 27
ADD	A,#data	Add immediate data to A	2 1	24
ADDC	A,Rr	Add register to A with carry flag	1 1	3*
ADDC	A,direct	Add direct byte to A with carry flag	2 1	35
ADDC	A,@Ri	Add indirect RAM to A with carry flag	1 1	36, 37
ADDC	A,#data	Add immediate data to A with carry flag	2 1	34
SUBB	A,Rr	Subtract register from A with borrow	1 1	9*
SUBB	A,direct	Subtract direct byte from A with borrow	2 1	95
SUBB	A,@Ri	Subtract indirect RAM from A with borrow	1 1	96, 97
SUBB	A,#data	Subtract immediate data from A with borrow	2 1	94
INC	A	Increment A	1 1	04
INC	Rr	Increment register	1 1	0*
INC	direct	Increment direct byte	2 1	05
INC	@Ri	Increment indirect RAM	1 1	06, 07
DEC	A	Decrement A	1 1	14
DEC	Rr	Decrement register	1 1	1*
DEC	direct	Decrement direct byte	2 1	15
DEC	@Ri	Decrement indirect RAM	1 1	16, 17
INC	DPTR	Increment data pointer	1 2	A3
MUL	AB	Multiply A & B	1 4	A4
DIV	AB	Divide A by B	1 4	84
DA	A	Decimal adjust A	1 1	D4

INSTRUCTION SET (continued)

mnemonic		description	bytes/ cycles	opcode (hex.)
Logic operations				
ANL	A,Rr	AND register to A	1 1	5*
ANL	A,direct	AND direct byte to A	2 1	55
ANL	A,@Ri	AND indirect RAM to A	1 1	56, 57
ANL	A,#data	AND immediate data to A	2 1	54
ANL	direct,A	AND A to direct byte	2 1	52
ANL	direct,#data	AND immediate data to direct byte	3 2	53
ORL	A,Rr	OR register to A	1 1	4*
ORL	A,direct	OR direct byte to A	2 1	45
ORL	A,@Ri	OR indirect RAM to A	1 1	46, 47
ORL	A,#data	OR immediate data to A	2 1	44
ORL	direct,A	OR A to direct byte	2 1	42
ORL	direct,#data	OR immediate data to direct byte	3 2	43
XRL	A,Rr	Exclusive-OR register to A	1 1	6*
XRL	A,direct	Exclusive-OR direct byte to A	2 1	65
XRL	A,@Ri	Exclusive-OR indirect RAM to A	1 1	66, 67
XRL	A,#data	Exclusive-OR immediate data to A	2 1	64
XRL	direct, A	Exclusive-OR to direct byte	2 1	62
XRL	direct,#data	Exclusive-OR immediate data to direct byte	3 2	63
CLR	A	Clear A	1 1	E4
CPL	A	Complement A	1 1	F4
RL	A	Rotate A left	1 1	23
RLC	A	Rotate A left through the carry flag	1 1	33
RR	A	Rotate A right	1 1	03
RRC	A	Rotate A right through the carry flag	1 1	13
SWAP	A	Swap nibbles within A	1 1	C4

mnemonic		description	bytes/ cycles	opcode (hex.)
Data transfer				
MOV	A,Rr	Move register to A	1 1	E*
MOV	A,direct **	Move direct byte to A	2 1	E5
MOV	A,@Ri	Move indirect RAM to A	1 1	E6, E7
MOV	A,#data	Move immediate data to A	2 1	74
MOV	Rr,A	Move A to register	1 1	F*
MOV	Rr,direct	Move direct byte to register	2 2	A*
MOV	Rr,#data	Move immediate data to register	2 1	7*
MOV	direct,A	Move A to direct byte	2 1	F5
MOV	direct,Rr	Move register to direct byte	2 2	8*
MOV	direct,direct	Move direct byte to direct	3 2	85
MOV	direct,@Ri	Move indirect RAM to direct byte	2 2	86, 87
MOV	direct,#data	Move immediate data to direct byte	3 2	75
MOV	@Ri,A	Move A to indirect RAM	1 1	F6, F7
MOV	@Ri,direct	Move direct byte to indirect RAM	2 2	A6, A7
MOV	@Ri,#data	Move immediate data to indirect RAM	2 1	76, 77
MOV	DPTR,#data16	Load data pointer with a 16-bit constant	3 2	90
MOVC	A,@A+DPTR	Move code byte relative to DPTR to A	1 2	93
MOVC	A,@A+PC	Move code byte relative to PC to A	1 2	83
MOVX	A,@Ri	Move external RAM (8-bit address) to A	1 2	E2, E3
MOVX	A,@DPTR	Move external RAM (16-bit address) to A	1 2	E0
MOVX	@Ri,A	Move A to external RAM (8-bit address)	1 2	F2, F3
MOVX	@DPTR,A	Move A to external RAM (16-bit address)	1 2	F0
PUSH	direct	Push direct byte onto stack	2 2	C0
POP	direct	Pop direct byte from stack	2 2	D0
XCH	A,Rr	Exchange register with A	1 1	C*
XCH	A,direct	Exchange direct byte with A	2 1	C5
XCH	A,@Ri	Exchange indirect RAM with A	1 1	C6, C7
XCHD	A,@Ri	Exchange LOW-order digit indirect RAM with A	1 1	D6, D7

**MOV A,ACC is not a valid instruction.

INSTRUCTION SET (continued)

mnemonic		description	bytes/ cycles	opcode (hex.)
Boolean variable manipulation				
CLR	C	Clear carry flag	1 1	C3
CLR	bit	Clear direct bit	2 1	C2
SETB	C	Set carry flag	1 1	D3
SETB	bit	Set direct bit	2 1	D2
CPL	C	Complement carry flag	1 1	B3
CPL	bit	Complement direct bit	2 1	B2
ANL	C,bit	AND direct bit to carry flag	2 2	82
ANL	C,/bit	AND complement of direct bit to carry flag	2 2	B0
ORL	C,bit	OR direct bit to carry flag	2 2	72
ORL	C,/bit	OR complement of direct bit to carry flag	2 2	A0
MOV	C,bit	Move direct bit to carry flag	2 1	A2
MOV	bit,C	Move carry flag to direct bit	2 2	92
Program and machine control				
ACALL	addr11	Absolute subroutine call	2 2	•1addr
LCALL	addr16	Long subroutine call	3 2	12
RET		Return from subroutine	1 2	22
RET1		Return from interrupt	1 2	32
AJMP	addr11	Absolute jump	2 2	▲1addr
LJMP	addr16	Long jump	3 2	02
SJMP	rel	Short jump (relative address)	2 2	80
JMP	@A+DPTR	Jump indirect relative to the DPTR	1 2	73
JZ	rel	Jump if A is zero	2 2	60
JNZ	rel	Jump if A is not zero	2 2	70
JC	rel	Jump if carry flag is set	2 2	40
JNC	rel	Jump if no carry flag	2 2	50
JB	bit,rel	Jump if direct bit is set	3 2	20
JNB	bit,rel	Jump if direct bit is not set	3 2	30
JBC	bit,rel	Jump if direct bit is set and clear bit	3 2	10
CJNE	A,direct,rel	Compare direct to A and jump if not equal	3 2	B5
CJNE	A,#data,rel	Compare immediate to A and jump if not equal	3 2	B4
CJNE	Rr,#data,rel	Compare immed. to reg. and jump if not equal	3 2	B*
CJNE	@Ri,#data,rel	Compare immed. to ind. and jump if not equal	3 2	B6, B7
DJNZ	Rr,rel	Decrement register and jump if not zero	2 2	D*
DJNZ	direct,rel	Decrement direct and jump if not zero	3 2	D5
NOP		No operation	1 1	00

Notes to Table 7

Data addressing modes

Rr	Working register R0-R7.
direct	128 internal RAM locations and any special function register (SFR).
@Ri	Indirect internal RAM location addressed by register R0 or R1.
#data	8-bit constant included in instruction.
#data16	16-bit constant included as bytes 2 and 3 of instruction.
bit	Direct addressed bit in internal RAM or SFR.
addr16	16-bit destination address. Used by LCALL and LJMP. The branch will be anywhere within the 64 K-byte program memory address space.
addr11	11-bit destination address. Used by ACALL and AJMP. The branch will be within the same 2 K-byte page of program memory as the first byte of the following instruction.
rel	Signed (two's complement) 8-bit offset byte. Used by SJMP and all conditional jumps. Range is −128 to + 127 bytes relative to first byte of the following instruction.

Hexadecimal opcode cross-reference to Table 8

*: 8, 9, A, B, C, D, E, F.

●: 11, 31, 51, 71, 91, B1, D1, F1.

▲: 01, 21, 41, 61, 81, A1, C1, E1.

Table 8 Instruction map

first hexadecimal chacter of opcode ↓ — second hexadecimal character of opcode →

	0	1	2	3	4	5	6 | 7	8 | 9 | A	B	C	D	E	F
0	NOP	AJMP page 0	LJMP addr16	RR A	INC A	INC dir	INC @Ri 0 | 1	INC Rr 0 | 1 | 2	3	4	5	6	7
1	JBC bit, addr8	ACALL page 0	LCALL addr16	RRC A	DEC A	DEC dir	DEC @Ri 0 | 1	DEC Rr 0 | 1 | 2	3	4	5	6	7
2	JB bit, addr8	AJMP page 1	RET	RL A	ADD A,#data	ADD A,dir	ADD A,@Ri 0 | 1	ADD A,Rr 0 | 1 | 2	3	4	5	6	7
3	JNB bit, addr8	ACALL page 1	RET1	RLC A	ADDC A,#data	ADDC A,dir	ADDC A,@Ri 0 | 1	ADDC A,Rr 0 | 1 | 2	3	4	5	6	7
4	JC addr8	AJMP page 2	ORL dir,A	ORL dir,#data	ORL A,#data	ORL A,dir	ORL A,@Ri 0 | 1	ORL A,Rr 0 | 1 | 2	3	4	5	6	7
5	JNC addr8	ACALL page 2	ANL dir,A	ANL dir,#data	ANL A,#data	ANL A,dir	ANL A,@Ri 0 | 1	ANL A,Rr 0 | 1 | 2	3	4	5	6	7
6	JZ addr8	AJMP page 3	XRL dir,A	XRL dir,#data	XRL A,#data	XRL A,dir	XRL A,@Ri 0 | 1	XRL A,Rr 0 | 1 | 2	3	4	5	6	7
7	JNZ addr8	ACALL page 3	ORL C,bit	JMP @A+DPTR	MOV A,#data	MOV dir,#data	MOV @Ri,#data 0 | 1	MOV Rr,#data 0 | 1 | 2	3	4	5	6	7
8	SJMP addr8	AJMP page 4	ANL C,bit	MOVC A,@A+PC	DIV AB	MOV dir,dir	MOV dir,@Ri 0 | 1	MOV dir,Rr 0 | 1 | 2	3	4	5	6	7
9	MOV DPTR, #data 16	ACALL page 4	MOV bit,C	MOVC A,@A+DPTR	SUBB A,#data	SUBB A,dir	SUBB A,@Ri 0 | 1	SUBB A,Rr 0 | 1 | 2	3	4	5	6	7
A	ORL C,/bit	AJMP page 5	MOV C,bit	INC DPTR	MUL AB		MOV @Ri,dir 0 | 1	MOV Rr,dir 0 | 1 | 2	3	4	5	6	7
B	ANL C,/bit	ACALL page 5	CPL bit	CPL C	CJNE A, #data, addr8	CJNE A,dir, addr8	CJNE @Ri,#data, addr8 0 | 1	CJNE Rr,#data, addr8 0 | 1 | 2	3	4	5	6	7
C	PUSH dir	AJMP page 6	CLR bit	CLR C	SWAP A	XCH A,dir	XCH A,@Ri 0 | 1	XCH A,Rr 0 | 1 | 2	3	4	5	6	7
D	POP dir	ACALL page 6	SETB bit	SETB C	DA A	DJNZ dir, addr8	XCHD A,@Ri 0 | 1	DJNZ Rr, addr8 0 | 1 | 2	3	4	5	6	7
E	MOVX A,@DPTR	AJMP page 7	MOVX A,@Ri 0 | 1		CLR A	MOV A,dir	MOV A,@Ri 0 | 1	MOV A,Rr 0 | 1 | 2	3	4	5	6	7
F	MOVX @DPTR,A	ACALL page 7	MOVX @Ri,A 0 | 1		CPL A	MOV dir,A	MOV @Ri,A 0 | 1	MOV Rr,A 0 | 1 | 2	3	4	5	6	7

DEVELOPMENT DATA

RATINGS

Limiting values in accordance with the Absolute Maximum System (IEC 134)

Input voltage on any pin with respect to ground (V_{SS})	V_I	−0,5 to + 7	V
Total power dissipation	P_{tot}	max. 2	W
Input, output current	$\pm I_I$, I_O	max. 10	mA
Storage temperature range	T_{stg}	−65 to + 150	°C
Operating ambient temperature range	T_{amb}	0 to + 70	°C

D.C. CHARACTERISTICS

V_{CC} = 5 V (± 10%); V_{SS} = 0 V; T_{amb} = 0 to + 70 °C; all voltages with respect to V_{SS} unless otherwise specified

parameter	symbol	min.	max.	unit	conditions
Supply current	I_{CC}	–	175	mA	all outputs disconnected; $\overline{EA} = V_{CC}$
Inputs					
Input voltage LOW	V_{IL}	−0,5	0,8	V	
Input voltage HIGH all inputs except RST and XTAL 2	V_{IH}	2	V_{CC} + 0,5	V	
Input voltage HIGH to RST and XTAL 2	V_{IH1}	2,5	V_{CC} + 0,5 V	V	XTAL 1 to V_{SS}
Outputs					
Output voltage LOW (Ports 1, 2, 3) (note 1)	V_{OL}	–	0,45	V	I_{OL} = 1,6 mA
Output voltage LOW (Port 0, ALE, $\overline{PSEN}$) (note 1)	V_{OL1}	–	0,45	V	I_{OL1} = 3,2 mA
Output voltage HIGH (Ports 1, 2, 3)	V_{OH}	2,4	–	V	I_{OH} = −80 μA
Output voltage HIGH (Port 0, ALE, $\overline{PSEN}$)	V_{OH1}	2,4	–	V	I_{OH1} = −400 μA
Input leakage current (Port 0, $\overline{EA}$)	$\pm I_{LI}$	–	10	μA	0,45 V < V_I < V_{CC}
Input current HIGH (RST)	I_{IH1}	–	500	μA	$V_I = V_{CC}$−1,5 V
current logic 0 (Ports 1, 2, 3)	I_{IL}	–	−800	μA	V_{IL} = 0,45 V;
Input current logic 0 (XTAL 2)	I_{IL2}	–	−3,2	mA	V_{IL} = 0,45 V; XTAL 1 to V_{SS}
Capacitance of I/O buffer	$C_{I/O}$	–	10	pF	f_c = 1 MHz; T_{amb} = 25 °C

Note 1

V_{OL} is degraded when the MAB8052AH rapidly discharges external capacitance.
This a.c. noise is most pronounced during emission of address data. When using external memory, locate the latch or buffer as close to the MAB8052AH as possible.

datum	emitting ports	time slot interval	degraded I/O lines	V_{OL} (max.)
address	P2, P0	TS3, TS9	P1, P3	0,8 V
write data	P0	TS6	P1, P3, ALE	0,8 V

DEVELOPMENT DATA

A.C. CHARACTERISTICS

V_{CC} = 5 V ± 10%; V_{SS} = 0 V; T_{amb} = 0 to + 70 °C; C_L = 100 pF (Port 0, ALE and $\overline{PSEN}$); C_L = 80 pF all other outputs unless otherwise specified (see waveforms Figs 21, 22 and 23).

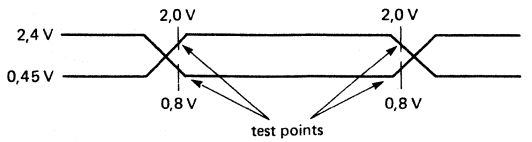
parameter	symbol	12 MHz		variable clock (note 1)		unit
		min.	max.	min.	max.	
Program memory						
ALE pulse duration	t_{LL}	127	–	$2t_{CK}-40$	–	ns
Address set-up time to ALE	t_{AL}	43	–	$t_{CK}-40$	–	ns
Address hold time after ALE	t_{LA}	48	–	$t_{CK}-35$	–	ns
Time from ALE to valid instruction input	t_{LIV}	–	233	–	$4t_{CK}-100$	ns
Time from ALE to control pulse $\overline{PSEN}$	t_{LC}	58	–	$t_{CK}-25$	–	ns
Control pulse duration $\overline{PSEN}$	t_{CC}	215	–	$3t_{CK}-35$	–	ns
Time from $\overline{PSEN}$ to valid instruction input	t_{CIV}	–	125	–	$3t_{CK}-125$	ns
Input instruction hold time after $\overline{PSEN}$	t_{CI}	0	–	0	–	ns
Input instruction float delay after $\overline{PSEN}$ (note 2)	t_{CIF}	–	63	–	$t_{CK}-20$	ns
Address valid after $\overline{PSEN}$ (note 2)	t_{AC}	75	–	$t_{CK}-8$	–	ns
Address to valid instruction input	t_{AIV}	–	302	–	$5t_{CK}-115$	ns
Address float time to $\overline{PSEN}$	t_{AFC}	−12	–	−12	–	ns

A.C. CHARACTERISTICS (continued)

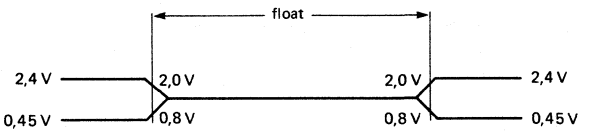
parameter	symbol	12 MHz		variable clock (note 1)		unit
		min.	max.	min.	max.	
External data memory						
$\overline{RD}$ pulse duration	t_{RR}	400	–	$6t_{CK}-100$	–	ns
$\overline{WR}$ pulse duration	t_{WW}	400	–	$6t_{CK}-100$	–	ns
Address hold time after ALE	t_{LA}	48	–	$t_{CK}-35$	–	ns
RD to valid data input	t_{RD}	–	250	–	$5t_{CK}-165$	ns
Data hold time after $\overline{RD}$	t_{DR}	0	–	0	–	ns
Data float delay after $\overline{RD}$	t_{DFR}	–	97	–	$2t_{CK}-70$	ns
Time from ALE to valid data input	t_{LD}	–	517	–	$8t_{CK}-150$	ns
Address to valid data input	t_{AD}	–	585	–	$9t_{CK}-165$	ns
Time from ALE to $\overline{RD}$ or $\overline{WR}$	t_{LW}	200	300	$3t_{CK}-50$	$3t_{CK}+50$	ns
Time from address to $\overline{RD}$ or $\overline{WR}$	t_{AW}	203	–	$4t_{CK}-130$	–	ns
Time from $\overline{RD}$ or $\overline{WR}$ HIGH to ALE HIGH	t_{WHLH}	43	123	$t_{CK}-40$	$t_{CK}+40$	ns
Data valid to $\overline{WR}$ transition	t_{DWX}	23	–	$t_{CK}-60$	–	ns
Data set-up time before $\overline{WR}$	t_{DW}	433	–	$7t_{CK}-150$	–	ns
Data hold time after $\overline{WR}$	t_{WD}	33	–	$t_{CK}-50$	0	ns
Address float delay after $\overline{RD}$	t_{AFR}	–	+ 12	–	+ 12	ns

Notes to the a.c. characteristics

1. $1/t_{CK}$ = 3,5 to 12 MHz (see Fig. 20 and Table 9).
2. Interfacing the MAB8052AH to devices with float times up to 75 ns is permitted. This limited bus contention will cause damage to port 0 drivers.



(a)



(b)

A.C. testing inputs are driven at 2,4 V for a logic 1 and 0,45 V for a logic 0. Timing measurements are taken at 2,0 V for a logic 1 and 0,8 V for logic 0. The float state is defined as the point at which a Port 0 pin sinks 3,2 mA or sources 400 μA at the voltage test levels.

Fig. 19 A.C. testing input, output waveform (a) and float waveform (b).

DEVELOPMENT DATA

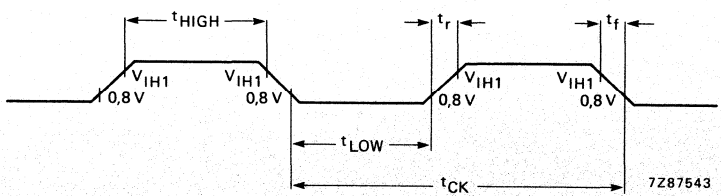


Fig. 20 External clock drive XTAL 2 (see Table 9).

Table 9 External clock drive XTAL 2 (see Fig. 20)

parameter	symbol	variable clock (f = 3,5 to 12 MHz)		unit
		min.	max.	
oscillator clock period	t_{CK}	83,3	286	ns
HIGH time	t_{HIGH}	20	$t_{CK}-t_{LOW}$	ns
LOW time	t_{LOW}	20	$t_{CK}-t_{HIGH}$	ns
rise time	t_r	–	20	ns
fall time	t_f	–	20	ns

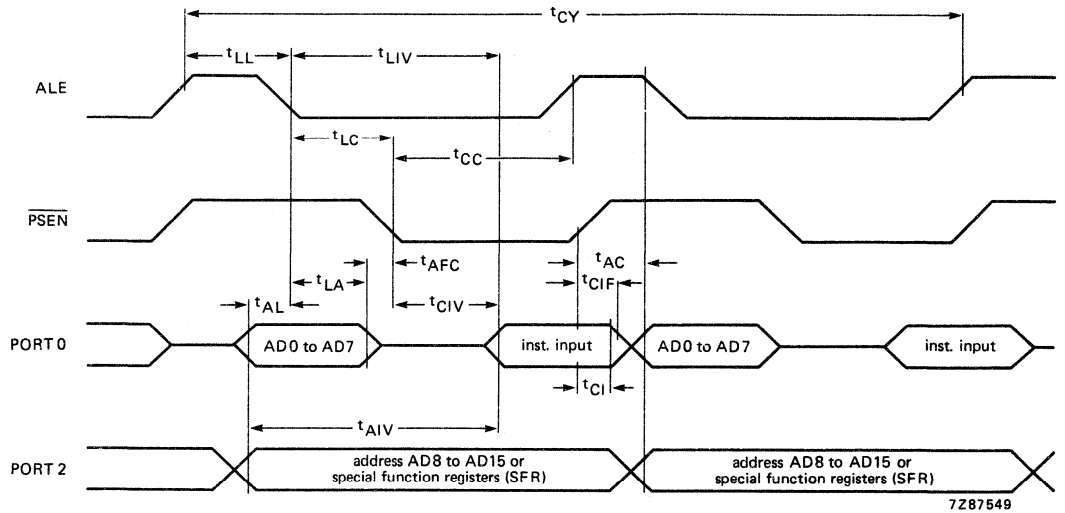


Fig. 21 Read from program memory.

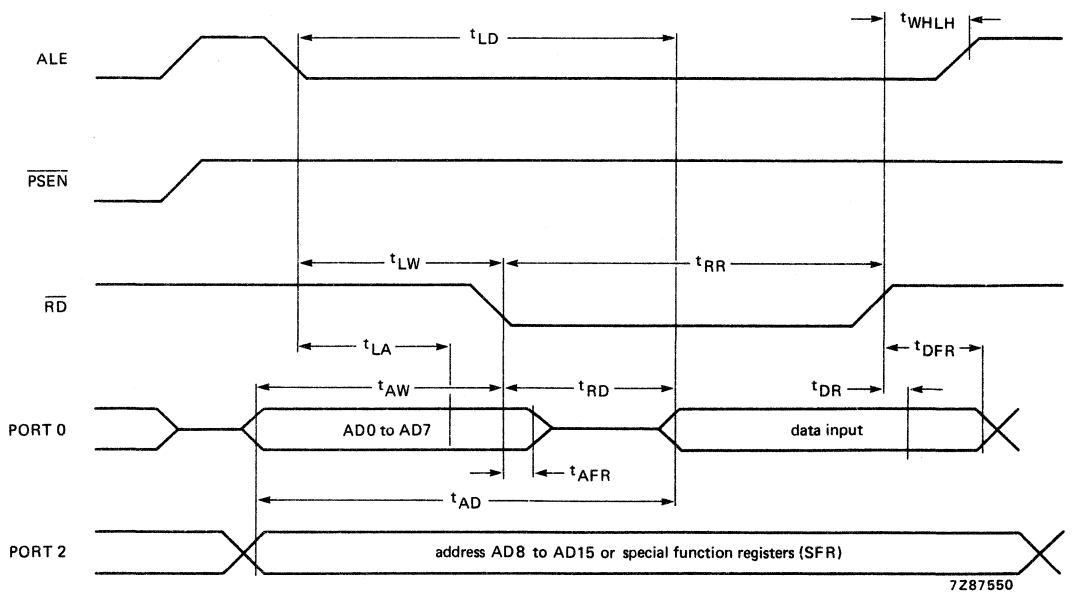


Fig. 22 Read from data memory.

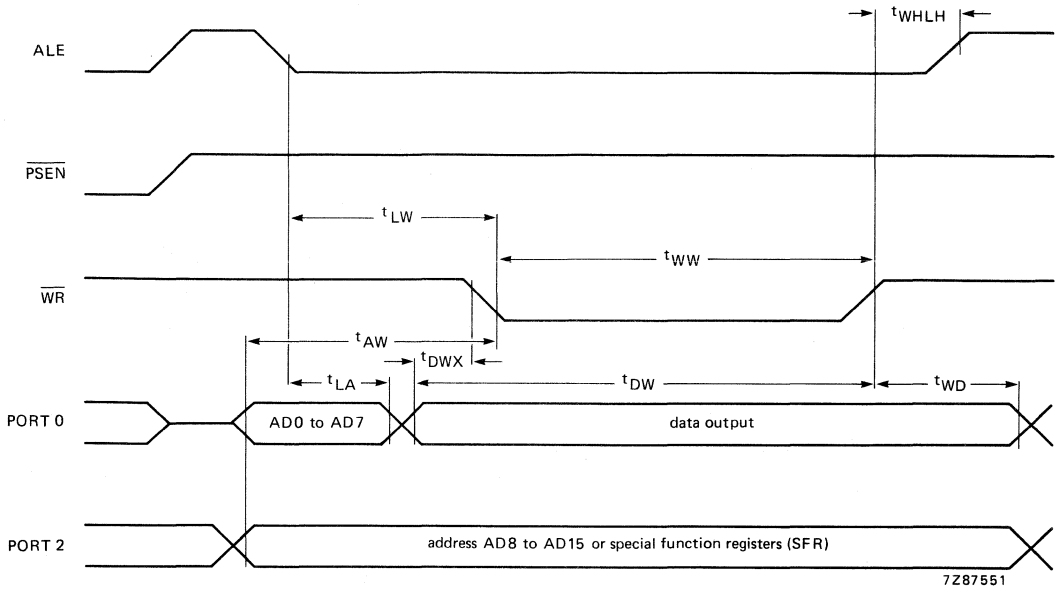


Fig. 23 Write to data memory.

DEVELOPMENT DATA

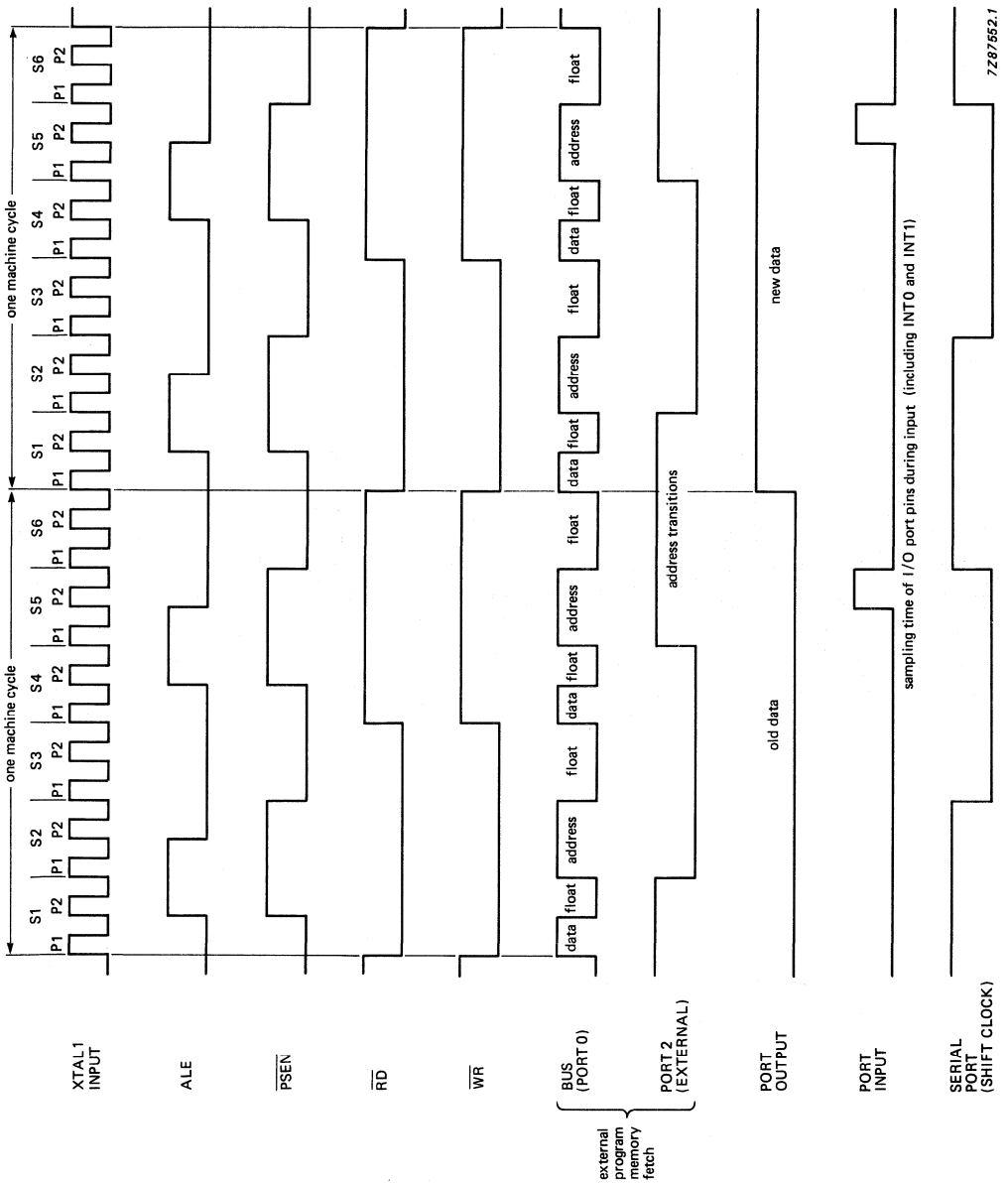


Fig. 24 Instruction cycle timing.

SINGLE-CHIP 8-BIT MICROCONTROLLER

GENERAL DESCRIPTION

The MAB80XXH family of single-chip 8-bit microcontrollers are fabricated in NMOS.
Three interchangeable (pin compatible) versions are available:

- MAB8048H with resident mask-programmed 1 K x 8 ROM, 64 x 8 RAM
- MAB8035HL without resident program memory for use with external EPROM/ROM, 64 x 8 RAM.
- MAB8049H with resident mask-programmed 2 K x 8 ROM, 128 x 8 RAM
- MAB8039HL without resident program memory for use with external EPROM/ROM, 128 x 8 RAM.
- MAB8050H with resident mask-programmed 4 K x 8 ROM, 256 x 8 RAM
- MAB8040HL without resident program memory for use with external EPROM/ROM, 256 x 8 RAM.

The MAB80XXH family are designed to be efficient control processors as well as arithmetic processors. Their instruction set allows the user to directly set and reset individual I/O lines as well as test individual bits within the accumulator. A large variety of branch and table look-up instructions enable efficient implementation of standard logic functions. Code efficiency is high; over 70% of the instructions are single byte; all others are two byte.

An on-chip 8-bit counter is provided, which can count either machine cycles (÷ 32) or external events. The counter can be programmed to cause an interrupt to the processor.
Program and data memories plus input/output capabilities can be expanded using standard devices.
For further detailed information see users manual 'single-chip 8-bit microcontrollers'.

Features

- 8-bit CPU, ROM, RAM and I/O
- Internal counter/timer
- Internal oscillator, clock driver
- Single-level interrupts: external and counter/timer
- 17 internal registers: accumulator, 16 addressable registers
- Over 90 instructions: 70% single byte
- All instructions 1 or 2 cycles
- Easily expandable memory and 27 I/O lines
- TTL compatible inputs and outputs
- Single 5 V supply
- Standard and extended temperature range (see Table 1)

Applications

- Peripheral interfaces and controllers
- Test and measuring instruments
- Sequencers
- Modems and data enciphering
- Environmental control systems
- Audio/video systems

PACKAGE OUTLINES

All versions: with type no. suffix P (see Table 1): 40-lead DIL; plastic (SOT-129).
MAB80XXH/HLWP: 44-lead plastic leaded chip-carrier (PLCC); SOT-187A.

Table 1 MAB80XXH versions

version	internal memory		RAM st/by.	frequency (MHz) min.	frequency (MHz) max.	temperature range (°C)
MAB8048H	1 K x 8 ROM	64 byte RAM	yes	1,0	11,0	0 to +70
MAB8035HL	none	64 byte RAM	yes	1,0	11,0	0 to +70
MAF8048H	1 K x 8 ROM	64 byte RAM	yes	1,0	11,0	−40 to +85
MAF8035HL	none	64 byte RAM	yes	1,0	11,0	−40 to +85
MAF80A48H	1 K x 8 ROM	64 byte RAM	yes	1,0	10,0	−40 to +110
MAF80A35HL	none	64 byte RAM	yes	1,0	10,0	−40 to +110
MAB8049H	2 K x 8 ROM	128 byte RAM	yes	1,0	11,0	0 to +70
MAB8039HL	none	128 byte RAM	yes	1,0	11,0	0 to +70
MAF8049H	2 K x 8 ROM	128 byte RAM	yes	1,0	11,0	−40 to +85
MAF8039HL	none	128 byte RAM	yes	1,0	11,0	−40 to +85
MAF80A49H	2 K x 8 ROM	128 byte RAM	yes	1,0	10,0	−40 to +110
MAF80A39HL	none	128 byte RAM	yes	1,0	10,0	−40 to +110
MAB8050H	4 K x 8 ROM	256 byte RAM	yes	1,0	6,0	0 to +70
MAB8040HL	none	256 byte RAM	yes	1,0	6,0	0 to +70
MAF8050H	4 K x 8 ROM	256 byte RAM	yes	1,0	6,0	−40 to +85
MAF8040HL	none	256 byte RAM	yes	1,0	6,0	−40 to +85
MAF80A50H	4 K x 8 ROM	256 byte RAM	yes	1,0	6,0	−40 to +110
MAF80A40HL	none	256 byte RAM	yes	1,0	6,0	−40 to +110

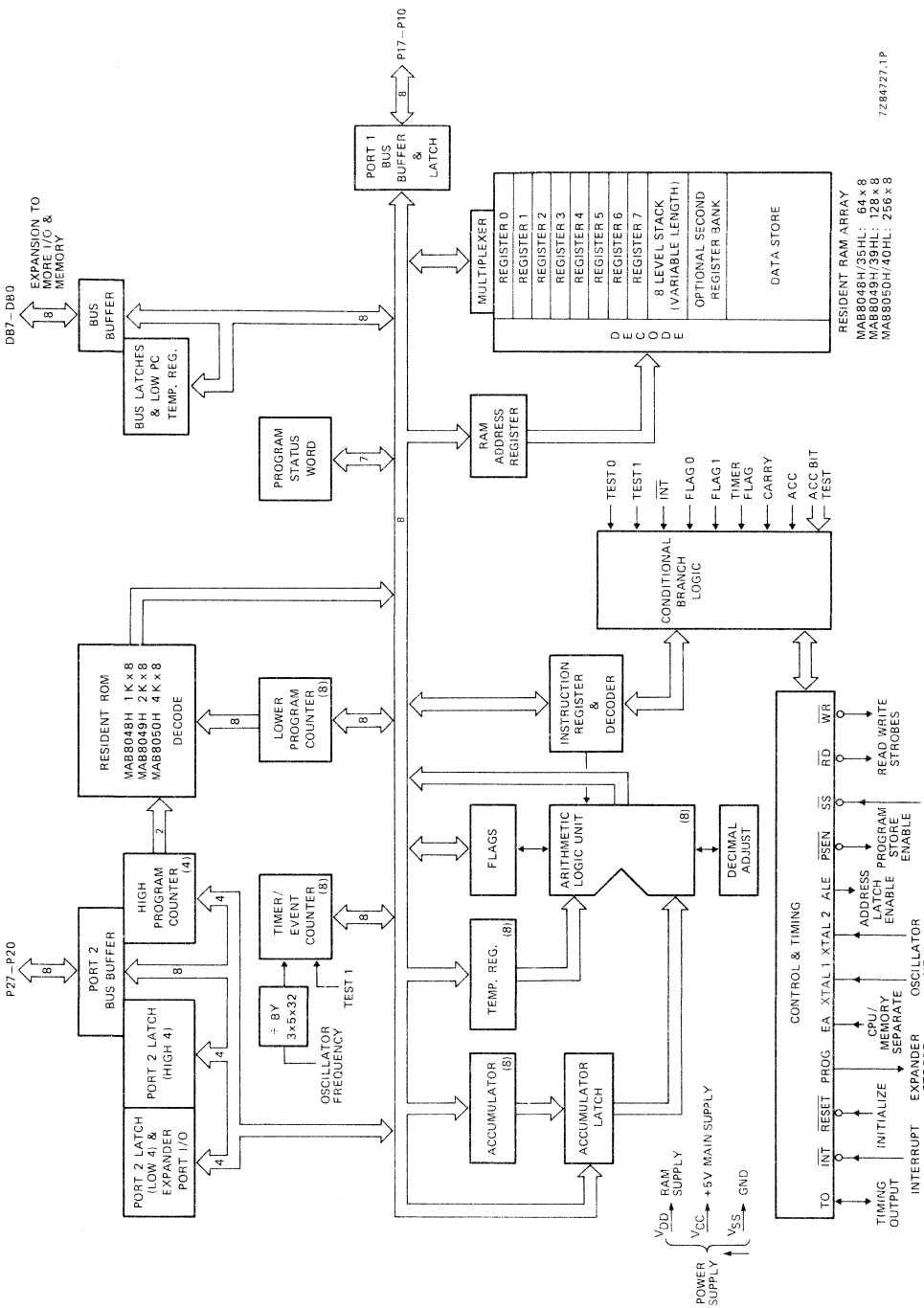


Fig. 1 Block diagram.

PINNING

12–19	DB0–DB7	**Data Bus**: bidirectional I/O port which can write or read using the $\overline{RD}$ and $\overline{WR}$ strobes. This port can also be statically latched. It contains the 8 lower order address bits during external memory access and receives the addressed instruction under control of $\overline{PSEN}$. $\overline{PSEN}$, ALE, $\overline{RD}$ and $\overline{WR}$ determine whether the access is an instruction fetch or a read/write access to external RAM.
27–34	P10–P17	**Port 1**: 8-bit quasi-bidirectional I/O port (note 1).
21–24	P10–P27	**Port 2**: 8-bit quasi-bidirectional I/O port (note 1).
35–38		P20–P23 contain the 4 higher order address bits during an access of external program memory.
25	$\overline{PROG}$	**Output strobe**: active LOW for 8243 I/O expander.
1	T0	**Test 0**: test input pin sensed using the JT0 and JNT0 instructions. **Clock**: clock output pin when designated by the ENT0 CLK instructions.
39	T1	**Test 1**: test input pin sensed using the JT1 and JNT1 instructions. Can be designated as the timer/counter input by the STRT CNT instruction.
6	$\overline{INT}$	**Interrupt**: interrupt input pin, which causes an interrupt in the current program, provided that the external interrupt is enabled. Can also be used as an input, testable using the JNI instruction. Interrupt is disabled during and after $\overline{RESET}$.
4	$\overline{RESET}$	**Reset**: active LOW input used to initialize the microcontroller. During program verification the address is latched by a '0' to '1' transition on $\overline{RESET}$ and the data at the addressed location is output on BUS (note 2).
11	ALE	**Address latch enable**: occurs each cycle and is used for timing and sampling. During external program or data memory access, ALE is used to strobe the address information multiplexed on the DB0 to DB7 outputs.
8	$\overline{RD}$	**Read BUS**: active LOW strobe used to gate data onto BUS lines when reading from an external source.
10	$\overline{WR}$	**Write BUS**: active LOW strobe used to write data from BUS lines to an external designation.
7	EA	**External access input**: when HIGH, forces instruction fetch from external memory.
9	$\overline{PSEN}$	**Program store enable**: active LOW strobe that occurs only during a fetch from external memory.
5	$\overline{SS}$	**Single step**: active LOW input used with ALE to cause the microcontroller to execute a single instruction.
2 3	XTAL 1 XTAL 2	**Crystal inputs**: inputs for a crystal, LC-network or an external timing signal to determine the internal oscillator frequency (note 2).
20	V_{SS}	**Ground**: circuit earth potential.
40	V_{CC}	**Power supply**: + 5 V main supply pin.
26	V_{DD}	**Power supply**: + 5 V RAM standby power supply; low power standby pin.

Notes

1. Each port line can be designated as an input or an output. A line is designated as an input by first writing a logic 1 to the line. $\overline{\text{RESET}}$ sets all lines to logic 1.
2. Non-standard TTL V_{IH}.

FUNCTIONAL DESCRIPTION

The following sections provide a detailed functional description of the MAB80XXH microcontroller as shown in Fig. 1. The generic term "MAB80XXH" is used to refer collectively to the MAB8048H/35HL, MAB8049H/39HL and MAB8050H/40HL.

Program memory (see Fig. 3)

The resident program memory consists of a 1024, 2048 or 4096 byte ROM (MAB8048H/49H/50H); the MAB8035HL/39HL/40HL versions do not have a resident program memory. The total addressing capability is 4096 bytes.
The program memory address space is divided into two 2048-byte banks MB0 and MB1. These two 2048 byte banks are divided into 8 pages of 256 bytes for conditional branches.
There are three locations in program memory which contain the first instruction to be executed after one of three events:

- Location 0 – if $\overline{\text{RESET}}$ is activated (LOW) then location 0 is activated
- Location 3 – if the external interrupt is enabled and the $\overline{\text{INT}}$ line is activated (LOW)
- Location 7 – if the T/C interrupt is enabled and the timer/counter has an overflow.

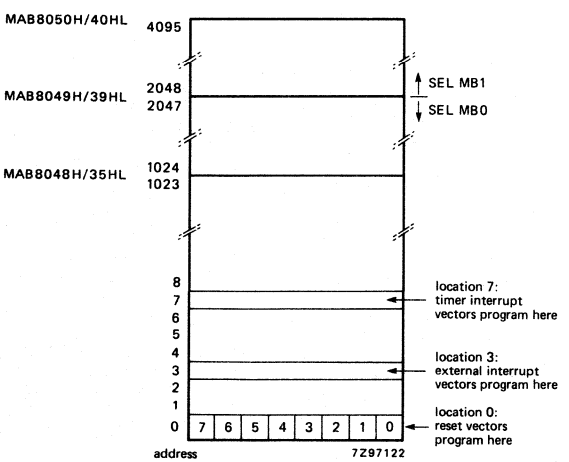


Fig. 3 Program memory map.

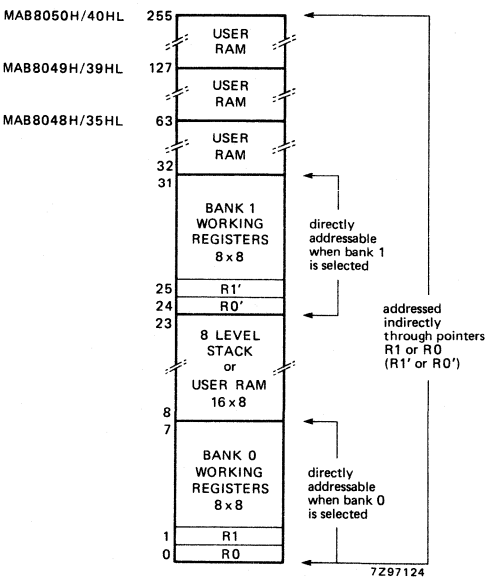


Fig. 4 Data memory map.

FUNCTIONAL DESCRIPTION (continued)

Data memory (see Fig. 4)

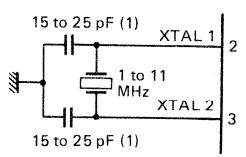
The resident data memory consists of a 64, 128 or 256 byte RAM. All locations are indirectly addressable using two RAM pointer registers R0, R1 or R0', R1'. The first 8 RAM locations (0 to 7) are designated as working registers bank 0 and are directly addressable. By selecting register bank 1, RAM locations 24 to 31 become the working registers, replacing those in register bank 0. RAM locations 8 to 23 are designated as the stack. Two bytes are used per CALL allowing up to 8 levels of subroutine nesting.
An extra 256 bytes of RAM may be added and addressed directly using the MOVX instructions. If the required extra RAM is greater than 320 bytes, additional (256 byte) banks of external memory can be selected, one at a time, by an I/O port.

Program counter and stack

The program counter (PC) is a 12-bit counter/register that points to the location from which the next instruction is to be fetched. When EA is logic 0 the PC can address locations 0 to 1023 (8048H), 2047 (8049H) or 4095 (8050H) of internal program memory. At the 1 K (8048H), 2 K (8049H) boundary, an automatic switch-over to external memory occurs. When EA is logic 1 all the program is fetched from external ROM/EPROM. The total address space is 4 K bytes. An interrupt or CALL to a subroutine causes the contents of the program counter to be stored in one of the 8 register pairs of the program stack. The pair to be used is determined by a 3-bit stack pointer which is part of the program status word (PSW). Data RAM locations 8 to 23 are available as stack registers and are used to store the program counter and 4 bits of PSW. The stack pointer, when initialized to 000B, points to RAM locations 8 and 9. The first subroutine jump or interrupt results in the program counter contents being transferred to locations 8 and 9 of the RAM array. The stack pointer is then incremented by one to point to locations 10 and 11 in anticipation of another CALL. Nesting of subroutines within subroutines can continue up to 8 times without overflowing the stack. If overflow does occur the deepest address stored (locations 8 and 9) will be overwritten and lost since the stack pointer overflows from 111 to 000. It also underflows from 000 to 111. The end of a subroutine, which is signalled by a return instruction (RET or RETR), causes the stack pointer to be decremented and the contents of the resulting register pair to be transferred to the program counter.

Oscillator and clock (see Figs 5, 6 and 7)

The MAB80XXH contains its own internal oscillator and clock driver. A crystal, LC-network or external timing signal (pulse generator) determines the oscillator frequency. The output of the oscillator is divided-by-three and is available at T0 (pin 1) by executing the ENT0 CLK instruction. This clock signal (CLK) is divided-by-five to define a machine (instruction) cycle. It is available at ALE (pin 11).



(1) Including crystal-socket stray capacities.

Fig. 5 Crystal oscillator mode. Crystal series impedance should be $< 75\ \Omega$ at 6 MHz and $< 180\ \Omega$ at 3,6 MHz. When using a ceramic oscillator both capacitors should be 30 pF.

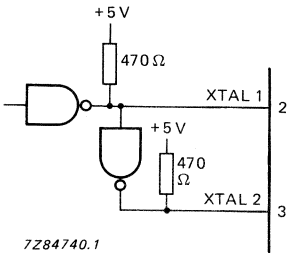
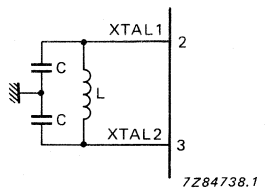


Fig. 6 Driving from external source. Both XTAL 1 and XTAL 2 should be driven. Resistors to V_{CC} (+ 5 V) are required to ensure V_{IH} = 3,8 V if TTL circuitry is used. The minimum HIGH and LOW times are 45%.

L (μH)	C (pF)	nom. f (MHz)
45	20	5,2
120	20	3,2



$$f \approx \frac{1}{2\pi\sqrt{LC'}} \; ; \; C' = \frac{C + 3C_{pp}}{2}$$

Fig. 7 LC oscillator mode. Each C should be ≈ 20 pF including stray capacitance C_{pp} ≈ 5 to 10 pF (pin-to-pin capacitance).

Timer/event counter

An internal counter is available which can count either external events or machine cycles (÷ 32). The machine cycles are divided-by-32 before they are applied to the input of the 8-bit counter. External events are applied directly to the input of the counter. The maximum frequency that can be counted is one third of the machine cycle frequency. The minimum positive duty cycle that can be detected is 0,2 times the cycle period. The counter is under program control and can be made to generate an interrupt to the processor when it overflows.

Interrupt

An interrupt may be generated by:

- An external input $\overline{INT}$ (pin 6)
  or
- Overflow of the internal counter, when enabled.

In either, the processor completes execution of the present instruction followed by a CALL to the interrupt service routine.
After service, a RETR instruction restores the machine to the state it was prior to the interrupt. The external interrupt has priority over the internal interrupt.

Input/output

The MAB80XXH has 27 I/O lines. They are arranged as three 8-line ports which serve as either inputs, outputs or bidirectional ports and 3 'test' inputs that can alter program sequences when tested by conditional jump instructions.

FUNCTIONAL DESCRIPTION (continued)

Ports 1 and 2 (see Fig. 8)

Ports 1 and 2 are both 8-bits wide and have identical characteristics. Data written to these ports is statically latched and remains unchanged until rewritten. As input lines these ports are non-latching, i.e. inputs must be present until read by an input instruction. Inputs are fully TTL compatible and outputs will drive one standard TTL load.

The lines of ports 1 and 2 are called quasi-bidirectional because of a special output circuit structure which allows each line to serve as an input, an output, or both even through outputs are statically latched. Each line is continuously pulled up to + 5 V through a resistor (≈ 50 Ω). Thus pull-up provides sufficient source current for a TTL HIGH level, yet can be pulled LOW by a standard TTL gate, thus allowing the pin to be used both as an input and an output. To provide fast switching times during a logic 0 to 1 transition, transistor TR 2 is switched on for one fifth of a machine cycle when a logic 1 is written to the line. When a logic 0 is written transistor TR 1 overcomes the pull-up and provides TTL current sinking capability. Since the pull-down transistor is low impedance, a logic 1 must first be written to any line which is to used as an input. $\overline{\text{RESET}}$ initializes all lines to the high impedance logic 1 state. This structure allows input and output on the same pin plus a mixture of input and output lines on the same port. The quasi-bidirectional port in combination with the ANL and ORL logical instructions facilitates the handling of single line inputs and outputs within an 8-bit microcontroller.

BUS (DB0–DB7)

Bus is a true bidirectional 8-bit port with associated input and output strobes. If the bidirectional feature is not required, BUS can serve as a statically latched output port or non-latching input port. Input and output lines on this port cannot be mixed.

As a static port, data is written and latched using the OUTL instruction and input using the INS instruction. These instructions generate pulses on the corresponding $\overline{\text{RD}}$ and $\overline{\text{WR}}$ output strobe lines, but in the static port mode they are not used. As a bidirectional port, the MOVX instructions are used to read from and write to the port. A write generates a pulse on the $\overline{\text{WR}}$ output line and output data is valid at the trailing edge of $\overline{\text{WR}}$. A read generates a pulse on the $\overline{\text{RD}}$ output line and input data must be valid at the trailing edge of $\overline{\text{RD}}$. When the BUS lines are not being written to or read from they are high impedance.

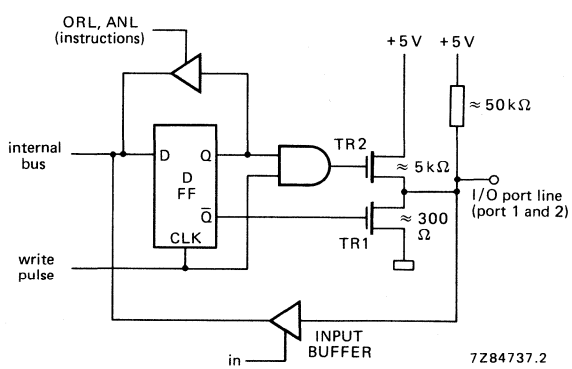


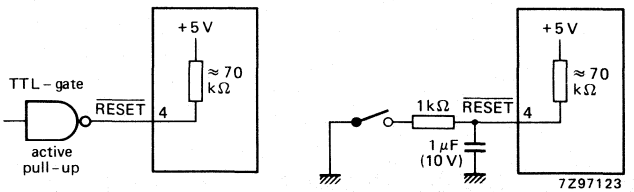
Fig. 8 Quasi-bidirectional port structure.

Test (T0, T1) and $\overline{INT}$

These three pins serve as inputs and are testable with the conditional jump instruction. They allow inputs to cause program branches without the necessity of loading an input port into the accumulator.

$\overline{RESET}$ (see Fig. 9)

This active LOW input is used to initialize the microcontroller.
This Schmitt-trigger input has an internal pull-up resistor which, in combination with an internal 1 μF capacitor, provides an internal reset pulse of sufficient duration to reset all circuitry. If the reset pulse is generated externally, the reset pin must be held at ground (0,45 V) for at least 10 ms after the power supply is within tolerance. Only 5 machine cycles (12,5 μs at 6 MHz) are required if power is already on and the oscillator has stabilized.



(a) (b)

Fig. 9 An external reset is shown in (a) and power-on reset in (b).

Single step ($\overline{SS}$)

This active LOW input when used in combination with ALE will cause the microcontroller to execute a single instruction, then wait until $\overline{SS}$ is reactivated.

Power-down mode (see Fig. 10)

In the MAB80XXH, power can be removed from all but the data RAM array, for low power standby operation. In the power-down mode the contents of the data RAM can be maintained while drawing typically 10% to 15% of the normal operating supply voltage. V_{CC} serves as the + 5 V supply pin for the bulk of the circuitry, while the V_{DD} pin supplies only the RAM array. In normal operation, both pins are at + 5 V. In the standby mode, V_{CC} is at ground and only V_{DD} is maintained at + 5 V. Applying $\overline{RESET}$ to the microcontroller through the reset pin inhibits any access to the RAM and ensures that the RAM cannot be inadvertently altered as power is removed from V_{CC}.

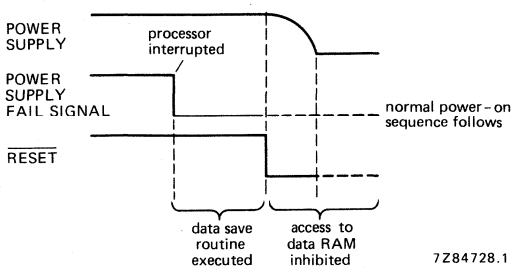


Fig. 10 Power down sequence.

FUNCTIONAL DESCRIPTION (continued)

Instruction set (see Tables 1, 2, 3 and 4)

The MAB80XXH instruction set consists of over 90 one and two-byte instructions. Program code efficiency is high because:

- Working registers and program variables are stored in the RAM locations 0 to 127, which require only a single byte to address
- Program memory is divided into pages of 256 bytes, which means that branch destination addresses require only one byte

The instruction set performs logical, arithmetic and test operations on bytes. It also manipulates and tests bits. A set of MOVE instructions operate indirectly on either RAM or ROM, which permits efficient access of pointers and data tables. The indirect jump instruction performs a multi-way branch (up to 256) on the contents of the accumulator to addresses stored in a look-up table. The 'decrement register and jump if not zero' instruction saves a bit each time it is used as opposed to using separate increment and test instructions. The on-chip counter provides the facility for external events or time to be counted off-line from the main program. The MAB80XXH can either test the counter (under program control) or cause its overflow to generate an interrupt. These features are essential for real-time applications.

Table 1 Symbols and definitions used in Table 2.

symbol	definition description
A	accumulator
addr	program memory address
Bb	bit designation (b = 0–7)
RBS	register bank select
C	carry (bit CY)
CNT	event counter
D	mnemonic for 4-bit digit (nibble)
data	8-bit number or expression
I	interrupt
MB	memory bank
MBFF	memory bank flip-flop
P	mnemonic for 'in-page' operation
PC	program counter
Pp	port designation (p = 0, 1, 2)
PSW	program status word
RB	register bank
Rr	register designation (r = 0–7)
Sn	serial I/O register
SP	stack pointer
T	timer
TF	timer flag
T0	test 0 input
T1	test 1 input
#	immediate data prefix
@	indirect address prefix
(X)	contents of X
((X))	contents of location addressed by X
←	is replaced by
↔	is exchanged with

Table 2 Instruction set

	mnemonic	opcode (hex.)	bytes/ cycles	description	function		notes
ACCUMULATOR	ADD A, Rr	6*	1/1	Add register contents to A	(A)←(A) + (Rr)	r = 0–7	1
	ADD A, @Rr	60 61	1/1	Add RAM data, addressed by Rr, to A	(A)←(A) + ((R0)) (A)←(A) + ((R1))		1
	ADD A, #data	03 data	2/2	Add immediate data to A	(A)←(A) + data		1
	ADDC A, Rr	7*	1/1	Add carry and register contents to A	(A)←(A) + (Rr) + (C)	r = 0–7	1
	ADDC A, @Rr	70 71	1/1	Add carry and RAM data, addresses by Rr, to A	(A)←(A) + ((R0)) + (C) (A)←(A) + ((R1)) + (C)		1
	ADDC A, #data	13 data	2/2	Add carry and immediate data to A	(A)←(A) + data + (C)		1
	ANL A, Rr	5*	1/1	'AND' Rr with A	(A)←(A) AND (Rr)	r = 0–7	
	ANL A, @Rr	50 51	1/1	'AND' RAM data, addressed by Rr, with A	(A)←(A) AND ((R0)) (A)←(A) AND ((R1))		
	ANL A, #data	53 data	2/2	'AND' immediate data with A	(A)←(A) AND data		
	ORL A, Rr	4*	1/1	'OR' Rr with A	(A)←(A) OR (Rr)	r = 0–7	
	ORL A, @Rr	40 41	1/1	'OR' RAM data, addressed by Rr, with A	(A)←(A) OR ((R0)) (A)←(A) OR ((R1))		
	ORL A, #data	43 data	2/2	'OR' immediate data with A	(A)←(A) OR data		
	XRL A, Rr	D*	1/1	'XOR' Rr with A	(A)←(A) XOR (Rr)	r = 0–7	
	XRL A, @Rr	D0 D1	1/1	'XOR' RAM, addressed by Rr, with A	(A)←(A) XOR ((R0)) (A)←(A) XOR ((R1))		
	XRL A, #data	D3 data	2/2	'XOR' immediate data with A	(A)←(A) XOR data		
	INC A	17	1/1	increment A by 1	(A)←(A) + 1		
	DEC A	07	1/1	decrement A by 1	(A)←(A) – 1		
	CLR A	27	1/1	clear A to zero	(A)←0		
	CPL A	37	1/1	one's complement A	(A)←NOT(A)		
	RL A	E7	1/1	rotate A left	$(A_{n+1})\leftarrow(A_n)$ $(A_0)\leftarrow(A_7)$	n = 0–6	

	mnemonic	opcode (hex.)	bytes/ cycles	description	function		notes
ACCUMULATOR (cont.)	RLC A	F7	1/1	rotate A left through carry	$(A_{n+1})\leftarrow(A_n)$ $(A_0)\leftarrow(C)$, $(C)\leftarrow(A_7)$	n = 0–6	2
	RR A	77	1/1	rotate A right	$(A_n)\leftarrow(A_{n+1})$ $(A_7)\leftarrow(A_0)$	n = 0–6	
	RRC A	67	1/1	rotate A right through carry	$(A_n)\leftarrow(A_{n+1})$ $(A_7)\leftarrow(C)$, $(C)\leftarrow(A_0)$	n = 0–6	2
	DA A	57	1/1	decimal adjust A			2
	SWAP A	47	1/1	swap nibbles of A	$(A_{4-7})\leftrightarrow(A_{0-3})$		
DATA MOVES	MOV A, Rr	F*	1/1	move register contents to A	(A)←(Rr)	r = 0–7	
	MOV A, @Rr	F0 F1	1/1	move RAM data, addressed by Rr, to A	(A)←((R0)) (A)←((R1))		
	MOV A, #data	23 data	2/2	move immediate data to A	(A)←data		
	MOV Rr, A	A*	1/1	move accumulator contents to register	(Rr)←(A)	r = 0–7	
	MOV @Rr, A	A0 A1	1/1	move accumulator contents to RAM location addressed by Rr	((R0))←(A) ((R1))←(A)		
	MOV Rr, #data	B* data	2/2	move immediate data to Rr	(Rr)←data	r = 0–7	
	MOV @rR, #data	B0 data B1 data	2/2	move immediate data to RAM location addressed by Rr	((R0))←data ((R1))←data		
	XCH A, Rr	2*	1/1	exchange accumulator contents with Rr	(A)↔(Rr)	r = 0–7	
	XCH A, @Rr	20 21	1/1	exchange accumulator contents with RAM data addressed by Rr	(A)↔((R0)) (A)↔((R1))		
	XCHD A, @Rr	30 31	1/1	exchange lower nibbles of A and RAM data addressed by Rr	$(A_{0-3})\leftrightarrow((R0_{0-3}))$ $(A_{0-3})\leftrightarrow((R1_{0-3}))$		
	MOV A, PSW	C7	1/1	move PSW contents to accumulator	(A)←(PSW)		
	MOV PSW, A	D7	1/1	move accumulator contents to PSW	(PSW)←(A)		3
	MOVP A, @A	A3	1/2	move indirectly addressed data in current page to A	(A)←((A))		
	MOVP3 A,@A	E3	1/2	move data in page 3 to A	(A)←((A))	in page 3	

	MOVX A,@Rr	80 81	1/2	move indirect the contents of external memory to A	(A)←((R))	r = 0–1	
	MOVX @Rr,A	90 91	1/2	move indirect the contents of A to external memory	((R))←(A)	r = 0–1	
FLAGS	CLR C	97	1/1	clear carry bit	(C)←0		2
	CPL C	A7	1/1	complement carry bit	(C)←NOT(C)		2
REGISTER	INC Rr	1*	1/1	increment register by 1	(Rr)←(Rr) + 1	r = 0–7	
	INC @Rr	10 11	1/1	increment RAM data, addressed by Rr, by 1	((R0))←((R0)) + 1 ((R1))←((R1)) + 1		
	DEC Rr	C*	1/1	decrement register by 1	(Rr)←(Rr) –1	r = 0–7	
BRANCH	JMP addr	● 4 address	2/2	unconditional jump within a 2 K bank	(PC_{8-10})←$addr_{8-10}$ (PC_{0-7})←$addr_{0-7}$ (PC_{11-12})←MBFF 0–1		
	JMPP @A	B3	1/2	indirect jump within a page	(PC_{0-7})←((A))		
	DJNZ Rr, addr	E* address	2/2	decrement Rr by 1 and jump if not zero to addr	(Rr)←(Rr) –1 if (Rr) not zero (PC_{0-7})←addr	r = 0–7	
	JF0 addr	B6 address	2/2	jump to addr if F0 = 1	if F0 = 1: (PC_{0-7})←addr		
	JF1 addr	76 address	2/2	jump to addr if F1 = 1	if F1 = 1: (PC_{0-7})←addr		
	JNI addr	86 address	2/2	jump to addr if $\overline{INT}$ = 0	if $\overline{INT}$ = 0: (PC_{0-7})←addr		
	JBb addr	▲ 2 address	2/2	jump to addr if Acc. bit b = 1	if b = 1: (PC_{0-7})←addr	b = 0–7	
	JC addr	F6 address	2/2	jump to addr if C = 1	if C = 1: (PC_{0-7})←addr		
	JNC addr	E6 address	2/2	jump to addr if C = 0	if C = 0: (PC_{0-7})←addr		
	JZ addr	C6 address	2/2	jump to addr if A = 0	if A = 0: (PC_{0-7})←addr		
	JNZ addr	96 address	2/2	jump to addr if A is NOT zero	if A ≠0: (PC_{0-7})←addr		
	JT0 addr	36 address	2/2	jump to addr if T0 = 1	if T0 = 1: (PC_{0-7})←addr		
	JNT0 addr	26 address	2/2	jump to addr if T0 = 0	if T0 = 0: (PC_{0-7})←addr		
	JT1 addr	56 address	2/2	jump to addr if T1 = 1	if T1 = 1: (PC_{0-7})←addr		
	JNT1 addr	46 address	2/2	jump to addr if T1 = 0	if T1 = 0: (PC_{0-7})←addr		
	JTF addr	16 address	2/2	jump to addr if Timer Flag = 1	if TF = 1: (PC_{0-7})←addr		4

	mnemonic	opcode (hex.)	bytes/ cycles	description	function	notes
TIMER/EVENT COUNTER	MOV A, T	42	1/1	move timer/event counter contents to accumulator	(A)←(T)	
	MOV T, A	62	1/1	move accumulator contents to timer/event counter	(T)←(A)	
	STRT CNT	45	1/1	start event counter		
	STRT T	55	1/1	start timer		
	STOP TCNT	65	1/1	stop timer/event counter		
	EN TCNTI	25	1/1	enable timer/event counter interrupt		
	DIS TCNTI	35	1/1	disable timer/event counter interrupt		
CONTROL	EN I	05	1/1	enable external interrupt		
	DIS I	15	1/1	disable external interrupt		
	SEL RB0	C5	1/1	select register bank 0	(RBS)←0	5
	SEL RB1	D5	1/1	select register bank 1	(RBS)←1	5
	SEL MB0	E5	1/1	select program memory bank 0	(MBFF0)←0, (MBFF1)←0	
	SEL MB1	F5	1/1	select program memory bank 1	(MBFF0)←1, (MBFF1)←0	
	ENT0 CLK	75	1/1	enable clock output onto T0		
SUBROUTINE	CALL addr	▲4 address	2/2	jump to subroutine	((SP))←(PC), ($PSW_{4, 6, 7}$) (SP)←(SP) + 1 (PC_{8-10})←$addr_{8-10}$ (PC_{0-7})←$addr_{0-7}$ (PC_{11-12})←MBFF 0–1	6
	RET	83	1/2	return from subroutine	(SP)←(SP) −1 (PC)←((SP))	6
	RETR	93	1/2	return from interrupt and restore bits 4, 6, 7 of PSW	(SP)←(SP) −1 ($PSW_{4, 6, 7}$) + (PC)←((SP))	6

INPUT/OUTPUT	OUTL BUS,A	02	1/2	output accumulator to BUS	(BUS)←A		
	IN A,Pp	09 0A	1/2	input port p data to accumulator	(A)←(P1) (A)←(P2)	p = 1–2	7
	INS A,BUS	08	1/2	input strobed BUS data into accumulator	(A)←(BUS)		
	OUTL Pp,A	39 3A	1/2	output accumulator data to port p	(P1)←(A) (P2)←(A)	p = 1–2	
	ANL BUS, # data	98	2/2	logical AND immediate data with BUS	(BUS)←(BUS) AND data		
	ANL Pp, # data	99 9A	2/2	AND port p data with immediate data	(P1)←(P1) AND data (P2)←(P2) AND data	p = 1–2	
	ORL Pp, # data	89 8A	2/2	OR port p data with immediate data	(P1)←(P1) OR data (P2)←(P2) OR data	p = 1–2	
	ORL BUS, # data	88	2/2	logical OR immediate data with BUS	(BUS)←(BUS) OR data		
	MOVD A,Pp	0C 0D 0E 0F	1/2	move contents of designated port (4–7) to A	(A_{0-3})←(Pp) (A_{4-7})←0	p = 4–7	
	MOVD Pp,A	3C 3D 3E 3F	1/1	move contents of A to designated port (4–7)	(Pp)←(A_{0-3})	p = 4–7	
	ANLD Pp,A	9C 9D 9E 9F	1/2	logical AND contents of A with designated port (4–7)	(Pp)←(Pp) AND (A_{0-3})	p = 4–7	
	ORLD Pp,A	8C 8D 8E 8F	1/1	logical OR contents of A with designated port (4–7)	(Pp)←(Pp) OR (A_{0-3})	p = 4–7	
	NOP	00	1/1	no operation			

Notes to Table 2.

1. PSW CY, AC affected
2. PSW CY affected
3. PSW PS affected
4. Execution of JTF and JNTF instruction resets the Timer Flag (TF).
5. PSW RBS affected
6. PSW SP_0, SP_1, SP_2 affected
7. (A) = 111 P23, P22, P21, P20.
8. (S1) has a different meaning for read and write operation, see serial I/O interface.

* : 8, 9, A, B, C, D, E, F
● : 0, 2, 4, 6, 8, A, C, E
▲ : 1, 3, 5, 7, 9, B, D, F

Table 3 Instruction timing (see also Figs 11 and 12)

	cycle 1					cycle 2				
instruction	S1	S2	S3	S4	S5	S1	S2	S3	S4	S5
IN A,P	fetch instruction	increment program counter	–	increment timer	–	–	read port	* –	–	–
OUTL P,A	¦	¦	–	¦	output to port	–	–	* –	–	–
ANL P,#data	¦	* ¦	–	¦	read port	fetch immediate data	–	*increment program counter	output to port	–
ORL P,#data	¦	* ¦	–	¦	read port	fetch immediate data	–	*increment program counter	output to port	–
INS A,BUS	¦	¦	–	¦	–	–	read port	* –	–	–
OUTL BUS,A	¦	¦	–	¦	output to port	–	–	* –	–	–
ANL BUS,#data	¦	* ¦	–	¦	read port	fetch immediate data	–	*increment program counter	output to port	–
ORL BUS,#data	¦	* ¦	–	¦	read port	fetch immediate data	–	*increment program counter	output to port	–
MOVX @R,A	¦	¦	output RAM address	¦	output data to RAM	–	–	* –	–	–
MOVX A,@R	¦	¦	output RAM address	¦	–	–	read data	* –	–	–
MOVD A,P	fetch instruction	increment program counter	output opcode/ address	increment timer	–	–	read P2 lower	* –	–	–

	cycle 1					cycle 2				
instruction	S1	S2	S3	S4	S5	S1	S2	S3	S4	S5
MOVD P,A	fetch instruction	increment program counter	output opcode/ address	increment timer	output data to P2 lower	–	–	* –	–	–
ANLD P,A			output opcode/ address		output data	–	–	* –	–	–
ORLD P,A			output opcode/ address		output data	–	–	* –	–	–
J (conditional)		*	sample condition	increment timer	–	fetch immediate data	–	* opdata program counter	–	–
STRT CNT STRT T		*	–	–	start counter					
STOP TCNT		*	–	–	stop counter					
EN I		*	–	enable interrupt	–					
DIS I		*	–	disable interrupt	–					
ENT0 CLK	fetch instruction	* increment program counter	–	enable clock	–					

* Valid instruction addresses are output at this time if external program memory is being accessed.

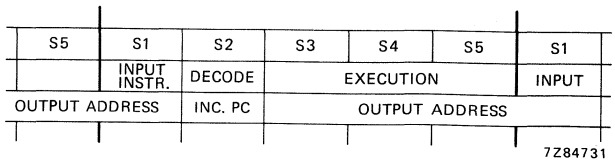


Fig. 11 Instruction cycle.

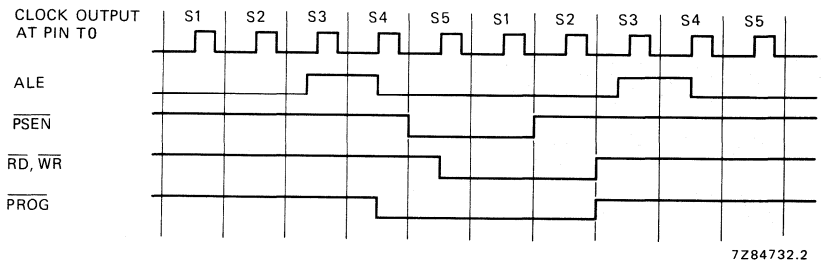


Fig. 12 Instruction cycle timing.

Table 4 Instruction map.

first hex. char. of opcode	second hexadicimal character of opcode															
	0 1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NOP	OUTL BUS,A	ADD A, #data	JMP page 0	EN I		DEC A	INS A,BUS	IN A, Pp 1	2		MOVD A,Pp 4	5	6	7
1	INC @Rr 0 1	JB0 addr	ADDC A, #data	CALL page 0	DIS I	JTF addr	INC A	INC Rr 0	1	2	3	4	5	6	7
2	XCH A, @Rr 0 1		MOV A, #data	JMP page 1	EN TCNTI	JNT0 addr	CLR A	XCH A,Rr 0	1	2	3	4	5	6	7
3	XCHD A, @Rr 0 1	JB1 addr		CALL page 1	DIS TCNTI	JT0 addr	CPL A		OUTL Pp,A 1	2		MOVD Pp,A 4	5	6	7
4	ORL A, @Rr 0 1	MOV A, T	ORL A, #data	JMP page 2	STRT CNT	JNT1 addr	SWAP A	ORL A,Rr 0	1	2	3	4	5	6	7
5	ANL A, @Rr 0 1	JB2 addr	ANL A, #data	CALL page 2	STRT T	JT1 addr	DA, A	ANL A,Rr 0	1	2	3	4	5	6	7
6	ADD A, @Rr 0 1	MOV T,A		JMP page 3	STOP TCNT		RRC A	ADD A,Rr 0	1	2	3	4	5	6	7
7	ADDC A, @Rr 0 1	JB3 addr		CALL page 3	ENT0 CLK	JF1 addr	RR A	ADDC A,Rr 0	1	2	3	4	5	6	7
8	MOVX A, @Rr 0 1		RET	JMP page 4	CLR F0	JN1 addr		ORL BUS, #data	ORL Pp, #data 1	2		ORLD Pp,A 4	5	6	7
9	MOVX @Rr,A 0 1	JB4 addr	RETR	CALL page 4	CPL F0	JNZ addr	CLR C	ANL BUS, #data	ANP Pp, #data 1	2		ANLD Pp,A 4	5	6	7
A	MOV @Rr, A 0 1		MOVP A, @A	JMP page 5	CLR F1		CPL C	MOV Rr,A 0	1	2	3	4	5	6	7
B	MOV @Rr, #data 0 1	JB5 addr	JMPP @A	CALL page 5	CPL F1	JF0 addr		MOV R, #data 0	1	2	3	4	5	6	7
C				JMP page 6	SEL RB0	JZ addr	MOV A, PSW	DEC Rr 0	1	2	3	4	5	6	7
D	XRL A, @Rr 0 1	JB6 addr	XRL A, #data	CALL page 6	SEL RB1		MOV PSW, A	XRL A,Rr 0	1	2	3	4	5	6	7
E			MOVP3 A @A	JMP page 7	SEL MB0	JNC addr	RL A	DJNZ Rr, addr 0	1	2	3	4	5	6	7
F	MOV A, @Rr 0 1	JB7 addr		CALL page 7	SEL MB1	JC addr	RLC A	MOV A,Rr 0	1	2	3	4	5	6	7

RATINGS

Limiting values in accordance with the Absolute Maximum System (IEC 134)

Input voltage with respect to V_{SS}			
except input EA	V_I	−0,5 to + 7	V
input EA	V_I	−0,5 to + 12	V
D.C. current into any input or output	$\pm I_I, \pm I_O$	max. 10	mA
Total power dissipation	P_{tot}	max. 1	W
Storage temperature range	T_{stg}	−65 to +150	°C
Operating ambient temperature range	T_{amb}	see Table 5	

D.C. CHARACTERISTICS (MAB8048H/35HL; MAB8049H/39HL; MAB8050H/40HL)

$V_{CC} = V_{DD} = 5$ V (± 10%); $V_{SS} = 0$ V; $T_{amb} = 0$ to + 70 °C; all voltages with respect to V_{SS} unless otherwise specified

parameter	symbol	min.	typ.	max.	unit
Supply current at $V_{DD} = 5$ V ± 10%; $V_{SS} = V_{CC} = 0$ V					
MAB8048H/35HL	I_{DD}	–	–	6	mA
MAB8049H/39HL	I_{DD}	–	–	8	mA
MAB8050H/40HL	I_{DD}	–	–	15	mA
Supply current (total) at $V_{DD} = V_{CC} = 5$ V ± 10%; $V_{SS} = 0$ V					
MAB8048H/35HL	$I_{DD} + I_{CC}$	–	–	80	mA
MAB8049H/39HL	$I_{DD} + I_{CC}$	–	–	90	mA
MAB8050H/40HL	$I_{DD} + I_{CC}$	–	–	100	mA
Inputs					
Input voltage LOW all inputs except XTAL 1, XTAL 2, $\overline{\text{RESET}}$	V_{IL}	–0,5	–	0,8	V
Input voltage LOW XTAL 1, XTAL 2, $\overline{\text{RESET}}$	V_{IL1}	–0,5	–	0,6	V
Input voltage HIGH all inputs except XTAL 1, XTAL 2, $\overline{\text{RESET}}$	V_{IH}	2,0	–	V_{CC}	V
Input voltage HIGH XTAL 1, XTAL 2, $\overline{\text{RESET}}$	V_{IH1}	3,8	–	V_{CC}	V
Outputs					
Output voltage LOW (DB0 to DB7) at $I_{OL} = 2$ mA	V_{OL}	–	–	0,45	V
Output voltage LOW ($\overline{\text{RD}}$, $\overline{\text{WR}}$, $\overline{\text{PSEN}}$, ALE) at $I_{OL1} = 1{,}8$ mA	V_{OL1}	–	–	0,45	V
Output voltage LOW ($\overline{\text{PROG}}$) at $I_{OL2} = 1$ mA	V_{OL2}	–	–	0,45	V
Output voltage LOW (all other outputs) at $I_{OL3} = 1{,}6$ mA	V_{OL3}	–	–	0,45	V
Output voltage HIGH (DB0 to DB7) at $-I_{OH} = 400\ \mu$A	V_{OH}	2,4	–	–	V
Output voltage HIGH ($\overline{\text{RD}}$, $\overline{\text{WR}}$, $\overline{\text{PSEN}}$, ALE) at $-I_{OH1} = 100\ \mu$A	V_{OH1}	2,4	–	–	V
Output voltage HIGH (all other outputs) at $-I_{OH} = 40\ \mu$A	V_{OL2}	2,4	–	–	V

D.C. CHARACTERISTICS (continued)

parameter	symbol	min.	typ.	max.	unit
Input leakage current (T1, $\overline{INT}$) at $V_{SS} < V_I < V_{CC}$	$\pm I_{IL}$	–	–	10	μA
Output leakage current (DB0 to DB7; high impedance) at $V_{SS} + 0{,}45\ V < V_I < V_{CC}$	$\pm I_{OZ}$	–	–	10	μA
LOW input load current (P10 to P17, P20 to P27 EA, $\overline{SS}$) at $V_{SS} + 0{,}45\ V < V_I < V_{CC}$	I_{LI}	–	–	−0,5	mA
LOW input load current ($\overline{RESET}$) at $V_{SS} < V_I < V_{CC}$	I_{LI1}	0,02	–	−0,3	mA

D.C. CHARACTERISTICS

MAF8048H/35HL; MAF8049H/39HL; MAF8050H/40HL (at $T_{amb} = -40$ to + 85 °C)
MAF80A48H/A35HL; MAF80A49H/A39HL; MAF80A50H/A40HL (at $T_{amb} = -40$ to + 110 °C)

$V_{CC} = V_{DD} = 5\ V$ (± 10%); $V_{SS} = 0\ V$; T_{amb} as above; all voltages with respect to V_{SS} unless otherwise specified

parameter	symbol	min.	typ.	max.	unit
Supply current at $V_{DD} = 5\ V \pm 10\%$; $V_{SS} = V_{CC} = 0\ V$					
MAF8048H/35HL; MAF80A48H/A35HL	I_{DD}	–	–	8	mA
MAF8049H/39HL; MAF80A49H/A39HL	I_{DD}	–	–	10	mA
MAF8050H/40HL; MAF80A50H/A40HL	I_{DD}	–	–	18	mA
Supply current (total) at $V_{DD} = 5\ V \pm 10\%$; $V_{SS} = V_{CC} = 0\ V$					
MAF8048H/35HL; MAF80A48H/A35HL	$I_{DD} + I_{CC}$	–	–	90	mA
MAF8049H/39HL; MAF80A49H/A39HL	$I_{DD} + I_{CC}$	–	–	100	mA
MAF8050H/40HL; MAF80A50H/A40HL	$I_{DD} + I_{CC}$	–	–	120	mA
Inputs					
Input voltage HIGH all inputs except XTAL 1, XTAL 2, $\overline{RESET}$	V_{IH}	2,2	–	V_{CC}	V
LOW input load current (P10 to P17, P20 to P27 EA, $\overline{SS}$) at $V_{SS} + 0{,}45\ V < V_I < V_{CC}$	$-I_{LI}$	–	–	0,6	mA

A.C. CHARACTERISTICS (MAB8048H/35HL; MAB8049H/39HL; MAB8050H/40HL)

$V_{CC} = V_{DD} = 5$ V (± 10%); $V_{SS} = 0$ V; $T_{amb} = 0$ to + 70 °C; note 1.
See waveforms Figs 14, 15, 16, 17 and 18

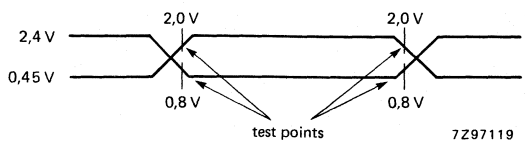
parameter	symbol	$f(t_{CL})$ (note 2)	11 MHz		unit
			min.	max.	
Clock period (note 2)	t_{CL}	$1/(f_{XTAL})$	90,9	1000	ns
ALE pulse duration	t_{LL}	$3{,}5t_{CL}-170$	150	–	ns
Address set-up time to ALE (note 3)	t_{AL}	$2t_{CL}-110$	70	–	ns
Address hold time after ALE	t_{LA}	$t_{CL}-40$	50	–	ns
Control pulse duration $\overline{RD}$, $\overline{WR}$	t_{CC1}	$7{,}5t_{CL}-200$	480	–	ns
Control pulse duration $\overline{PSEN}$	t_{CC2}	$6t_{CL}-200$	350	–	ns
Data set-up time before $\overline{WR}$	t_{DW}	$6{,}5t_{CL}-200$	390	–	ns
Data set-up time after $\overline{WR}$	t_{WD}	$t_{CL}-50$	40	–	ns
Data hold time $\overline{RD}$, $\overline{PSEN}$	t_{DR}	$1{,}5t_{CL}-30$	0	110	ns
$\overline{RD}$ to data input	t_{RD1}	$6t_{CL}-170$	–	375	ns
$\overline{PSEN}$ to data input	t_{RD2}	$4{,}5t_{CL}-170$	–	240	ns
Address set-up time to $\overline{WR}$	t_{AW}	$5t_{CL}-150$	300	–	ns
Address set-up time to data input ($\overline{RD}$)	t_{AD1}	$10{,}5t_{CL}-220$	–	730	ns
Address set-up time to data input ($\overline{PSEN}$)	t_{AD2}	$7{,}5t_{CL}-200$	–	460	ns
Address floating to $\overline{RD}$, $\overline{WR}$	t_{AFC1}	$2t_{CL}-40$	140	–	ns
Address floating to $\overline{PSEN}$ (note 3)	t_{AFC2}	$5t_{CL}-40$	10	–	ns
ALE to control pulse $\overline{RD}$, $\overline{WR}$	t_{LAFC1}	$3t_{CL}-75$	200	–	ns
ALE to control pulse $\overline{PSEN}$	t_{LAFC2}	$1{,}5t_{CL}-75$	60	–	ns
Control pulse to ALE $\overline{RD}$, $\overline{WR}$, $\overline{PROG}$	t_{CA1}	$t_{CL}-40$	50	–	ns
Control pulse to ALE $\overline{PSEN}$	t_{CA2}	$4t_{CL}-40$	320	–	ns
Port control set-up to $\overline{PROG}$	t_{CP}	$1{,}5t_{CL}-80$	50	–	ns
Port control hold to $\overline{PROG}$	t_{PC}	$4t_{CL}-260$	100	–	ns
$\overline{PROG}$ to Port 2 input must be valid	t_{PR}	$8{,}5t_{CL}-120$	–	650	ns
Input data hold time from $\overline{PROG}$	t_{PF}	$1{,}5t_{CL}$	0	150	ns
Output data set-up time	t_{DP}	$6t_{CL}-290$	250	–	ns
Output data hold time	t_{PD}	$1{,}5t_{CL}-90$	40	–	ns
$\overline{PROG}$ pulse duration	t_{PP}	$10{,}5t_{CL}-250$	700	–	ns
Port 2 I/O data set-up time to ALE	t_{PL}	$4t_{CL}-200$	160	–	ns
Port 2 I/O data hold time to ALE	t_{LP}	$1{,}5t_{CL}-120$	15	–	ns

A.C. CHARACTERISTICS (continued)

parameter	symbol	f(t_{CL}) (note 2)	11 MHz		unit
			min.	max.	
Port output from ALE	t_{PV}	$4{,}5t_{CL}+100$	–	510	ns
T0 repetition rate	t_{0PRR}	$3t_{CL}$	270	–	ns
Cycle time	t_{CY}	$1/(f_{XTAL} \times 15)$	1,36	15	μs
MAF80A48H/A35HL; MAF80A49H/A39HL; MAF80A50H/A40HL					
→ Clock period (note 2)	t_{CL}	$1/(f_{XTAL})$	90,9	1000	ns

Notes to A.C. characteristics

1. Control outputs: C_L = 80 pF.
   Bus outputs: C_L = 150 pF.
2. f(t_{CL}) assumes 50% duty cycle on XTAL 1 and XTAL 2; minimum frequency = 1 MHz.
3. Bus high-impedance load: 20 pF.



A.C. testing inputs are driven at 2,4 V for a logic 1 and 0,45 V for a logic 0. Output timing measurements are taken 2,0 V for a logic 1 and 0,8 V for a logic 0.

Fig. 13 A.C. testing input, output waveform.

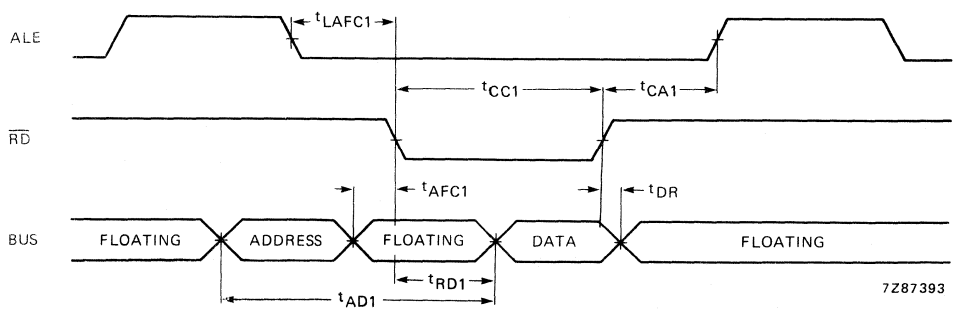


Fig. 14 Read from external data memory.

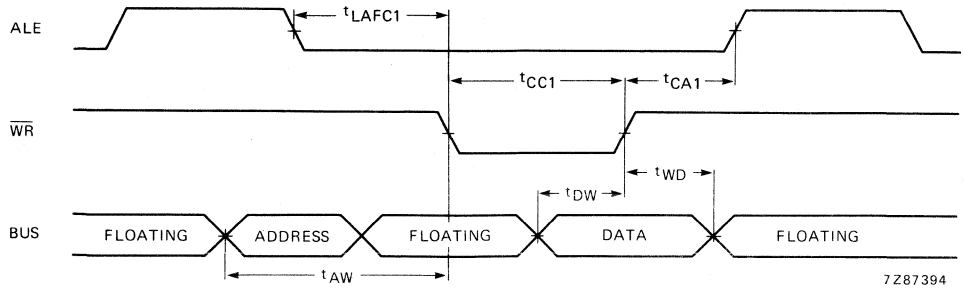


Fig. 15 Write to external memory.

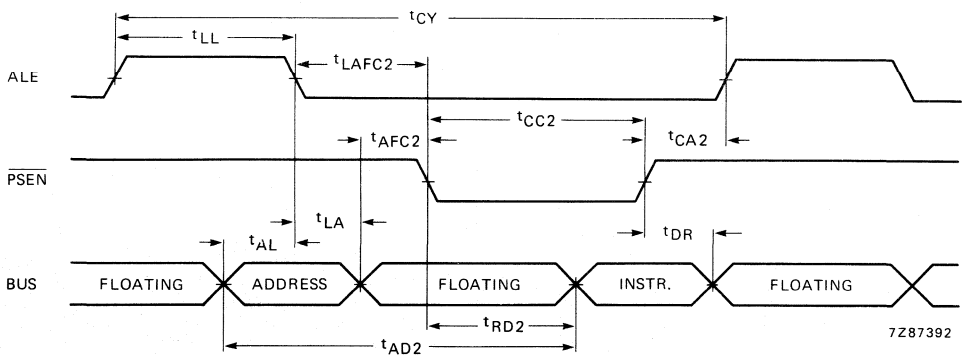


Fig. 16 Instruction fetch from program memory.

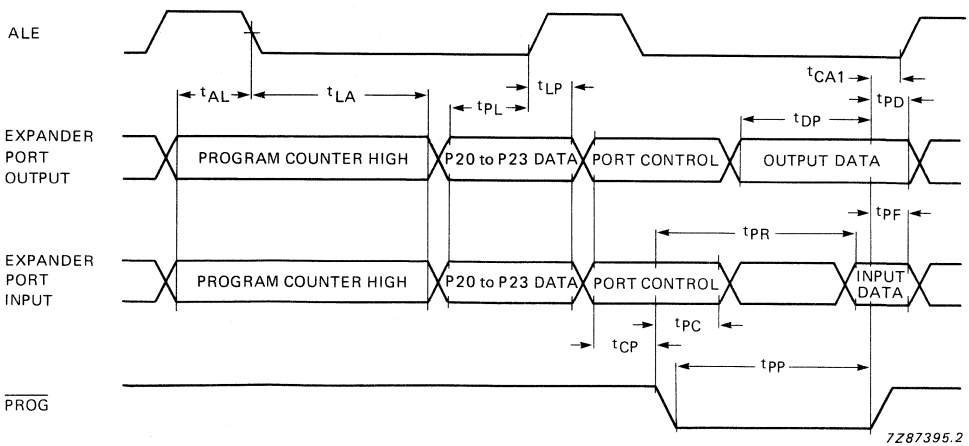


Fig. 17 Port 2 timing.

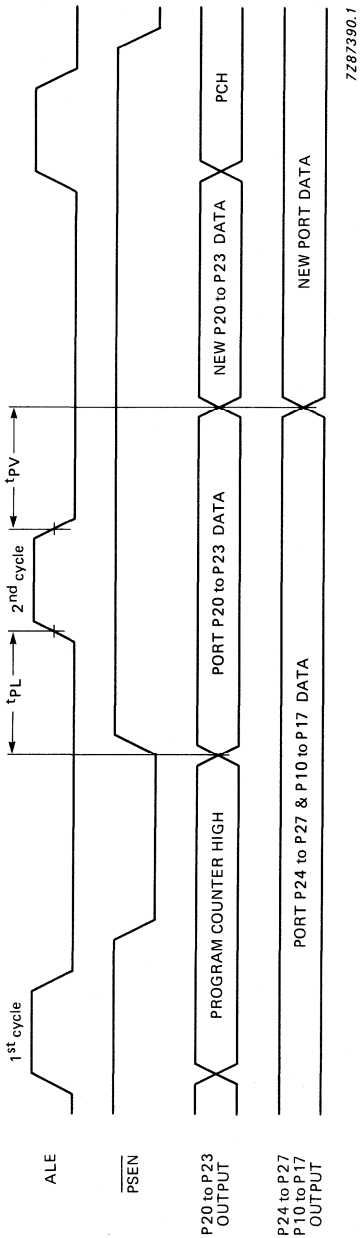


Fig. 18 I/O port timing.

SINGLE-CHIP 8-BIT MICROCONTROLLER

DESCRIPTION

The MAB84X1 family of microcontrollers is fabricated in NMOS. The family consists of 5 devices:

- MAB8401 — 128 bytes RAM, external program memory, with 8-bit LED-driver (10mA), emulation of MAB/F8422/42* possible
- MAB/MAF8411 — 1K byte ROM/64 bytes RAM plus 8-bit LED-driver
- MAB/MAF8421 — 2K bytes ROM/64 bytes RAM plus 8-bit LED-driver
- MAB/MAF8441 — 4K bytes ROM/128 bytes RAM plus 8-bit LED-driver
- MAB/MAF8461 — 6K bytes ROM/128 bytes RAM plus 8-bit LED-driver

Each version has 20 quasi-bidirectional I/O port lines, one serial I/O line, one single-level vectored interrupt, an 8-bit timer event counter and on-board clock oscillator and clock circuits. Two 20-pin versions, MAB/F8422 and MAB/F8442* are also available.

This microcontroller family is designed to be an efficient controller as well as an arithmetic processor. The instruction set is based on that of the MAB8048. The microcontrollers have extensive bit handling abilities and facilities for both binary and BCD arithmetic.

For detailed information see the "8-bit Single-chip Microcontrollers user manual".

* See data sheet on MAB/F8422/42.

Features

- 8-bit: CPU, ROM, RAM and I/O in a single 28-lead DIL package
- 1K, 2K, 4K or 6K ROM bytes plus a ROM-less version
- 64 or 128 RAM bytes
- 20 quasi-bidirectional I/O port lines
- Two testable inputs: one of which can be used to detect zero cross-over, the other is also the external interrupt input
- Single level vectored interrupts: external, timer/event counter, serial I/O
- Serial I/O that can be used in single or multi-master systems (serial I/O data via an existing port line and clock via a dedicated line)
- 8-bit programmable timer/event counter
- Internal oscillator, generated with inductor, crystal, ceramic resonator or external source
- Over 80 instructions (based on MAB8048) all of 1 or 2 cycles
- Single 5 V power supply (± 10%)
- Operating temperature ranges:

0 to + 70 °C	MAB84X1 family
−40 to + 85 °C	MAF84X1 family only
−40 to + 110 °C	MAF84AX1 family only

PACKAGE OUTLINES

MAB8401B: 28-lead 'Piggy-back' package (with up to 28-pin EPROM on top).
MAB8401WP: 68-lead plastic leaded chip-carrier (PLCC) (SOT-188A).
MAB/MAF8411/21/41/61P: 28-lead DIL; plastic (SOT-117).
MAF84A11/21/41/61P: 28-lead DIL; plastic (SOT-117).
MAB8411/21/41/61T: 28-lead mini-pack; plastic (SO-28; SOT-136A).

PINNING

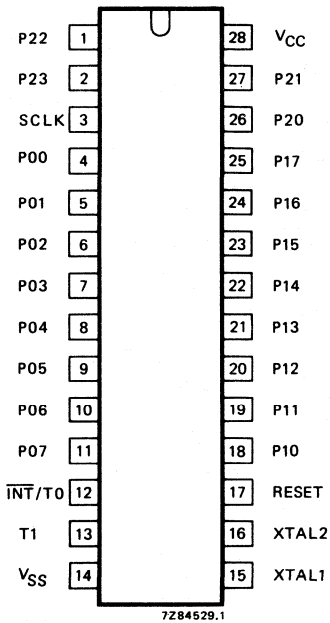
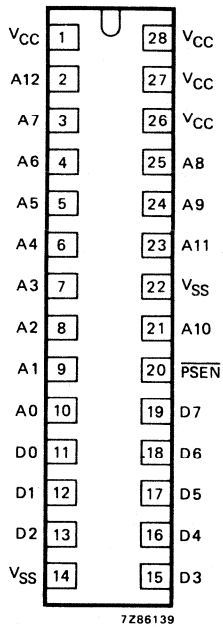


Fig. 1 Pinning diagram for mask-programmable devices MAB8411, MAB8421, MAB8441, MAB8461 and for MAB8401 'Piggy-back' version bottom pinning (for top pinning see Fig. 2).

PINNING DESIGNATION

V_{SS}	14	**Ground**
V_{CC}	28	**Power supply,** + 5 V
P00 – P07	4 – 11	**Port 0,** 8-bit quasi-bidirectional I/O port
P10 – P17	18 – 25	**Port 1,** 8-bit quasi-bidirectional I/O port with 8-bit LED driver
P20 – P23	26, 27, 1, 2	**Port 2,** 4-bit quasi-bidirectional I/O port, P23 is the serial data I/O in serial I/O mode
SCLK	3	Bidirectional clock for serial I/O
$\overline{INT}$/T0	12	External interrupt input (sensitive to a negative-going edge min LOW > 7 clock pulses, min HIGH > 4 clock pulses), testable using the JT0 or JNT0 instructions.
T1	13	Input pin, testable using the JT1 or JNT1 instructions. It can be designated as event counter input using the STRT CNT instruction. It can also be used to detect zero cross-over of slowly moving a.c. inputs.
RESET	17	Input to initialize the processor (active HIGH).
XTAL1	15	Connection to timing component (crystal) that determines the frequency of the internal oscillator. It is also the input for an external clock source.
XTAL2	16	Connection to other side of the timing component.

MAB8401B (top pinning)



PIN DESIGNATION

designation	pin	function
V_{SS}	14, 22	**Ground**
V_{CC}	1, 26–28	**Power supply,** + 5 V
A0–A12	10–3, 25, 24, 21, 23, 2	Address outputs ←
D0–D7	11–13, 15–19	Data inputs
$\overline{PSEN}$	20	Program store enable

Fig. 2 Pinning diagram for MAB8401B 'Piggy-back' version top pinning (for bottom pinning see Fig. 1); to access a 2732 or 2764 EPROM.

Note

Access times for ROMS/EPROMS to be below 1 μs.

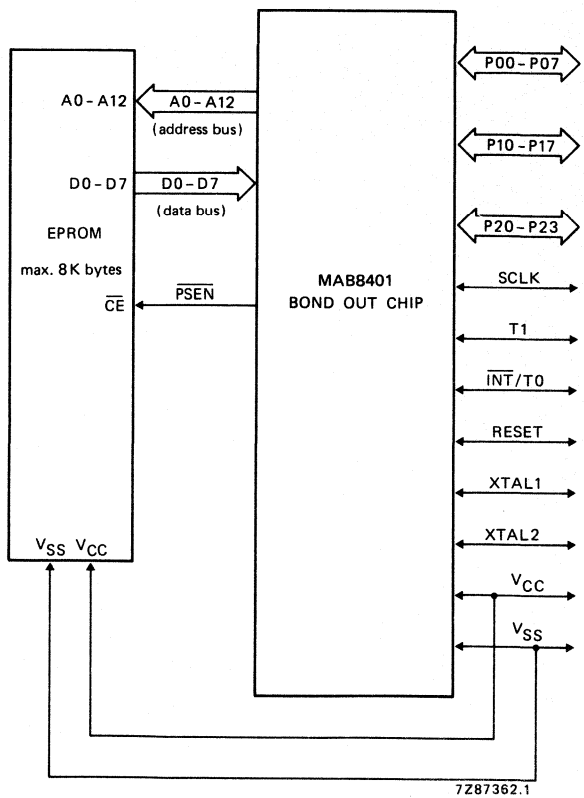


Fig. 2a Connection of EPROM to 'Piggy-back' package MAB8401B.

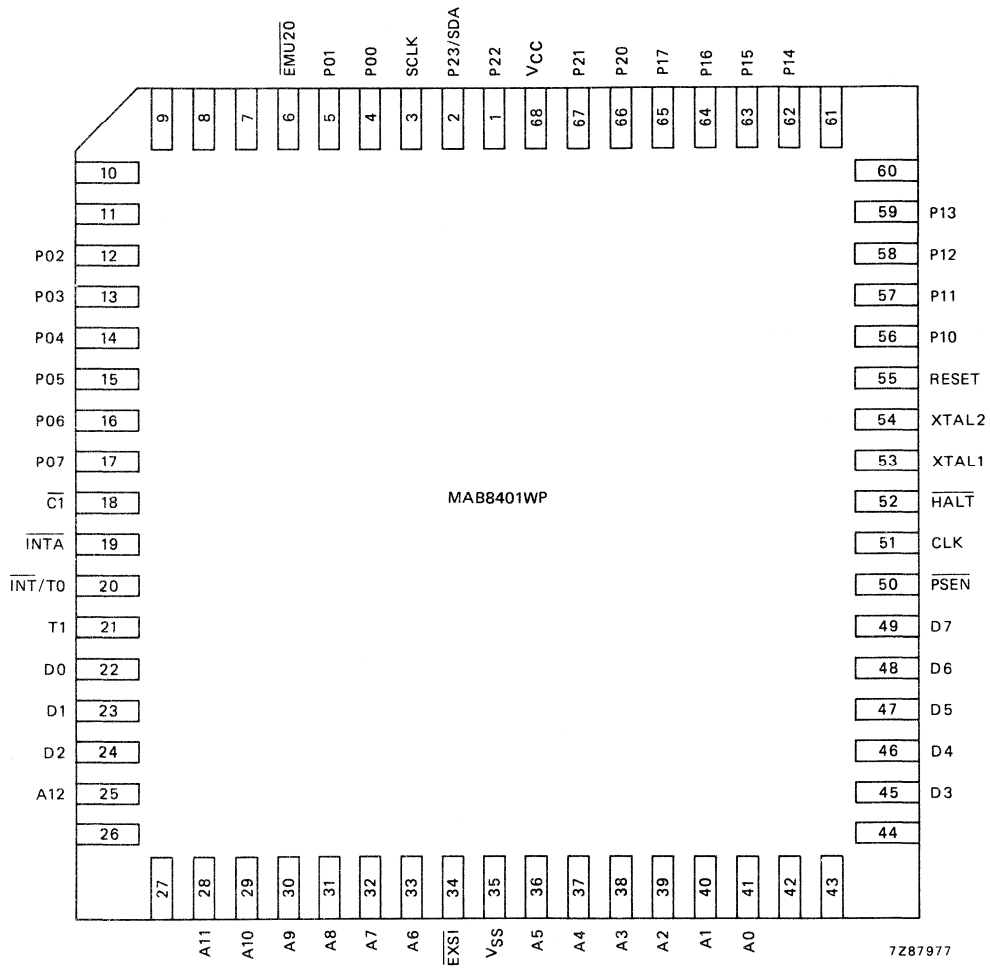
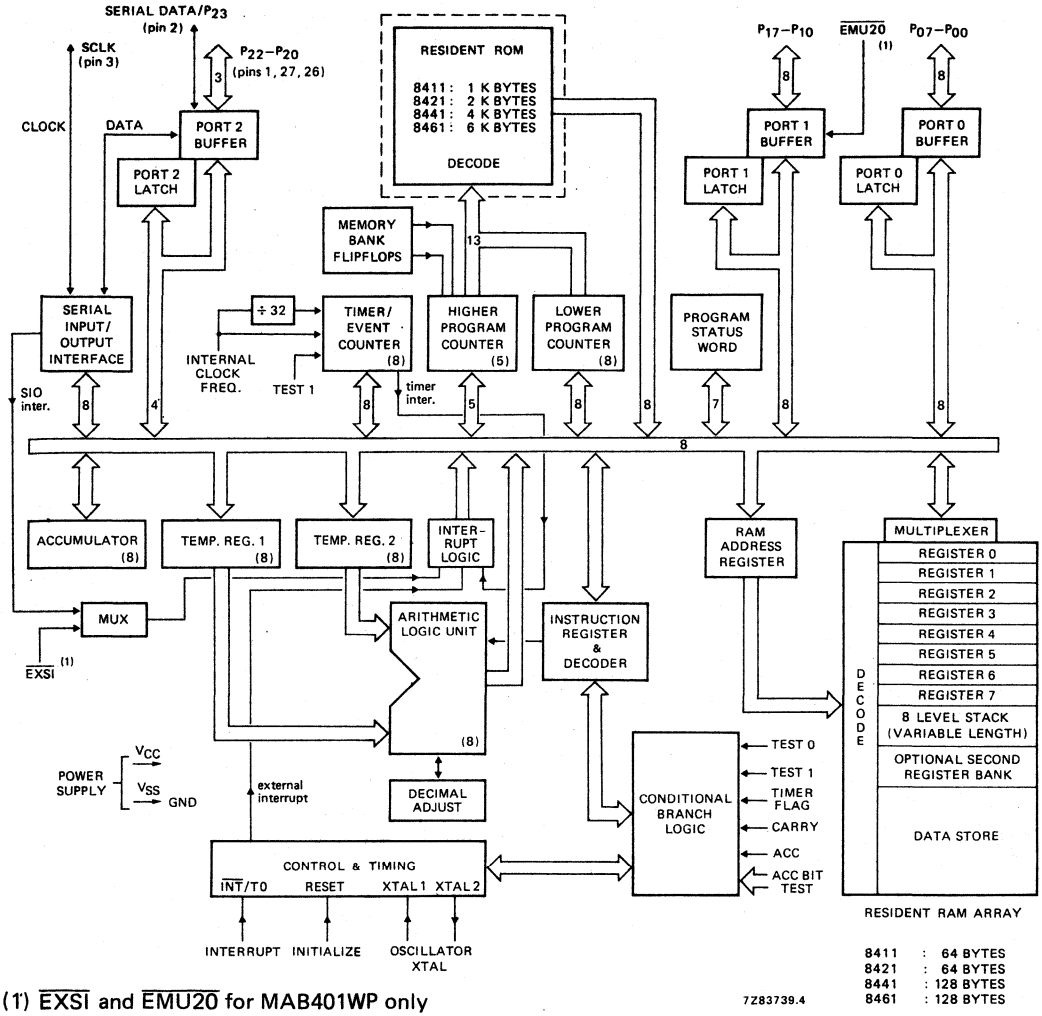


Fig. 3 Pinning diagram; PLCC.

CHIP CARRIER DESIGNATION

designation	pad no.	function
V_{SS}	35	**Ground**
V_{CC}	68	**Power supply,** + 5 V
P00–P07	4–5, 12–17	**Port 0,** 8-bit quasi-bidirectional I/O port
P10–P17	56–59, 62–65	**Port 1,** 8-bit quasi-bidirectional I/O port
P20–P22	66, 67, 1	**Port 2,** 4-bit quasi-bidirectional I/O port; P23 is the serial
P23/SDA	2	data I/O in series I/O mode
SCLK	3	Bidirectional clock for serial I/O
$\overline{INT}$/T0	20	External interrupt input (sensitive to a negative-going edge), testable using the JT0 or JNT0 instructions

T1	21	Input pin, testable using the JT1 or JNT1 instructions. It can be designated as event counter input using the STRT CNT inctruction. It can also be used to detect zero cross-over of slowly moving a.c. inputs.
RESET	55	Input to initialize the processor (active HIGH)
XTAL1	53	Connection to timing component (e.g. crystal) that determines the frequency of the internal oscillator. It is also the input for an external clock source.
XTAL2	54	Connection to other side of the timing component
$\overline{\text{EXSI}}$	34	External serial I/O interrupt (active-LOW) for emulation of MAB/F8422/42.
A0–A12	41–36, 33–28 25	Program memory address outputs (active HIGH); A0 = LSB, A12 = MSB. Address output change after begin Phi3 of TS8.
D0–D7	22–24, 45–49	Data input lines (active HIGH) used for reading external program memory. D0 = LSB, D7 = MSB.
CLK	51	Clock output buffered from XTAL2. On the positive-going edge the (internal) Phi clock goes HIGH.
$\overline{\text{PSEN}}$	50	Program store enable. This signal is used for enabling the external EPROM (e.g. on the 'Piggy-back' version). For emulation, it enables the emulation memory and it indicates machine cycles. Active LOW during TS9, TS10 of each machine cycle and TS1 of the following machine cycle.
$\overline{\text{C1}}$	18	Cycle 1 indication output (active LOW). During emulation, this signal indicates the opcode fetch cycle (useful for external instruction decoding, real-time trace). Active from start of TS10 of the cycle preceding cycle 1, until the start of TS10 of cycle 1.
$\overline{\text{HALT}}$	52	Halt input (active LOW). If activated, the current instruction is finished and the microcontroller stops execution (HALT mode). The next program counter address is available on the address bus. Program counter and timer/event counter are no longer updated. The serial I/O finishes the current transmit/receive action and goes into the idle state. Interrupts are *not* sampled in the HALT mode, they are only sampled when the microcontroller is running. Interrupt routines can be single-stepped as a normal program.
$\overline{\text{INTA}}$	19	Interrupt acknowledge output (active LOW). It indicates any interrupt acception. Active from start of TS8 of the interrupted cycle, until start of TS7 of the second cycle of the (internally forced 'CALL vector address' instruction. During $\overline{\text{INTA}}$ active, the address bus shows the address that has been saved in the stack (return address); the C1 output indicates opcode fetch cycles as if a user CALL was executed.
$\overline{\text{EMU20}}$	6	Emulate 20-pin version MAB/F8422/42 (active-LOW).



(1) $\overline{EXSI}$ and $\overline{EMU20}$ for MAB401WP only

Fig. 4a Block diagram of the MAB84X1 family.

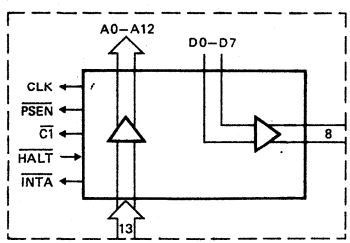


Fig. 4b Replacement for dotted part in Fig. 4a for the MAB8401WP bond-out version.

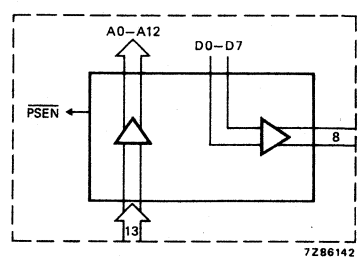


Fig. 4c Replacement of dotted part in Fig. 4a for the MAB8401B 'Piggy-back' version.

FUNCTIONAL DESCRIPTION (for more detail see Microcontroller Users Manual)

Bond-out version MAB8401WP

The bond-out version is a microcontroller that contains no on-board ROM, but has all address and data lines brought out to access an external ROM or EPROM. So, this version has more pins than the standard microcontrollers with on-board ROM. It has all the features of the other members of the MAB84X1-family, including emulation facilities for the MAB/F8422/42 (20-pin version). It can address 8K bytes of external ROM. The RAM has 128 bytes.

Piggy-back version MAB8401B

The Piggy-back version is a special package that has standard pinning to the bottom which facilitates insertion as a mask-programmed device. An EPROM is mounted on top in an additional socket. Thus, the total package height is greater than the standard DIL package. Emulation of the 8422/42 is not possible.

Program and data memory

The program memory (ROM) is mask-programmed at our factory. Because the MAB84X1-family offers a range of ROM capacities to suit the application, ROM expansion is not required. Figure 5 shows the program memory map. Program memory is arranged in banks of 2K bytes, that are selected by SEL MB instructions.

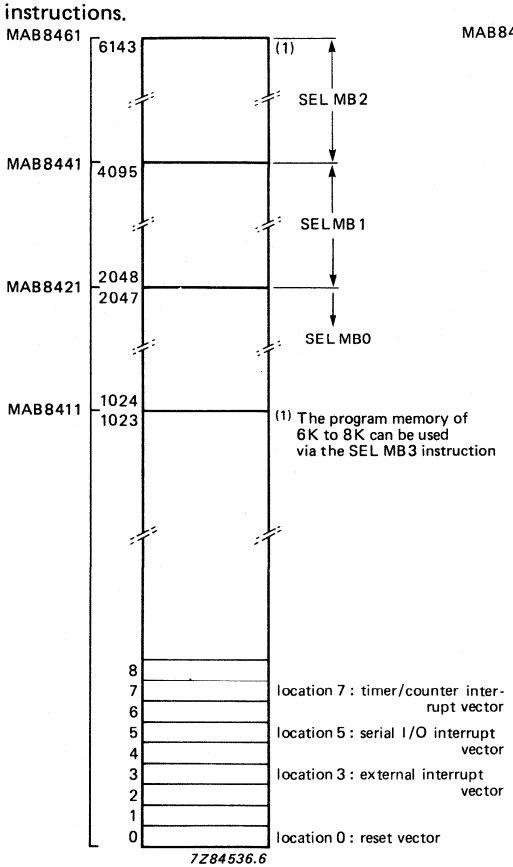


Fig. 5 The program memory map.

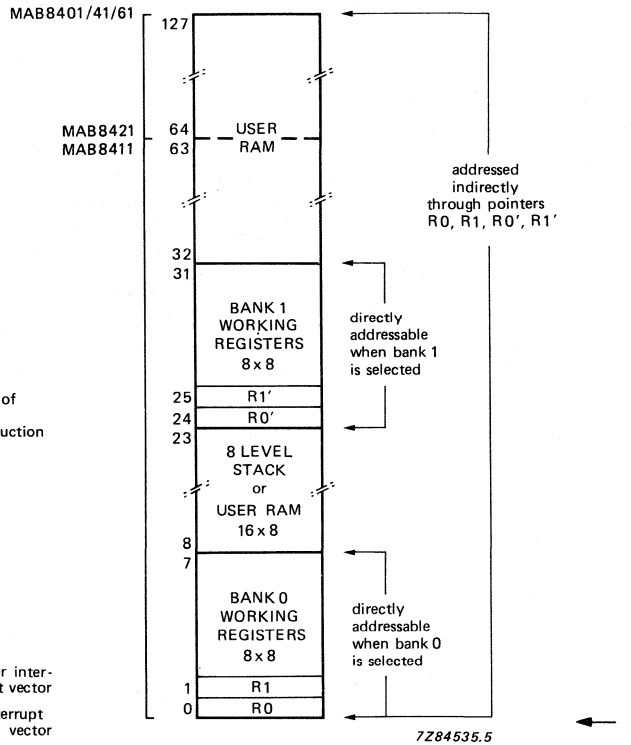


Fig. 6 The data memory map.

FUNCTIONAL DESCRIPTION (continued)

The data memory (RAM) consists of 64 or 128 bytes (8-bit words). All locations are indirectly addressable using RAM pointer registers and up to 16 designated location can be addressed directly. The memory also includes an 8-level program counter stack addressed by a 3-bit stack pointer. Figure 6 shows the data memory map.

On-chip peripheral functions

In addition to the CPU and memories, an interrupt system, I/O facilities, and an 8-bit timer/event counter are integrated on-chip to assist the CPU in repetitions, complicated or time-critical tasks. The I/O facilities include the I/O pins, parallel ports and a serial I/O port, consisting of a data line SDA shared with a parallel port line (P23), and a dedicated clock line SCLK.

I/O facilities

The MAB84XX family has 23 I/O lines arranged as:

- Two parallel ports of 8 lines (P00–P07, P10–P17). Each line of Port 1 can sink 10 mA.
- A parallel port of 4 lines (P20–P23)
- A serial I/O consisting of a data line shared with a parallel port line (P23) and a separate clock line SCLK;
- An external interrupt and test input $\overline{\text{INT}}$/T0, which when used as a test input can be tested by the conditional jump instructions JT0 or JNT0;
- A test input T1, which can alter program sequences when tested by conditional jump instructions JT1 or JNT1. T1 can also be used as an input to the timer/event counter or to detect zero cross-over of slowly moving a.c. signals.

All parallel port lines are available in three optional output configurations (except P23 – option 1 only):

- Option 1; open drain output without pull-up transistor (Fig. 7(a))
- Option 2; open drain output with pull-up transistor (Fig. 7(b))
- Option 3; push-pull output with pull-up transistor (Fig. 7(c))

If the inputs and outputs on a port are mixed (mixed-mode), the inputs should be options 1 or 2 but not option 3. This prevents cross-currents via TR2 and an external connection to ground, while switching the output on the same port and in parallel, masking the inputs with logic '1' s.

The MAB84X1 family serial I/O interface has been designed to eliminate the heavy processing load imposed upon a normal microcontroller performing serial data transfer. Whereas a normal microcontroller must regularly monitor the serial data bus for the presence of data, the serial I/O interface detects, receives and converts the serial data stream into a parallel format without interrupting the execution of the current program. An interrupt is sent to the microcontroller only when a complete byte is received. Then, the microcontroller reads the data byte in one instruction. Likewise, for transmission, the serial I/O interface performs parallel to serial conversion and subsequent serial output of the data and the microcontroller is only interrupted in the execution of its programmed tasks when a complete byte has been transmitted. The design of the serial I/O interface allows any number of MAB84X1 family devices and peripheral circuits with I^2C bus compatibility to be interconnected by the two-line serial bus. This is achieved by allocating a specific 7-bit address to each device and ensuring that a device reacts only to a message preceded by its own address or the 'general call' address.

Address recognition is performed by the interface hardware so that the microcontroller need only be interrupted when a valid address is received. This saves significant processing time and memory space compared to a conventional microcontroller with a software serial interface. When the address facility is not required, for instance in a system with only two microcontrollers, direct data transfer is possible. In multi-master systems, an automatically invoked arbitration procedure prevents two or more devices transmitting simultaneously.

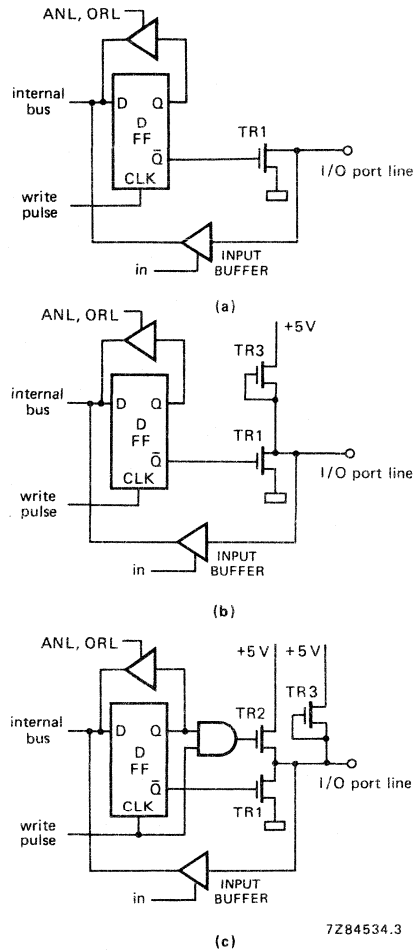


Fig. 7 Quasi-bidirectional I/O interface with (a) open drain output without pull-up transistor, (b) open drain output with pull-up transistor, (c) push-pull output with pull-up transistor.

Serial I/O interface

Figure 8 shows the serial I/O interface. The clock line of the serial bus has exclusive use of pin 3 (SCLK) while the data line shares pin 2 (serial data) with the I/O line P23 of port 2. When the serial I/O is enabled, P23 is disabled as a parallel port line (P23 and SCLK only open drain).

The microcontroller and interface communicate via the internal microcontroller bus and the Serial Interrupt Request line. Data and information controlling the operation of the interface are stored in four registers:

- data shift register S0,
- serial I/O interface status word S1,
- serial clock control word S2,
- address register.

FUNCTIONAL DESCRIPTION (continued)

Serial I/O interface (continued)

Data shift register S0

S0 is the shift register that converts serial data to parallel format and vice versa. A pending interrupt is generated only after a complete byte has been transmitted, or after a complete data byte, specific or general call address has been received. The most significant bit is transmitted first.

Status word S1

S1 provides information about the state of the interface and stores interface control information from the microcontroller. The four most significant bits are common to both read and write instructions, with a separate 4 read-only control bits and 4 write-only interface status bits.

MST and TRX

These bits determine the operating mode of the serial I/O interface (Table 2).

Table 1 Operating modes of the serial I/O interface.

MST	TRX	mode
0	0	slave receiver
1	0	master receiver
0	1	slave transmitter
1	1	master transmitter

BB: Bus Busy

This bit indicates the status of the bus.

PIN: Pending Interrupt Not

PIN = '0' indicates that there is an interrupt pending. This causes a Serial Interrupt Request when the serial interrupt mechanism is enabled.

ESO: Enable Serial Output

The ESO flag enables/disables the serial I/O interface: ESO = '1' enables
ESO = '0' disables

BC0, BC1 and BC2

These bits indicate the number of bits received or transmitted in a serial data stream.

Bits ES0, BC0, BC1 and BC2 can only be written via software.

AL: Arbitration Lost

The AL flag is set via the hardware when the serial I/O interface, as a master transmitter, loses the bus arbitration procedure.

ASS: Addressed As Slave

This flag is set via the hardware when the interface detects either its own address or the 'general call' address as the first byte of a transfer and if the interface has been programmed to operate in the address recognition mode.

AD0: Address Zero

This flag is set via the hardware after the general call address is detected when the interface is operating in the address recognition mode.

LRB: Last Received Bit

This contains either the last data bit received or, for a transmitting device in the acknowledge mode, the acknowledge from the receiving device.

Bits AL, AAS, AD0 and LRB can only be read via software.

Clock control register S2

Bits 0 to 4 of S2 are used to set the frequency of the serial clock signal. When a 4,43 MHz crystal is used, the frequency of the serial clock can be varied between 100 kHz and 720 Hz. An asymmetrical clock with a HIGH to LOW ratio of 3 to 1 is produced by setting bit 5. The asymmetrical clock allows a microcontroller more time per clock period for sampling the data line, making the timing of this action less critical. Bit 6 is used to activate the acknowledge mode of the serial I/O. S2 is a write-only register.

Address register

The address register contains the 7-bit address back-up latches and the bit (ALS) used to enable/disable the address recognition mode. Only when ES0 = 0 can the address register be written using the MOV S0,A and MOV S0,#data instructions.

Serial I/O interrupt logic

The interrupt logic is enabled by the EN SI instruction and disabled by DIS SI. When the interrupt logic is enabled, a pending interrupt results in a serial I/O interrupt to the controller, causing a jump to location 5 in the ROM. When the logic is disabled, the presence of an interrupt is still indicated by the PIN bit in register S1. Therefore, an interrupt can still be serviced but a vectored interrupt will not occur.

Interrupt system

External events and real-time on-chip peripherals require servicing by the CPU asynchronous to the execution of any particular section of code. To tie the asynchronous activities of these functions to normal program execution, three single-level nested interrupts are provided.

Each interrupt vectors to a separate location in the program memory for its service program. Each source can be individually enabled or disabled. When more than one interrupt occurs simultaneously, their priority will be: (1) external, (2) serial I/O and, (3) timer/event counter. An additional external interrupt can be created using the timer/event counter interrupt.

FUNCTIONAL DESCRIPTION (continued)

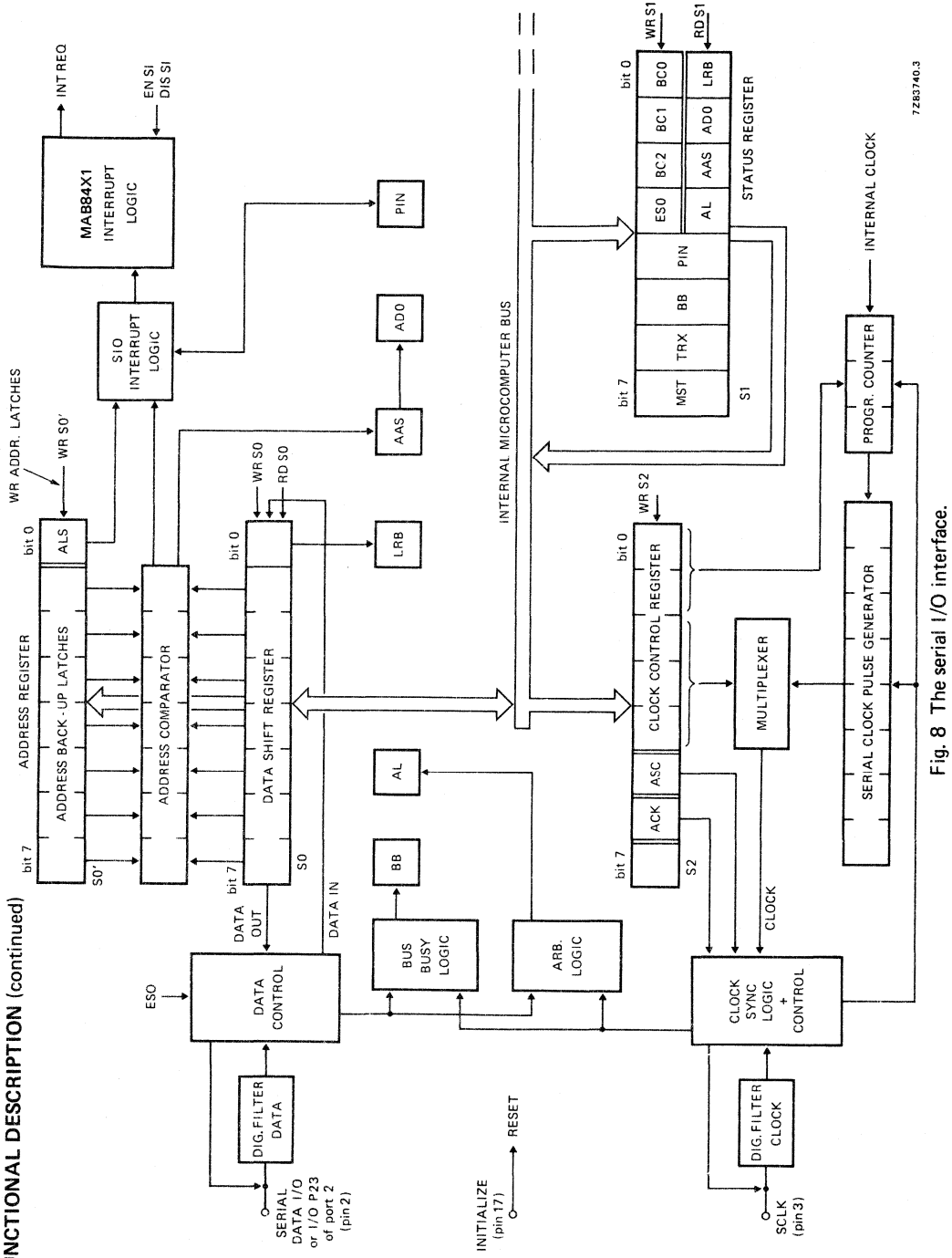


Fig. 8 The serial I/O interface.

Test input T1

The T1 input line can be used as:

- a test input for branch instructions,
- an input for zero voltage cross-over detection,
- an external input to the event counter.

An internal pull-up transistor is provided as a ROM mask option. This is useful when the input is from a switch or standard TTL output.

When T1 is used as a test input, the JT1 or JNT1 instructions test for a HIGH or a LOW respectively.

When used for zero-cross detection purposes, the T1 input must be coupled through a capacitor of typical value 1 μF and operation carried out using the T1 input without the pull-up transistor. The maximum input voltage amplitude is 3 V (peak-to-peak), with a maximum operational frequency of 1 kHz. The T1 input has an on-chip d.c. offset circuit which self-biases the input to its exact switching level of 1V. As a consequence a small change will cause a digital transition to occur. The switching level of the T1 input circuit is within the bias voltage of ± 135 mV. Upon each positive cycle on the pin, the event counter is incremented and an overflow will set the timer flag TF. Zero cross-over detection used in conjunction with the timer/event counter interrupt, is useful in thyristor control of power equipment. Figure 9 illustrates, (a) the input waveform, (b) the input diagram and (c) the on-chip self-stabilized bias.

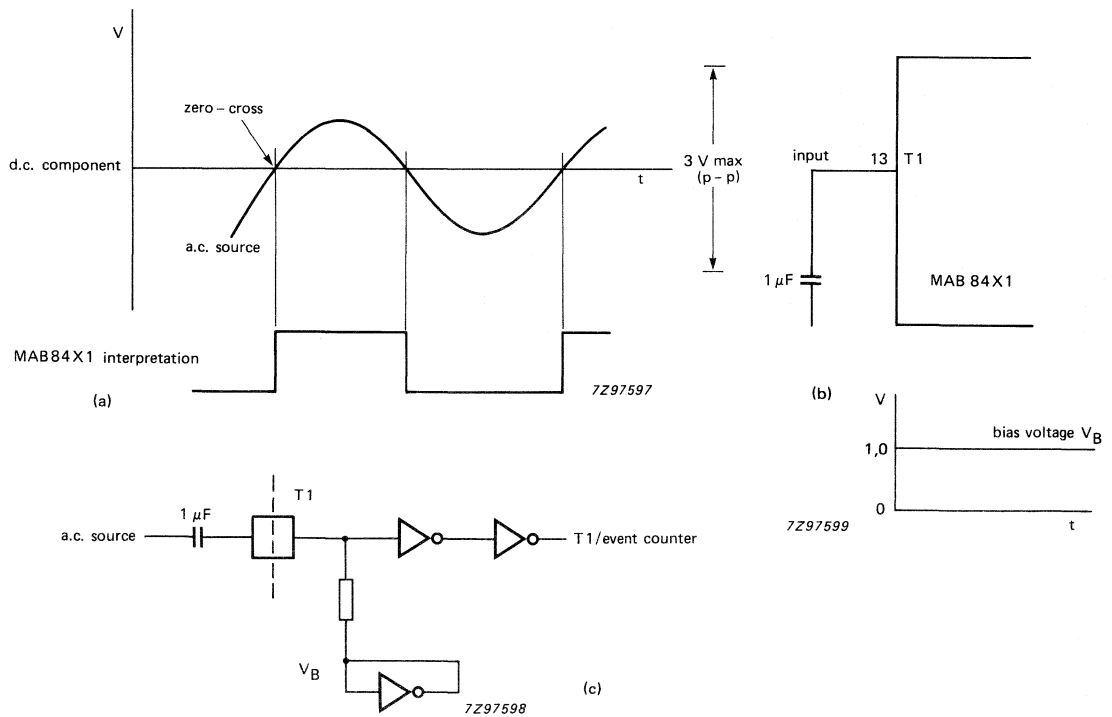


Fig. 9 Zero-cross detection circuitry.

The operation of T1 as an input to the timer/event counter is described under the heading Timer/event counter.

High current outputs

Ten pins are provided that can sink high currents:

– P23 (serial data), pin 2	5 mA at 0,45 V (open drain),
– SCLK, pin 3	5 mA at 0,45 V (open drain),
– P10 – P17▲	10 mA at 1 V

▲ P10 to P17 may be connected in parallel if their logic outputs are always the same.

Purchase of Philips' I²C components conveys a license under the Philips' I²C patent to use the components in the I²C-system provided the system conforms to the I²C specifications defined by Philips.

FUNCTIONAL DESCRIPTION (continued)

Timer/event counter

An 8-bit binary up-counter is provided. This can count external events, machine cycles divided by 32, or machine cycles directly. When used as a timer, the input to the counter is either the overflow or input of a 5-bit prescaler. When used as an event counter, LOW to HIGH transitions on T1 (pin 13) are counted. The maximum rate at which the counter may be incremented is once every machine cycle (200 kHz for a 5 μs machine cycle). Figure 10 illustrates the timer/event counter.

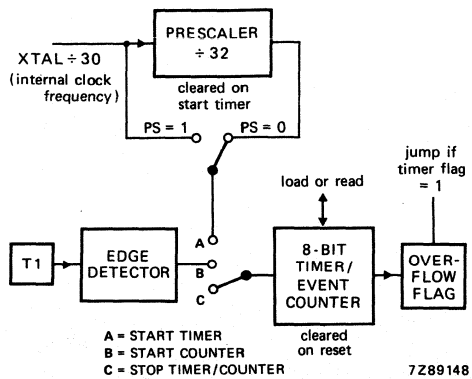


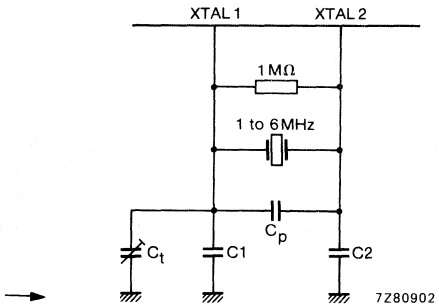
Fig. 10 The timer/event counter.

Differences between the MAB8021 and MAB8048 microcontrollers, and the MAB84X1 family.

	8021	8048	8401, 8411 8421, 8441, 8461
ROM capacity (bytes)	1K	1K	ROMless, 1K, 2K, 4K, 6K
RAM capacity (bytes)	64	64	64, 64, 128, 128
paralel I/O lines	8 + 8 + 4	8 + 8 + 8	8 + 8 + 4
single inputs	1	3	2
serial I/O	no	no	yes, 2-line multi-transmitter
timer	8 bit	8 bit	8 bit
prescaler	mod. 32	mod. 32	mod. 1 & mod. 32
machine cycle time (μs)	10	2,5	5
for clock (MHz)	3	6	6
instruction set	8021	8048	8048 with omissions; 5 new serial I/O instructions; 2 new register instructions; 2 new control instructions; 1 new cond. branch instruction
interrupts	none	2 external timer/ event counter	3 external serial I/O timer/event counter
no. of pins (DIL)	28	40	(PLCC-68) 28

OSCILLATOR CIRCUITRY

Clock frequency is determined by using the internal oscillator or by connecting an external clock to XTAL1. Where the internal oscillator is used, the frequency is set by a crystal between XTAL1 and XTAL2, or by a ceramic resonator or an inductor, each with two associated capacitors, between XTAL1 and XTAL2 (see Fig. 11a). A machine cycle consists of 10 states, each state being 3 oscillator periods. The common 6 MHz crystal gives a 5 μs machine cycle. The MAB84X1 family has dynamic logic, and therefore, for adequate refreshing the oscillator frequency must be at least 1 MHz.



1. Crystal – AT-cut
2. Ceramic resonator
   C1 = C2 = 27 pF
   C1 may be trimmed
   $C_p \leqslant 6{,}75$ pF (parasitic capacitance)

Fig. 11a Quartz crystal or ceramic resonator mode.

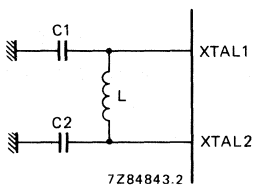


Fig. 11b LC pi-network.

LC oscillator timing

frequency	C1 = C2	L
3,0 MHz	33 pF	100 μH
4,0 MHz	33 pF	56 μH
4,4 MHz	33 pF	47 μH
5,0 MHz	33 pF	33 μH
6,0 MHz	33 pF	22 μH

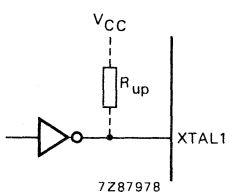


Fig. 11c External drive.

Drive XTAL1
Leave XTAL2 open

Driver may be high-speed CMOS or any TTL
t_r, $t_f < 10$ ns

PROGRAM STATUS WORD

The program status word (PSW) is an 8-bit word in the CPU which stores information about the current status of the microcontroller (Fig. 12). The PSW bits are:

bits 0, 1 and 2 — stack pointer bits (SP_0, SP_1, SP_2);

bit 3 — prescaler select (PS); 0 = divide-by-32; 1 = no prescaling;

bit 4 — working register bank select (RBS):
0 = register bank 0
1 = register bank 1;

bit 5 — not used (1);

bit 6 — auxiliary carry (AC):
half-carry bit is generated by an ADD instruction and used by the decimal adjust instruction DA A;

bit 7 — carry (CY):
the carry flag indicates that the previous operation has resulted in an overflow of the accumulator.

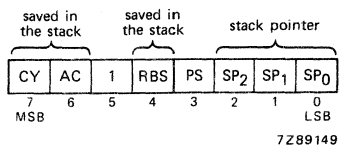


Fig. 12 Program status word.

All bits can be read using MOV A, PSW and bit 3 can be written with MOV PSW, A.

Bits 6 and 7 can be set and cleared by CPU operation. Bit 4 is changed by the SEL RB instruction, bit 3 by the MOV PSW,A instruction, and bits 0, 1 and 2 by the CALL, RET or RETR instructions and when an interrupt occurs. Bits 4, 6 and 7 are stored in the program counter stack during subroutine and interrupt calls. These bits are restored to the PSW with RETR (return and restore) instruction.

Note: The RET instruction has no restore feature and should not be used at the end of an interrupt because this would leave any further interrupts disabled.

The MAB84X1 family has arithmetic, logical and branching capabilities. The DA A, SWAP A, and XCHD instructions simplify BCD arithmetic and the handling of nibbles. The MOVP A,@A instruction permits efficient table look up from the current ROM page.

The conditional branch logic within the processor enables several conditions, internal and external to the processor, to be tested by the user's program. Table 2 lists the conditional branch instructions used to change the program execution sequence. The DJNZ instruction decrements a designated register and branches if the contents are not zero. This instruction makes the register an efficient program loop counter. The JMPP @A instruction allows multiway branches to destinations indirectly addressed by the contents of the accumulator.

Table 2 Conditional branches

TEST	JUMP CONDITION	JUMP INSTRUCTION
accumulator	0 or non-zero	JZ, JNZ
accumulator bit test	1	JB0 to JB7
carry flag	0 or 1	JNC, JC
timer overflow flag	1	JTF
test input $\overline{INT}$	0 or 1	JNT0, JT0
test input T1	0 or 1	JNT1, JT1
test flag 0	1	JF0
test flag 1	1	JF1
register	non-zero	DJNZ

RESET

A positive-going signal on the RESET input:

- sets the program counter to zero,
- selects location 0 of memory bank 0, and register bank 0,
- sets the stack pointer to zero ('000'B); pointing to RAM address 8,
- disable the interrupts (external, timer and serial I/O),
- stops the timer/event counter, then sets it to zero,
- sets the timer prescaler to divide-by-32,
- resets the timer flag,
- sets all ports to logic '1' (input mode),
- sets the serial I/O to slave receiver mode and disables serial I/O.

Automatic reset at power-up may be obtained by connecting the RESET pin to V_{CC} through a 1 μF capacitor C, together with a diode to V_{SS} (cathode to RESET pin). This arrangement is satisfactory, if both the voltage (V_{CC}) rise time and the oscillator start-up time do not exceed either 1 or 10 ms respectively.

The power-on reset circuit is shown in figure 13. At power-on the current drawn by RESET commences to charge the capacitor C. The difference between this increasing capacitor voltage and V_{CC} is known as V_{RESET}. The charging circuit is designed to hold V_{RESET} above the lower threshold of a Schmitt trigger arrangement long enough to effect a complete reset. The minimum time required; is the oscillator start-up time plus two machine cycles.

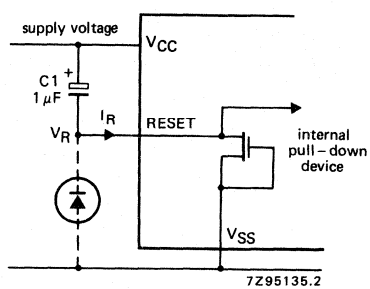


Fig. 13 Typical power-on reset circuitry.

INSTRUCTION SET

The instruction set consists of over 80 one and two byte instructions and is based on the MAB8048 instruction set. New instructions include those for serial I/O operation and memory bank selection. Program code efficiency is high because all ROM locations on a 256 byte page require only a single byte address.

Table 3 gives the instruction set of the MAB84XX family and Table 4 shows the instruction map. The following symbols and abbreviations are used.

Note: During development of software on a PMDS or similar system, it is important to ensure that no jump instruction (direct or indirect), outreaches the final address range of the device.

symbol	description
A	the accumulator
AC	the auxiliary carry flag
addr	program memory address (11-bits)
Bb	bit designation (b = 0–7)
BS	the bank switch
C	carry flag
CLK	clock signal
CNT	event counter
D	nibble designation (4-bits)
DBF	program memory bank flip-flop
data	number or expression (8-bits)
F0, F1	flags 0 and 1
I	interrupt
$\overline{\text{INT}}$	external interrupt
P	'in-page' operation designation
Pp	port designation (p = 1, 2 or 4–7)
PSW	program status word
Rr	register designation (r = 0, 1 or 0–7)
SP	stack pointer
T	timer
TF	timer flag
T0, T1	test 0 and 1 inputs
#	immediate data prefix
@	indirect address prefix
$	current value of program counter
←	is replaced by
↔	is exchanged with

Table 3 MAB84XX family instruction set

	mnemonic	opcode (hex.)	bytes/ cycles	description	function		notes
ACCUMULATOR	ADD A, Rr	6*	1/1	Add register contents to A	(A)←(A) + (Rr)	r = 0–7	1
	ADD A, @Rr	60 61	1/1	Add RAM data, addressed by Rr, to A	(A)←(A) + ((R0)) (A)←(A) + ((R1))		1
	ADD A, #data	03 data	2/2	Add immediate data to A	(A)←(A) + data		1
	ADDC A, Rr	7*	1/1	Add carry and register contents to A	(A)←(A) + (Rr) + (C)	r = 0–7	1
	ADDC A, @Rr	70 71	1/1	Add carry and RAM data, addressed by Rr, to A	(A)←(A) + ((R0)) + (C) (A)←(A) + ((R1)) + (C)		1
	ADDC A, #data	13 data	2/2	Add carry and immediate data to A	(A)←(A) + data + (C)		1
	ANL A, Rr	5*	1/1	'AND' Rr with A	(A)←(A) AND (Rr)	r = 0–7	
	ANL A, @Rr	50 51	1/1	'AND' RAM data, addressed by Rr, with A	(A)←(A) AND ((R0)) (A)←(A) AND ((R1))		
	ANL A, #data	53 data	2/2	'AND' immediate data with A	(A)←(A) AND data		
	ORL A, Rr	4*	1/1	'OR' Rr with A	(A)←(A) OR (Rr)	r = 0–7	
	ORL A, @Rr	40 41	1/1	'OR' RAM data, addressed by Rr, with A	(A)←(A) OR ((R0)) (A)←(A) OR ((R1))		
	ORL A, #data	43 data	2/2	'OR' immediate data with A	(A)←(A) OR data		
	XRL A, Rr	D*	1/1	'XOR' Rr with A	(A)←(A) XOR (Rr)	r = 0–7	
	XRL A, @Rr	D0 D1	1/1	'XOR' RAM, addressed by Rr, with A	(A)←(A) XOR ((R0)) (A)←(A) XOR ((R1))		
	XRL A, #data	D3 data	2/2	'XOR' immediate data with A	(A)←(A) XOR data		
	INC A	17	1/1	increment A by 1	(A)←(A) + 1		
	DEC A	07	1/1	decrement A by 1	(A)←(A) – 1		
	CLR A	27	1/1	clear A to zero	(A)←0		
	CPL A	37	1/1	one's complement A	(A)←NOT(A)		
	RL A	E7	1/1	rotate A left	$(A_{n+1})\leftarrow(A_n)$ $(A_0)\leftarrow(A_7)$	n = 0–6	

	mnemonic	opcode (hex.)	bytes/ cycles	description	function		notes
ACCUMULATOR (cont.)	RLC A	F7	1/1	rotate A left through carry	$(A_{n+1})\leftarrow A_n$ $(A_0)\leftarrow(C)$, $(C)\leftarrow(A_7)$	n = 0–6	2
	RR A	77	1/1	rotate A right	$(A_n)\leftarrow(A_{n+1})$ $(A_7)\leftarrow(A_0)$	n = 0–6	
	RRC A	67	1/1	rotate A right through carry	$(A_n)\leftarrow(A_{n+1})$ $(A_7)\leftarrow(C)$, $(C)\leftarrow(A_0)$	n = 0–6	2
	DA A	57	1/1	decimal adjust A			2
	SWAP A	47	1/1	swap nibbles of A	$(A_{4-7})\leftrightarrow(A_{0-3})$		
DATA MOVES	MOV A, Rr	F*	1/1	move register contents to A	(A)←(Rr)	r = 0–7	
	MOV A, @Rr	F0 F1	1/1	move RAM data, addressed by Rr, to A	(A)←((R0)) (A)←((R1))		
	MOV A, #data	23 data	2/2	move immediate data to A	(A)←data		
	MOV Rr, A	A*	1/1	move accumulator contents to register	(Rr)←(A)	r = 0–7	
	MOV @Rr, A	A0 A1	1/1	move accumulator contents to RAM location addressed by Rr	((R0))←(A) ((R1))←(A)		
	MOV Rr, #data	B* data	2/2	move immediate data to Rr	(Rr)←data		
	MOV @Rr, #data	B0 data B1 data	2/2	move immediate data to RAM location addressed by Rr	((R0))←data ((R1))←data		
	XCH A, Rr	2*	1/1	exchange accumulator contents with Rr	(A)↔(Rr)	r = 0–7	
	XCH A, @Rr	20 21	1/1	exchange accumulator contents with RAM data addressed by Rr	(A)↔((R0)) (A)↔((R1))		
	XCHD A, @Rr	30 31	1/1	exchange lower nibbles of A and RAM data addressed by Rr	$(A_{0-3})\leftrightarrow((R0_{0-3}))$ $(A_{0-3})\leftrightarrow((R1_{0-3}))$		
	MOV A, PSW	C7	1/1	move PSW contents to accumulator	(A)←(PSW)		
	MOV PSW, A	D7	1/1	move accumulator bit 3 to PSW_3	$(PSW_3)\leftarrow(A_3)$		3
	MOVP A, @A	A3	1/2	move indirectly addressed data in current page to A	$(PC_{0-7})\leftarrow(A)$, $(A)\leftarrow((PC))$		

FLAGS	CLR C	97	1/1	clear carry bit	(C)←0		2
	CPL C	A7	1/1	complement carry bit	(C)←NOT(C)		2
REGISTER	INC Rr	1*	1/1	increment register by 1	(Rr)←(Rr) + 1	r = 0–7	
	INC @Rr	10 11	1/1	increment RAM data, addressed by Rr, by 1	((R0))←((R0)) + 1 ((R1))←((R1)) + 1		
	DEC Rr	C*	1/1	decrement register by 1	(Rr)←(Rr) −1	r = 0–7	
	DEC @Rr	C0 C1	1/1	decrement RAM data, addressed by Rr, by 1	((R0))←((R0)) −1 ((R1))←((R1)) −1		
BRANCH	JMP addr	• 4 address	2/2	unconditional jump within a 2K bank	(PC_{8-10})←$addr_{8-10}$ (PC_{0-7})←$addr_{0-7}$ (PC_{11-12})←MBFF 0–1		
	JMPP @A	B3	1/2	indirect jump within a page	(PC_{0-7})←((A))		
	DJNZ Rr, addr	E* address	2/2	decrement Rr by 1 and jump if not zero to addr	(Rr)←(Rr) −1 if (Rr) not zero (PC_{0-7})←addr	r = 0–7	
	DJNZ @Rr, addr	E0 address	2/2	decrement RAM data, addressed by Rr, by 1 and jump if not zero to addr	((R0))←((R0)) −1 if ((R0)) not zero (PC_{0-7})←addr		
		E1 address			((R1))←((R1)) −1 if ((R1)) not zero (PC_{0-7})←addr		
	JBb addr	▲ 2 address	2/2	jump to addr if Acc. bit b = 1	if b = 1: (PC_{0-7})←addr	b = 0–7	
	JC addr	F6 address	2/2	jump to addr if C = 1	if C = 1: (PC_{0-7})←addr		
	JNC addr	E6 address	2/2	jump to addr if C = 0	if C = 0: (PC_{0-7})←addr		
	JZ addr	C6 address	2/2	jump to addr if A = 0	if A = 0: (PC_{0-7})←addr		
	JNZ addr	96 address	2/2	jump to addr if A is NOT zero	if A≠0: (PC_{0-7})←addr		
	JT0 addr	36 address	2/2	jump to addr if T0 = 1	if T0 = 1: (PC_{0-7})←addr		
	JNT0 addr	26 address	2/2	jump to addr if T0 = 0	if T0 = 0: (PC_{0-7})←addr		
	JT1 addr	56 address	2/2	jump to addr if T1 = 1	if T1 = 1: (PC_{0-7})←addr		
	JNT1 addr	46 address	2/2	jump to addr if T1 = 0	if T1 = 0: (PC_{0-7})←addr		
	JTF addr	16 address	2/2	jump to addr if Timer Flag = 1	if TF = 1: (PC_{0-7})←addr		4
	JNTF addr	06 address	2/2	jump to addr if Timer Flag = 0	if TF = 0: (PC_{0-7})←addr		

	mnemonic	opcode (hex.)	bytes/ cycles	description	function	notes
TIMER/EVENT COUNTER	MOV A, T	42	1/1	move timer/event counter contents to accumulator	(A)←(T)	
	MOV T, A	62	1/1	move accumulator contents to timer/event counter	(T)←(A)	
	STRT CNT	45	1/1	start event counter		
	STRT T	55	1/1	start timer		
	STOP TCNT	65	1/1	stop timer/event counter		
	EN TCNTI	25	1/1	enable timer/event counter interrupt		
	DIS TCNTI	35	1/1	disable timer/event counter interrupt		
CONTROL	EN I	05	1/1	enable external interrupt		
	DIS I	15	1/1	disable external interrupt		
	SEL RB0	C5	1/1	select register bank 0	(RBS)←0	5
	SEL RB1	D5	1/1	select register bank 1	(RBS)←1	5
	SEL MB0	E5	1/1	select program memory bank 0	(MBFF0)←0, (MBFF1)←0	
	SEL MB1	F5	1/1	select program memory bank 1	(MBFF0)←1, (MBFF1)←0	
	SEL MB2	A5	1/1	select program memory bank 2	(MBFF0)←0, (MBFF1)←1	
	SEL MB3	B5	1/1	select program memory bank 3	(MBFF0)←1, (MBFF1)←1	
SUBROUTINE	CALL addr	▲ 4 address	2/2	jump to subroutine	((SP))←(PC), ($PSW_{4, 6, 7}$) (SP)←(SP) + 1 (PC_{9-10})←$addr_{8-10}$ (PC_{0-7})←$addr_{0-7}$ (PC_{11-12})←MBFF 0–1	6
	RET	83	1/2	return from subroutine	(SP)←(SP) −1 (PC)←((SP))	6
	RETR	93	1/2	return from interrupt and restore bits 4, 6, 7 of PSW	(SP)←(SP) −1 ($PSW_{4, 6, 7}$) + (PC)←((SP))	6

PARALLEL INPUT/OUTPUT	IN A, Pp	08 09 0A	1/2	input port p data to accumulator	(A)←(P0) (A)←(P1) (A)←(P2)	7
	OUTL Pp, A	38 39 3A	1/2	output accumulator data to port p	(P0)←(A) (P1)←(A) (P2)←(A)	
	ANL Pp, #data	98 data 99 data 9A data	2/2	AND port p data with immediate data	(P0)←(P0) AND data (P1)←(P1) AND data (P2)←(P2) AND data	
	ORL Pp, #data	88 data 89 data 8A data	2/2	OR port p data with immediate data	(P0)←(P0) OR data (P1)←(P1) OR data (P2)←(P2) OR data	
	OUTL PO,A	90	1/2	Output accumulator data to port ϕ	(P0)←(A)	9
SERIAL INPUT/OUTPUT	MOV A, S_n	0C 0D	1/2	move serial I/O register contents to accumulator	(A)←(S0) (A)←(S1)	8
	MOV S_n, A	3C 3D 3E	1/2	move accumulator contents to serial I/O register	(S0)←(A) (S1)←(A) (S2)←(A)	
	MOV S_n, #data	9C 9D 9E	2/2	move immeidate date to serial I/O register	(S0)←data (S1)←data (S2)←data	
	EN SI	85	1/1	enable serial I/O interrupt		
	DIS SI	95	1/1	disable serial I/O interrupt		
	NOP	00	1/1	no operation		

Notes to Table 3.

1. PSW CY, AC affected
2. PSW CY affected
3. PSW PS affected
4. Execution of JTF and JNTF instructions resets the Timer Flag (TF).
5. PSW RBS affected
6. PSW SP_0, SP_1, SP_2 affected
7. (A) = 1111 P23, P22, P21, P20.
8. (S1) has a different meaning for read and write operation, see serial I/O interface.
9. Only for software-transfer from the MAB8021.

* : 8, 9, A, B, C, D, E, F
● : 0, 2, 4, 6, 8, A, C, E
▲ : 1, 3, 5, 7, 9, B, D, F

Table 4 MAB84X1 family instruction set

first hexadecimal character of opcode ↓

second hexadecimal character of opcode →

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NOP			ADD A,#data	JMP page 0	EN I	JNTF addr	DEC A	IN A,Pp				MOV A,Sn			
									0	1	2		0	1		
1	INC @Rr		JBO addr	ADDC A,#data	CALL page 0	DIS I	JTF addr	INC A	INC Rr							
	0	1							0	1	2	3	4	5	6	7
2	XCH A,@Rr			MOV A,#data	JMP page 1	EN TCNTI	JNTO addr	CLR A	XCH A,Rr							
	0	1							0	1	2	3	4	5	6	7
3	XCHD A,@Rr		JB1 addr		CALL page 1	DIS TCNTI	JTO addr	CPL A	OUTL Pp,A				MOV Sn,A			
	0	1							0	1	2		0	1	2	
4	ORL A,@Rr		MOV A,T	ORL A,#data	JMP page 2	STRT CNT	JNT1 addr	SWAP A	ORL A,Rr							
	0	1							0	1	2	3	4	5	6	7
5	ANL A,@Rr		JB2 addr	ANL A,#data	CALL page 2	STRT T	JT1 addr	DA A	ANL A,Rr							
	0	1							0	1	2	3	4	5	6	7
6	ADD A,@Rr		MOV T,A		JMP page 3	STOP TCNT		RRC A	ADD A,Rr							
	0	1							0	1	2	3	4	5	6	7
7	ADDC A,@Rr		JB3 addr		CALL page 3			RR A	ADDC A,Rr							
	0	1							0	1	2	3	4	5	6	7
8				RET	JMP page 4	EN SI			ORL Pp,#data							
									0	1	2					
9	OUTL PO,A		JB4 addr	RETR	CALL page 4	DIS SI	JNZ addr	CLR C	ANL Pp,#data				MOV Sn,#data			
									0	1	2		0	1	2	
A	MOV @Rr,A			MOVP A,@A	JMP page 5	SEL MB2		CPL C	MOV Rr,A							
	0	1							0	1	2	3	4	5	6	7
B	MOV @Rr,#data		JB5 addr	JMPP @A	CALL page 5	SEL MB3			MOV Rr,#data							
	0	1							0	1	2	3	4	5	6	7
C	DEC @Rr				JMP page 6	SEL RB0	JZ addr	MOV A,PSW	DEC Rr							
	0	1							0	1	2	3	4	5	6	7
D	XRL A,@Rr		JB6 addr	XRL A,#data	CALL page 6	SEL RB1		MOV PSW,A	XRL A,Rr							
	0	1							0	1	2	3	4	5	6	7
E	DJNZ @Rr,addr				JMP page 7	SEL MB0	JNC addr	RL A	DJNZ Rr, addr							
	0	1							0	1	2	3	4	5	6	7
F	MOV A,@Rr		JB7 addr		CALL page 7	SEL MB1	JC addr	RLC A	MOV A,Rr							
	0	1							0	1	2	3	4	5	6	7

Table 5 shows the additional MAB84X1 family instructions (including the five for serial I/O operation) that are not part of the MAB8048 instruction set.

Table 5 MAB84X1 family instructions not in the MAB8048 instruction set

serial I/O	register	control	conditional branch

MOV A, S_n
MOV S_n, A
MOV S_n, #data
EN SI
DIS SI
	DEC @Rr
	DJNZ @Rr, addr
		SEL MB2
		SEL MB3
			JNTF addr

Table 6 shows the MAB8048 instructions omitted from the MAB84X1 family instruction set.

Table 6 MAB8048 instructions not in the MAB84X1 family instruction set

data moves	flags	branch	control

MOVX A, @R
MOVX @R, A
MOVP3 A, @A
MOVD A, P
MOVD P, A
ANLD P, A
ORLD P, A
	CLR F0
	CPL F0
	CLR F1
	CPL F1
		* JNI addr
		JF0 addr
		JF1 addr
		
		* replaced by
		JT0 JNT0.
			ENT0 CLK

ABSOLUTE MAXIMUM RATINGS

Limiting values in accordance with the Absolute Maximum System (IEC 134)

Stress above those listed under 'Absolute maximum ratings' may cause permanent damage to the device. This is a stress rating only and functional operation of the device, at these, or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

Input voltage on any pin with respect to ground (V_{SS})		V_I	−0,5 to + 7	V
Total power dissipation		P_{tot}	1	W
Input/output for all pins except port 1		I_I, I_O	max. 10	mA
Input/output current for port 1		I_I, I_O	max. 20	mA
Storage temperature		T_{stg}	−65 to + 150	°C
Operating temperature	standard	T_{amb}	0 to + 70	°C
	extended	T_{amb}	−40 to + 85	°C
	automotive	T_{amb}	−40 to + 110	°C

D.C. CHARACTERISTICS

V_{CC} = 5 V (± 10%); V_{SS} = 0 V; all voltages with respect to V_{SS} unless otherwise specified

parameter	symbol	min.	max.	unit	conditions
Supply current					
MAB	I_{CC}	–	85	mA	0 to + 70 °C
MAF	I_{CC}	–	100	mA	–40 to + 85 °C
MAF84A	I_{CC}	–	100	mA	–40 to + 110 °C
Inputs					
Input voltage LOW (except P23 and SCLK)	V_{IL}	–0,5	0,8	V	
Input voltage LOW (P23 and SCLK)	V_{IL1}	–0,5	1,5	V	
Input voltage HIGH (all inputs except XTAL1, P23 and SCLK)	V_{IH}	2	V_{CC} +0,5	V	
Input voltage HIGH (XTAL1, P23 and SCLK)	V_{IH1}	3,0	V_{CC} +0,5	V	
Outputs					
Output voltage LOW (P00–P07)	V_{OL}	–	0,45	V	I_{OL} = 1,6 mA
Output voltage LOW (P10–P17 for 8401/11/21/41/61)	V_{OL12}	–	1,0	V	I_{OL12} = 10 mA
Output voltage LOW (P20–P22)	V_{OL2}	–	0,45	V	I_{OL2} = 1,6 mA
Output voltage LOW (P23, SCLK)	V_{OL3}	–	0,45	V	I_{OL3} = 5 mA
Output voltage LOW (non-standard pins of bond-out versions)	V_{OL4}	–	0,45	V	I_{OL4} = 0,4 mA
Output voltage HIGH (all outputs unless (open drain)	V_{OH}	2,4	–	V	I_{OH} = –50 μA
Output leakage current	± I_{OL}	–	10	μA	$V_{SS} < V_I < V_{CC}$

A.C. CHARACTERISTICS (all versions except bond-out)

V_{CC} = 5 V ± 10%; V_{SS} = 0 V.

parameter	symbol		min.	max.	unit
Frequency	f_{XTAL}	MAB/MAF84X1	1	6	MHz
		MAF84AX1	1	5	MHz
Cycle time	t_{CY}	MAB/MAF84X1	5	30	μs
		MAF84AX1	6	30	μs

A.C. CHARACTERISTICS (bond-out versions)

V_{CC} = 5 V ± 10%; V_{SS} = 0 V.

parameter	symbol	min.	max.	unit
f_{CL} = 6 MHz				
Control pulse duration $\overline{PSEN}$ (9CP)	t_{CC}	1,5	9	μs
Address to $\overline{PSEN}$ L set-up (1CP)	t_{AS}	167	–	ns
Data to $\overline{PSEN}$ H set-up (1CP + 120 ns)	t_{DS}	600	–	ns
Data hold time	t_{DR}	0	–	ns
Address to data-in (10CP–t_{DS})	t_{AD}	–	1,07	μs
Time from $\overline{PSEN}$ L to C1 (3CP)	t_{PC}	500	–	ns
Time from $\overline{INTA}$ L to $\overline{PSEN}$ (3CP)	t_{IP0}	500	–	ns
Time from $\overline{INTA}$ H to $\overline{PSEN}$ (6CP)	t_{IP1}	1	–	μs
$\overline{HALT}$ set-up to $\overline{PSEN}$ (15CP)	t_{HS}	2,5	–	μs
$\overline{HALT}$ hold time from $\overline{PSEN}$ (3CP)	t_{HH}	500	–	ns

Note: CP = clock pulse.

T1 ZERO-CROSS CHARACTERISTICS

T_{amb} = 0 to + 70 °C; V_{CC} = 5 V ± 10%; V_{SS} = 0 V; C_L = 80 pF

parameter	symbol	min.	max.	unit	conditions
Zero-cross detection input (T1) peak-to-peak	$V_{ZX(p-p)}$	1	3	V	a.c. coupled, C = 1,0 μF
Zero-cross accuracy	A_{ZX}	–	± 135	mV	50 Hz sine wave
Zero-cross detection input frequency (T1)	F_{ZX}	0,05	1	kHz	

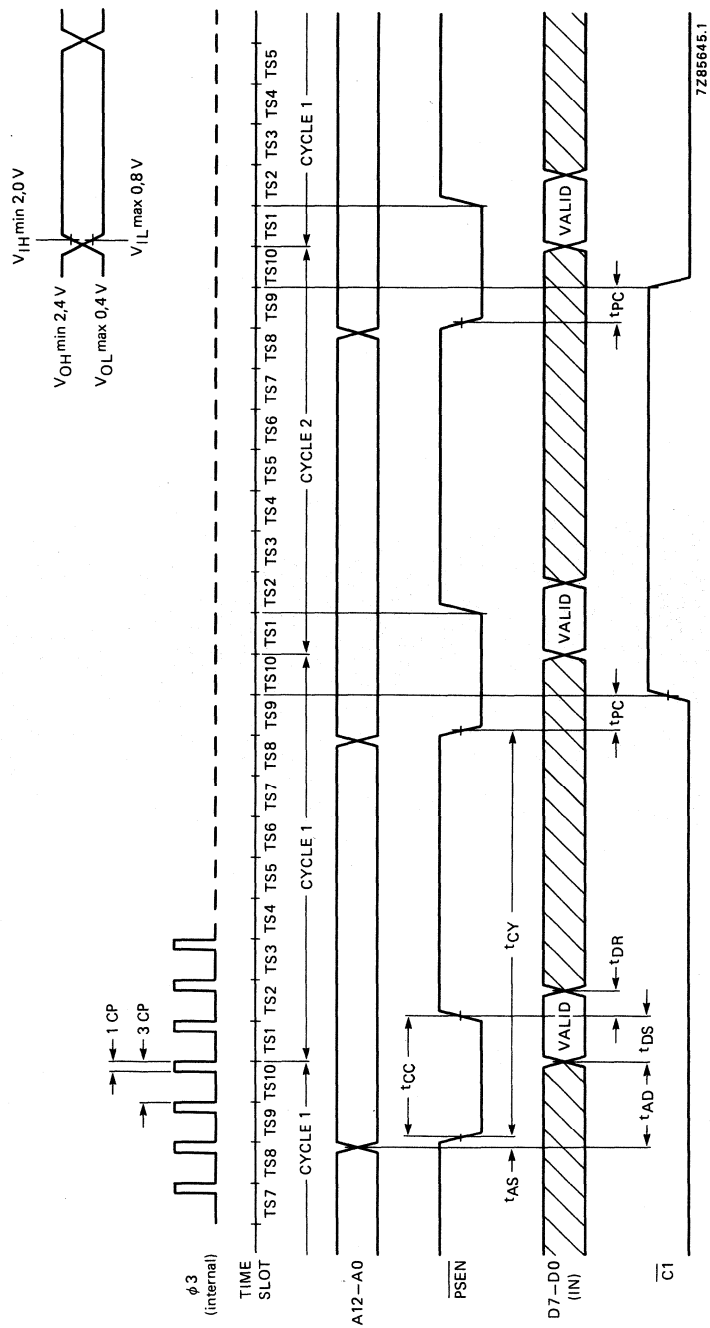


Fig. 14 Memory access timing MAB8401B/WP and I/O voltage parameters.

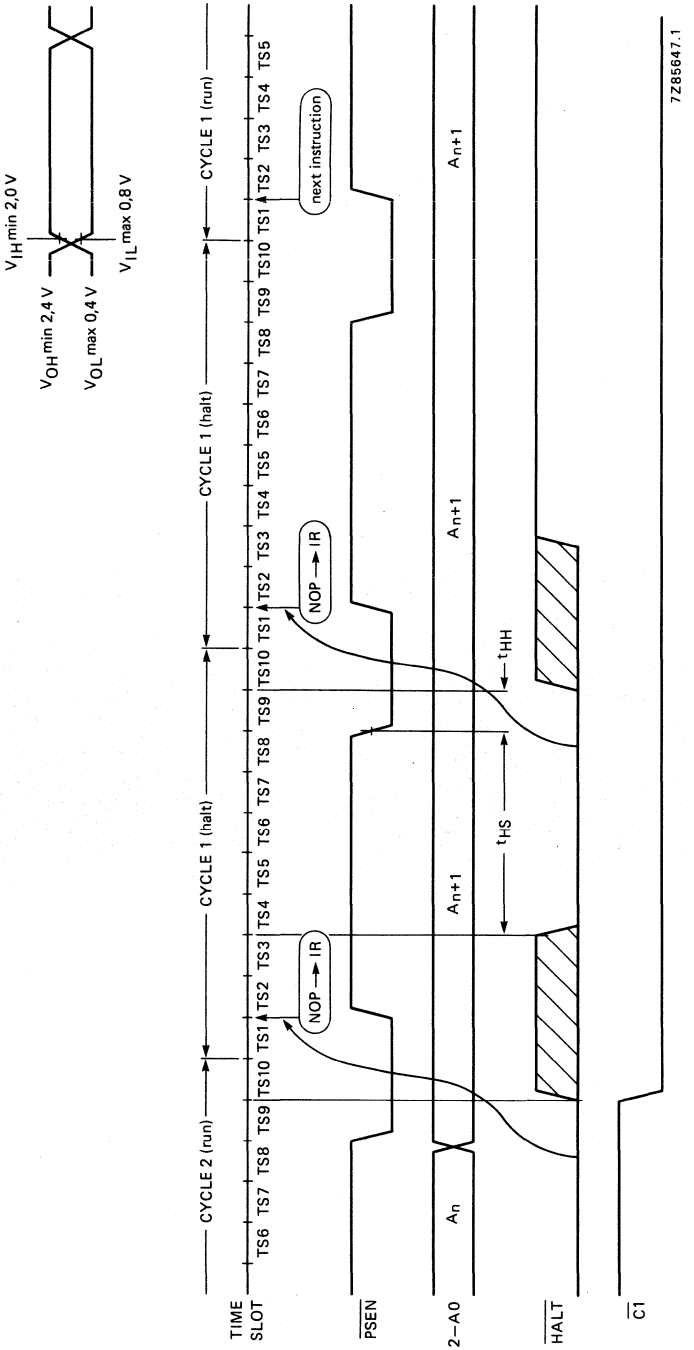


Fig. 15 $\overline{INTA}$ and $\overline{HALT}$ timing MAB8401 and I/O voltage parameters.

SINGLE-CHIP 8-BIT MICROCONTROLLER

DESCRIPTION

The MAB8422/8442 is a high-performance microcontroller incorporating dedicated hardware, memory capacity and I/O lines. This dedication means a microcontroller can be economically installed in high-volume products where its main function is control.

The MAB8422/8442 is a 20 pin, single-chip 8-bit microcontroller that has been developed from the 28 pin MAB8421/8441 microcontrollers. The versions are:

- MAB8422 - 2K x 8 ROM/64 bytes RAM
- MAB8442 - 4K x 8 ROM/128 bytes RAM

Each version has 15 I/O port lines comprising one 8-bit parallel port (P0), one 2-bit parallel port (P10 and P11 that are shared with the serial I/O lines SDA and SCL), one 3-bit parallel port (P20 - P22) and two input lines ($\overline{INT}$/T0 and T1).

The serial I/O interface is I²C compatible and therefore the MAB8422/8442 can operate as a slave or a master in single and multi-master systems. Conversion from parallel to serial data when transmitting, and vice versa when receiving, is done mainly in software. There is a minimum of hardware for the serial I/O implemented. This hardware is controlled by the status of the SDA and SCL lines and can be read or written under software control. Standard software for I²C-bus control is available upon request. For detailed information see the user manual 'Single-chip 8-bit microcontrollers'.

Features

- 8-bit: CPU, ROM, RAM and I/O
- 20 pin package
- MAB8422: 2K x 8 ROM/64 bytes RAM
- MAB8442: 4K x 8 ROM/128 bytes RAM
- 13 quasi-bidirectional I/O port lines
- Two testable inputs T1 and $\overline{INT}$/T0
- High current output on P0 (I_{OL} = 10 mA at V_{OL} = 1 V)
- One interrupt line combined with the testable input line $\overline{INT}$/T0
- Single-level interrupts: external, timer/event counter, serial I/O
- I²C-compatible serial I/O that can be used in single or multi-master systems (serial I/O data and clock via P10 and P11 port lines, respectively)
- 8-bit programmable timer/event counter
- Internal oscillator, generated with inductor, crystal, ceramic resonator or external source
- Over 80 instructions (based on MAB8048)
- All instructions 1 or 2 cycles, cycle time dependent on oscillator frequency
- Single power supply
- Operating temperature ranges: 0 to +70 °C (MAB84X2)
  −40 to +85 °C (MAF84X2)
  −40 to +110 °C (MAF84AX2)

PACKAGE OUTLINES

MAB/MAF84X2, MAF84AX2: 20-lead DIL; plastic (SOT-146)

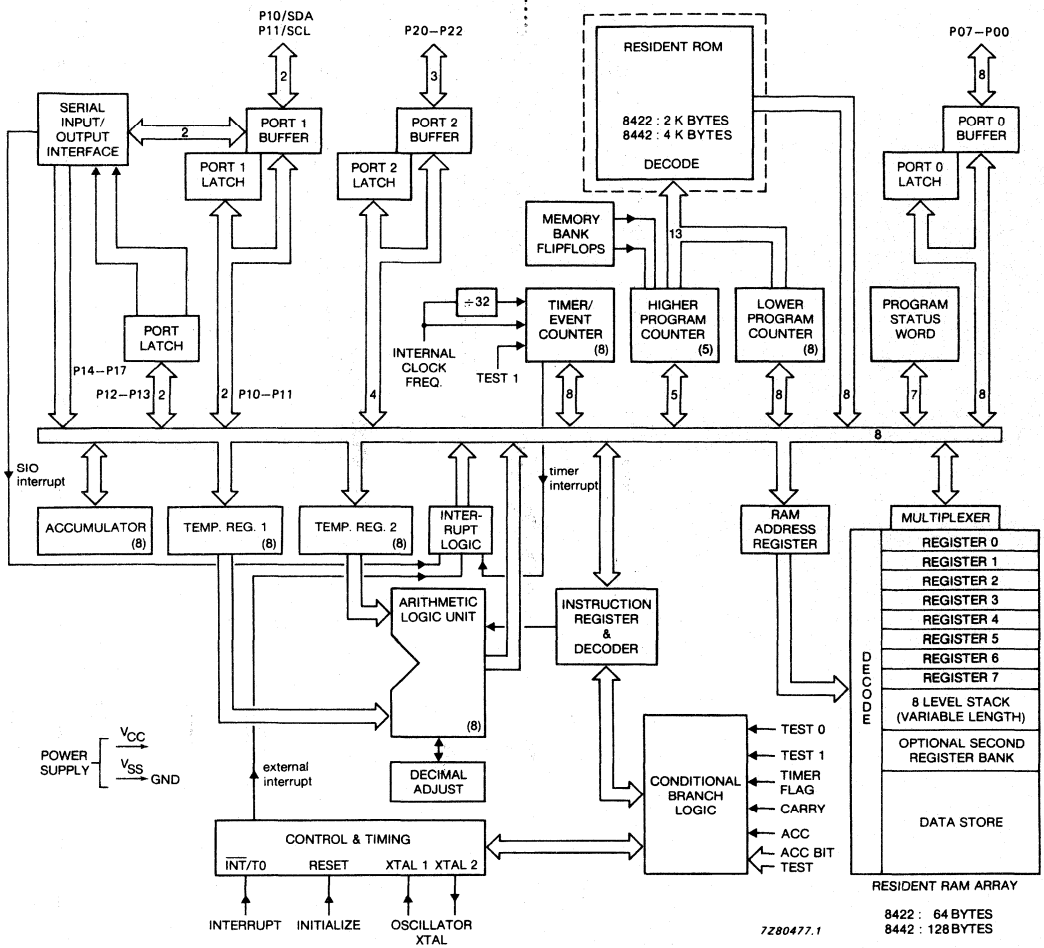


Fig. 1 Block diagram of the MAB8422/8442.

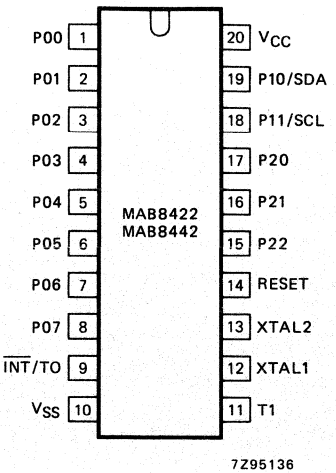


Fig. 2 Pinning diagram.

DEVELOPMENT DATA

PINNING

Designation	Pin number	Function
P00-P07	1-8	8-bit quasi-bidirectional I/O port (Port 0 high current output).
$\overline{INT}$/T0	9	External interrupt input (sensitive to a negative going edge) and/or input, testable using the JT0 or JNT0 instructions.
V_{SS}	10	Ground.
T1	11	Input pin, testable using the JT1 or JNT1 instructions. It can be designated as event counter input using the STRT CNT instruction. It can also be used to detect zero cross-over of slowly moving a.c. inputs.
XTAL1	12	Connection to timing component that determines the frequency of the internal oscillator. It is also the input for an external clock source.
XTAL2	13	Connection to the other side of the timing component.
RESET	14	Input to initialize the processor (active HIGH).
P20-P22	17-15	Quasi-bidirectional port.
P11/SCL	18	Quasi-bidirectional port in parallel port mode. Serial clock in serial I/O mode.
P10/SDA	19	Quasi-bidirectional port in parallel port mode. Serial data I/O in serial I/O mode.
V_{CC}	20	Power supply.

FUNCTIONAL DESCRIPTION

Program and data memory

The non-volatile program memory (ROM), as shown in Fig. 3, is arranged in two banks of 2K bytes, that are selected by SEL MB instructions, and each bank is further divided into 256-byte pages. Only the unconditional jump instructions (JMP and CALL) can be used to cross page boundaries. Memory bank boundaries can also be crossed using these instructions provided that the appropriate memory bank has been selected.

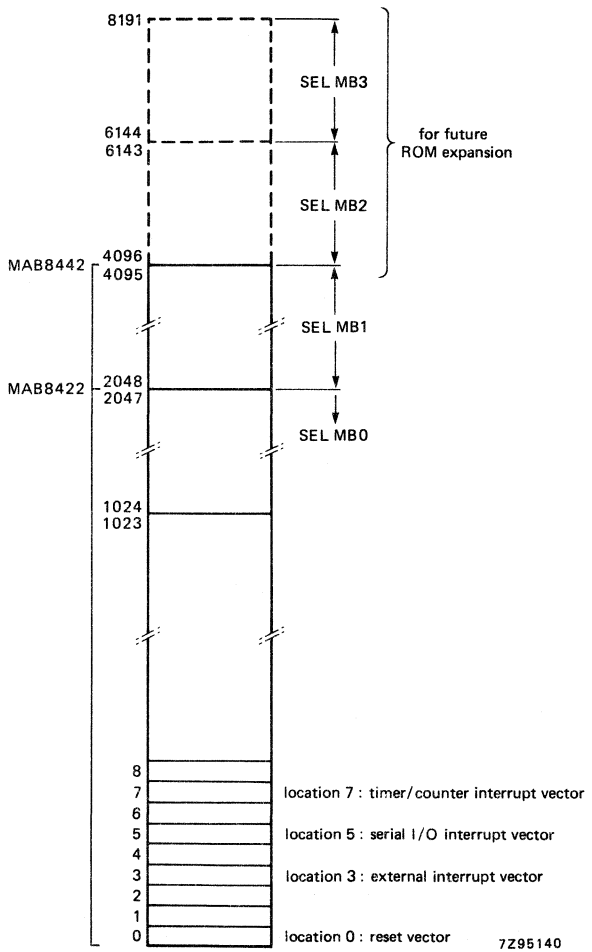


Fig. 3 Program memory map.

In the volatile data memory (RAM), all locations are indirectly addressable using RAM pointer registers and up to 16 designated locations can be addressed directly. The memory also includes an 8-level program counter stack addressed by a 3-bit stack pointer and two register banks, each with 8 registers. The data memory is shown in Fig. 4.

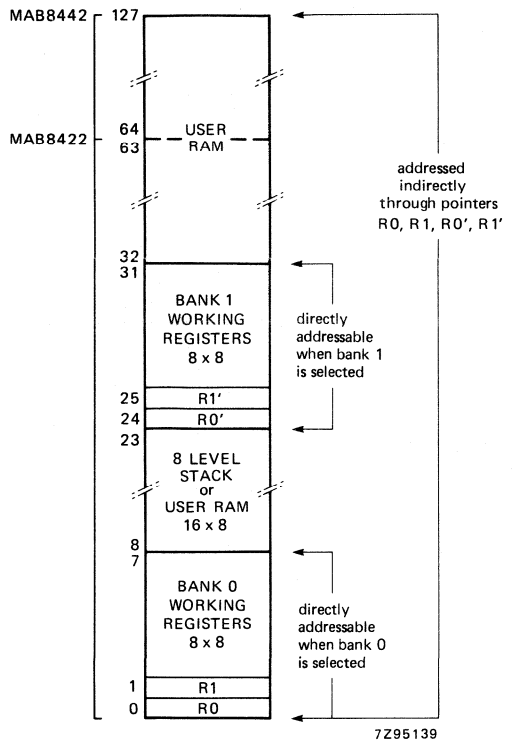


Fig. 4 Data memory map.

Instruction set

The instruction set consists of over 80 one and two byte instructions. It is identical to the MAB84X1 instruction set except that the instructions MOV Sn, A, MOV A, Sn and MOV Sn, #data are not used. Program code efficiency is high because all RAM locations on a 256 byte page require only a single byte address.

On-chip peripheral functions

In addition to the CPU and memories, an interrupt system, I/O facilities, and an 8-bit timer/event counter are integrated on-chip to assist the CPU in repetitious, complicated or time-critical tasks. The I/O facilities include the I/O pins, parallel ports and a serial I/O port sharing the two pins of the parallel port P1.

FUNCTIONAL DESCRIPTION (continued)

I/O facilities (see Fig. 5)

The MAB8422/8442 has 13 I/O lines and 2 testable inputs arranged as:

- An 8 line parallel port P00-P07, high current outputs with three optional output configurations:
  - A push-pull output with pull-up (Fig. 5 (a))
  - Open drain with pull-up (Fig. 5 (b))
  - Open drain without pull-up (Fig. 5 (c))
- A 2 line parallel/serial port P10/SDA and P11/SCL, open-drain without pull-up, as output configuration. Schmitt-trigger input. After RESET, P10/SDA and P11/SCL will be in the parallel port mode. To stay in this mode the internal port latches, P14 and P13, must be kept in the logic '1' state. Inputs P12-P17 are not valid in the parallel port mode. After a RESET, the microcontroller remains in the parallel port mode until the serial I/O mode is enabled.
- A 3 line parallel port P20-P22 with the same output configurations as P00-P07 but without high-current output;
- An external interrupt and test input INT/T0, which when used as a test input can be tested by the conditional jump instructions JT0 and JNT0;
- A test input T1, tested by the conditional jump instructions JT1 and JNT1. T1 can also be used as an input to the timer/event counter or to detect zero cross-over of slowly moving a.c. signals.

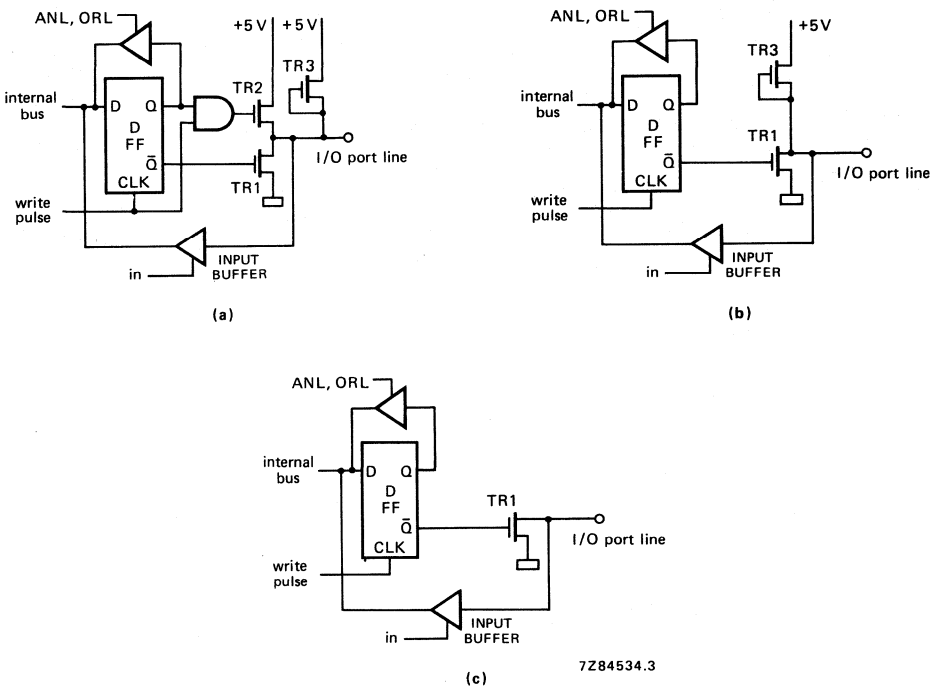


Fig. 5 Port option configurations.

Test input T1

The T1 input line can be used as:

- a test input for branch instructions,
- an input for zero voltage cross-over detection,
- an external input to the event counter.

An internal pull-up transistor is provided as a ROM mask option. This is useful when the input is from a switch or standard TTL output.

When T1 is used as a test input, the JT1 or JNT1 instructions test for a HIGH or a LOW respectively.

Zero cross-over detection

When used for zero-cross detection purposes, the T1 input must be coupled through a capacitor of typical value 1 μF and operation carried out using the T1 input without the pull-up transistor. The maximum input voltage amplitude is 3 V (peak-to-peak), with a maximum operational frequency of 1 kHz. The T1 input has an on-chip d.c. offset circuit which self-biases the input near to its exact switching level of 1 V. As a consequence a small change will cause a digital transition to occur. The switching level of the T1 input circuit is within the bias voltage of ±135 mV. Upon each positive cycle on the pin, the event counter is incremented and an overflow will set the timer flag TF. Zero cross-over detection used in conjunction with the timer/event counter interrupt, is useful in thyristor control of power equipment. Figure 6 illustrates, (a) the input waveform, (b) the input diagram and (c) the on-chip self-stabilized bias.

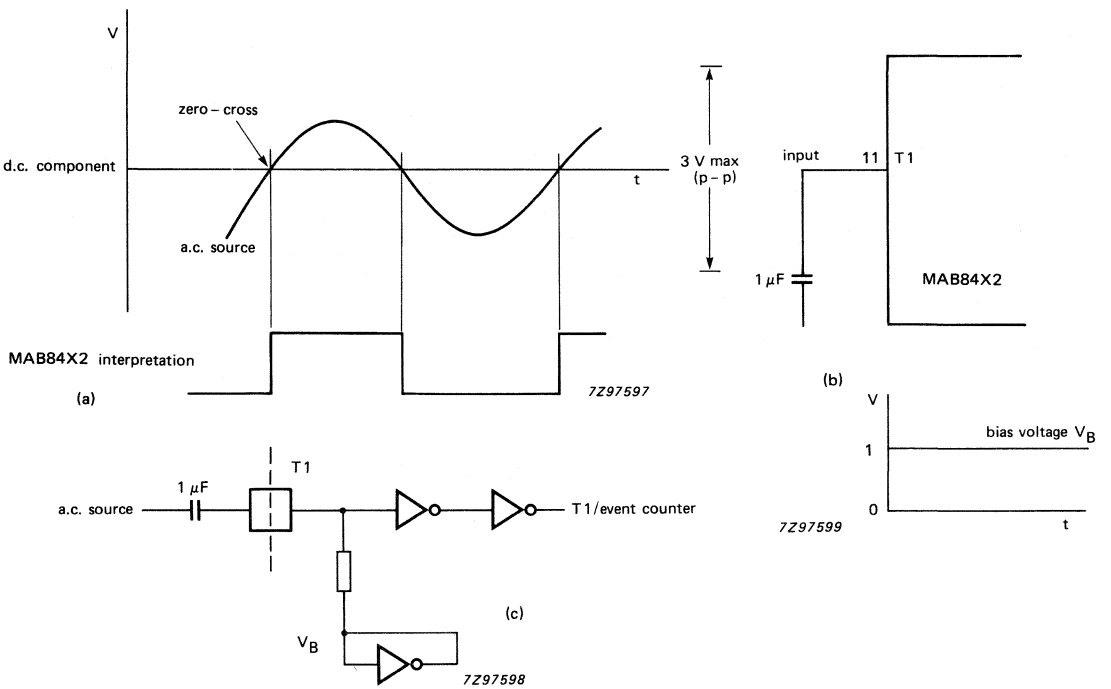


Fig. 6 Zero-cross voltage detection circuitry.

FUNCTIONAL DESCRIPTION (continued)

Timer/event counter

An 8-bit binary up-counter is provided. This can count external events, modulo-32 machine cycles, or machine cycles directly. When used as a timer, the input to the counter is either the overflow or input of a 5-bit prescaler. When used as an event counter, LOW to HIGH transitions on T1 (pin 13) are counted. The maximum rate at which the counter may be incremented is once every machine cycle (200 kHz for a 5 μs machine cycle). Fig. 7 illustrates the timer/event counter.

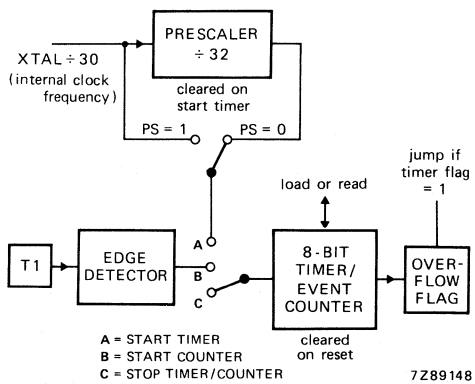


Fig. 7 Timer/event counter.

Interrupt system

External events and real-time on-chip peripherals require servicing by the CPU asynchronous to the execution of any particular section of code. To tie the asynchronous activities of these functions to normal program execution a multiple-source, single-level nested interrupt system is provided.

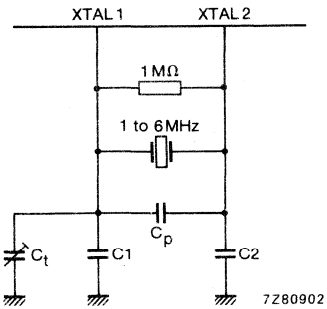
The MAB8422/8442 handles interrupts from three sources as follows:

- $\overline{\text{INT}}$/T0; externally via pin 9
- $\overline{\text{SIOINT}}$; from the internal serial I/O port
- TCNT interrupt; from the internal timer/event counter.

Each interrupt vectors to a separate location in the program memory for its service program. Each source can be individually enabled or disabled. When more than one interrupt occurs simultaneously, their priority will be: (1) external, (2) serial I/O and, (3) timer/event counter. An additional external interrupt can be created using the timer/event counter interrupt.

OSCILLATOR CIRCUITRY (see Fig. 8)

The clock frequency is determined by using the internal oscillator or by connecting an external clock to XTAL1. Where the internal oscillator is used the frequency is set by a crystal, a ceramic resonator or an inductor (each with associated capacitors) between XTAL1 and XTAL2. A machine cycle consists of 10 states, each state being 3 oscillator periods. The MAB8422/8442 has a dynamic logic, and therefore, for adequate refreshing the oscillator frequency must be at least 1 MHz.



1. Crystal - AT-cut
2. Ceramic resonator
   $C_1 = C_2 = 27$ pF
   C_t is optional
   $C_p < 6{,}75$ pF (parasitic capacitance)

Fig. 8(a) Quartz crystal or ceramic resonator mode.

If the frequency has to be trimmed, then a trimmer capacitor C_t should be connected in parallel with the fixed capacitor C_1.

Table 1 shows the LC values for timing generation with the LC oscillator.

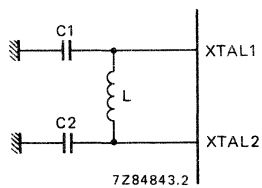
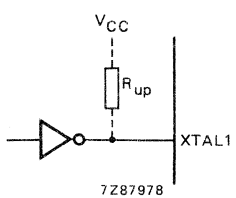


Fig. 8(b) LC pi-network.

Table 1 LC oscillator timing

frequency	C1 = C2	L
3,0 MHz	33 pF	100 μH
4,0 MHz	33 pF	56 μH
4,4 MHz	33 pF	47 μH
5,0 MHz	33 pF	33 μH
6,0 MHz	33 pF	22 μH



Drive XTAL1
Leave XTAL2 open

Driver may be high-speed CMOS or any TTL
t_r, $t_f < 10$ ns

Fig. 9 External drive.

RESET

A positive-going signal on the RESET input:

- sets the program counter to zero,
- selects location 0 of memory bank 0, and register bank 0,
- sets the stack pointer to zero (000); pointing to RAM address 8,
- disables the interrupts (external), timer and serial I/O),
- stops the timer/event counter, then sets it to zero,
- sets the timer prescaler to modulo-32,
- resets the timer flag,
- sets all ports to logic '1' (input mode),
- sets ports P10/SDA and P11/SCL to the parallel part mode and disables the serial I/O.

Automatic reset at power-up may be obtained by connecting the RESET pin to V_{CC} through a 1 μF capacitor C1, together with a diode to V_{SS} (cathode to RESET pin). This arrangement is satisfactory, if both the voltage (V_{CC}) rise time and the oscillator start-up time do not exceed either 1 or 10 ms respectively.

The power-on reset circuit is shown in Fig. 11; the input characteristics are shown in Fig. 12. At power-on the current drawn by RESET commences to charge the capacitor C1. The difference between this increasing capacitor voltage and V_{CC} is known as V_{RESET}. The charging circuit is designed to hold V_{RESET} above the lower threshold of a schmitt trigger arrangement, long enough to effect a complete reset. The minimum time required is the oscillator start-up time, plus two machine cycles.

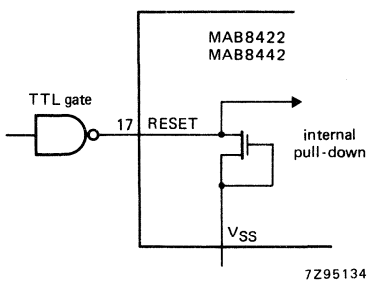


Fig. 10 External reset.

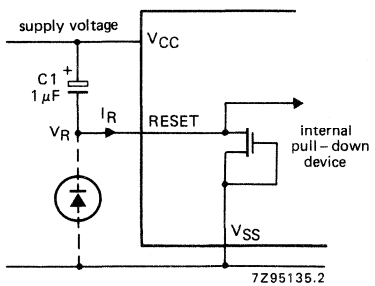


Fig. 11 Power-on reset circuitry.

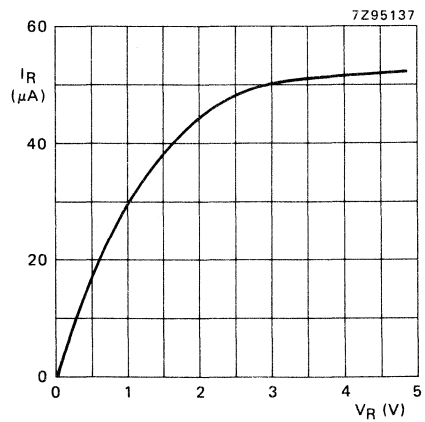


Fig. 12 Power-on reset input characteristics.

INSTRUCTION SET

The instruction set consists of over 80 one and two byte instructions and is based on the MAB8048 instruction set. New instructions include those for serial I/O operation and memory bank selection. Program code efficiency is high because all ROM locations on a 256 byte page require only a single byte address.

Table 2 gives the instruction set of the MAB84X2 family and Table 3 shows the instruction map. The following symbols and abbreviations are used.

symbol	description
A	the accumulator
AC	the auxiliary carry flag
addr	program memory address (11-bits)
Bb	bit designation (b = 0–7)
BS	the bank switch
C	carry flag
CLK	clock signal
CNT	event counter
D	nibble designation (4-bits)
DBF	program memory bank flip-flop
data	number of expression (8-bits)
F0, F1	flags 0 and 1
I	interrupt
$\overline{INT}$	external interrupt
P	'in-page' operation designation
Pp	port designation (p = 1, 2 or 4–7)
PSW	program status word
Rr	register designation (r = 0, 1 or 0–7)
SP	stack pointer
T	timer
TF	timer flag
T0, T1	test 0 an 1 inputs
#	immediate data prefix
@	indirect address prefix
S	current value of program counter
←	is replaced by
↔	is exchanged with

Table 2 Instruction set

	mnemonic	opcode (hex.)	bytes/ cycles	description	function		notes
ACCUMULATOR	ADD A, Rr	6*	1/1	Add register contents to A	(A)←(A) + (Rr)	r = 0–7	1
	ADD A, @Rr	60 61	1/1	Add RAM data, addressed by Rr, to A	(A)←(A) + ((R0)) (A)←(A) + ((R1))		1
	ADD A, #data	03 data	2/2	Add immediate data to A	(A)←(A) + data		1
	ADDC A, Rr	7*	1/1	Add carry and register contents to A	(A)←(A) + (Rr) + (C)	r = 0–7	1
	ADDC A, @Rr	70 71	1/1	Add carry and RAM data, addressed by Rr, to A	(A)←(A) + ((R0)) + (C) (A)←(A) + ((R1)) + (C)		1
	ADDC A, #data	13 data	2/2	Add carry and immediate data to A	(A)←(A) + data + (C)		1
	ANL A, Rr	5*	1/1	'AND' Rr with A	(A)←(A) AND (Rr)	r = 0–7	
	ANL A, @Rr	50 51	1/1	'AND' RAM data, addressed by Rr, with A	(A)←(A) AND ((R0)) (A)←(A) AND ((R1))		
	ANL A, #data	53 data	2/2	'AND' immediate data with A	(A)←(A) AND data		
	ORL A, Rr	4*	1/1	'OR' Rr with A	(A)←(A) OR (Rr)	r = 0–7	
	ORL A, @Rr	40 41	1/1	'OR' RAM data, addressed by Rr, with A	(A)←(A) OR ((R0)) (A)←(A) OR ((R1))		
	ORL A, #data	43 data	2/2	'OR' immediate data with A	(A)←(A) OR data		
	XRL A, Rr	D*	1/1	'XOR' Rr with A	(A)←(A) XOR (Rr)	r = 0–7	
	XRL A, @Rr	D0 D1	1/1	'XOR' RAM, addressed by Rr, with A	(A)←(A) XOR ((R0)) (A)←(A) XOR ((R1))		
	XRL A, #data	D3 data	2/2	'XOR' immediate data with A	(A)←(A) XOR data		
	INC A	17	1/1	increment A by 1	(A)←(A) + 1		
	DEC A	07	1/1	decrement A by 1	(A)←(A) – 1		
	CLR A	27	1/1	clear A to zero	(A)←0		
	CPL A	37	1/1	one's complement A	(A)←NOT(A)		
	RL A	E7	1/1	rotate A left	$(A_{n+1})\leftarrow(A_n)$ $(A_0)\leftarrow(A_7)$	n = 0–6	

	mnemonic	opcode (hex.)	bytes/ cycles	description	function		notes
ACCUMULATOR (cont.)	RLC A	F7	1/1	rotate A left through carry	$(A_{n+1})\leftarrow A_n$ $(A_0)\leftarrow(C)$, $(C)\leftarrow(A_7)$	n = 0–6	2
	RR A	77	1/1	rotate A right	$(A_n)\leftarrow(A_{n+1})$ $(A_7)\leftarrow(A_0)$	n = 0–6	
	RRC A	67	1/1	rotate A right through carry	$(A_n)\leftarrow(A_{n+1})$ $(A_7)\leftarrow(C)$, $(C)\leftarrow(A_0)$	n = 0–6	2
	DA A	57	1/1	decimal adjust A			2
	SWAP A	47	1/1	swap nibbles of A	$(A_{4-7})\leftrightarrow(A_{0-3})$		
DATA MOVES	MOV A, Rr	F*	1/1	move register contents to A	(A)←(Rr)	r = 0–7	
	MOV A, @Rr	F0 F1	1/1	move RAM data, addressed by Rr, to A	(A)←((R0)) (A)←((R1))		
	MOV A, #data	23 data	2/2	move immediate data to A	(A)←data		
	MOV Rr, A	A*	1/1	move accumulator contents to register	(Rr)←(A)	r = 0–7	
	MOV @Rr, A	A0 A1	1/1	move accumulator contents to RAM location addressed by Rr	((R0))←(A) ((R1))←(A)		
	MOV Rr, #data	B* data	2/2	move immediate data to Rr	(Rr)←data		
	MOV @Rr, #data	B0 data B1 data	2/2	move immediate data to RAM location addressed by Rr	((R0))←data ((R1))←data		
	XCH A, Rr	2*	1/1	exchange accumulator contents with Rr	$(A)\leftrightarrow(Rr)$	r = 0–7	
	XCH A, @Rr	20 21	1/1	exchange accumulator contents with RAM data addressed by Rr	$(A)\leftrightarrow((R0))$ $(A)\leftrightarrow((R1))$		
	XCHD A, @Rr	30 31	1/1	exchange lower nibbles of A and RAM data addressed by Rr	$(A_{0-3})\leftrightarrow((R0_{0-3}))$ $(A_{0-3})\leftrightarrow((R1_{0-3}))$		
	MOV A, PSW	C7	1/1	move PSW contents to accumulator	(A)←(PSW)		
	MOV PSW, A	D7	1/1	move accumulator bit 3 to PSW_3	$(PSW_3)\leftarrow(A_3)$		3
	MOVP A, @A	A3	1/2	move indirectly addressed data in current page to A	$(PC_{0-7})\leftarrow(A)$, $(A)\leftarrow((PC))$		

Table 2 Instruction set (continued)

FLAGS	CLR C	97	1/1	clear carry bit	(C)←0	2
	CPL C	A7	1/1	complement carry bit	(C)←NOT(C)	2
REGISTER	INC Rr	1*	1/1	increment register by 1	(Rr)←(Rr) + 1 r = 0–7	
	INC @Rr	10 11	1/1	increment RAM data, addressed by Rr, by 1	((R0))←((R0)) + 1 ((R1))←((R1)) + 1	
	DEC Rr	C*	1/1	decrement register by 1	(Rr)←(Rr) – 1 r = 0–7	
	DEC @Rr	C0 C1	1/1	decrement RAM data, addressed by Rr, by 1	((R0))←((R0)) – 1 ((R1))←((R1)) – 1	
BRANCH	JMP addr	• 4 address	2/2	unconditional jump within a 2 K bank	(PC_{8-10})←$addr_{8-10}$ (PC_{0-7})←$addr_{0-7}$ (PC_{11-12})←MBFF 0–1	
	JMPP @A	B3	1/2	indirect jump within a page	(PC_{0-7})←((A))	
	DJNZ Rr, addr	E* address	2/2	decrement Rr by 1 and jump if not zero to addr	(Rr)←(Rr) – 1 r = 0–7 if (Rr) not zero (PC_{0-7})←addr	
	DJNZ @Rr, addr	E0 address	2/2	decrement RAM data, addressed by Rr by 1 and jump if not zero to addr	((R0))←((R0)) – 1 if ((R0)) not zero (PC_{0-7})←addr	
		E1 address			((R1))←((R1)) – 1 if ((R1)) not zero (PC_{0-7})←addr	
	JBb addr	▲ 2 address	2/2	jump to addr if Acc. bit b = 1	if b = 1 : (PC_{0-7})←addr b = 0–7	
	JC addr	F6 address	2/2	jump to addr if C = 1	if C = 1 : (PC_{0-7})←addr	
	JNC addr	E6 address	2/2	jump to addr if C = 0	if C = 0 : (PC_{0-7})←addr	
	JZ addr	C6 address	2/2	jump to addr if A = 0	if A = 0 : (PC_{0-7})←addr	
	JNZ addr	96 address	2/2	jump to addr if A is NOT zero	if A≠0 : (PC_{0-7})←addr	
	JT0 addr	36 address	2/2	jump to addr if T0 = 1	if T0 = 1: (PC_{0-7})←addr	
	JNT0 addr	26 address	2/2	jump to addr if T0 = 0	if T0 = 0: (PC_{0-7})←addr	
	JT1 addr	56 address	2/2	jump to addr if T1 = 1	if T1 = 1: (PC_{0-7})←addr	
	JNT1 addr	46 address	2/2	jump to addr if T1 = 0	if T1 = 0: (PC_{0-7})←addr	
	JTF addr	16 address	2/2	jump to addr if Timer Flag = 1	if TF = 1: (PC_{0-7})←addr	4
	JNTF addr	06 address	2/2	jump to addr if Timer Flag = 0	if TF = 0: (PC_{0-7})←addr	

	mnemonic	opcode (hex.)	bytes/ cycles	description	function	notes
TIMER/EVENT COUNTER	MOV A, T	42	1/1	move timer/event counter contents to accumulator	(A)←(T)	
	MOV T, A	62	1/1	move accumulator contents to timer/event counter	(T)←(A)	
	STRT CNT	45	1/1	start event counter		
	STRT T	55	1/1	start timer		
	STOP TCNT	65	1/1	stop timer/event counter		
	EN TCNTI	25	1/1	enable timer/event counter interrupt		
	DIS TCNTI	35	1/1	disable timer/event counter interrupt		
CONTROL	EN I	05	1/1	enable external interrupt		
	DIS I	15	1/1	disable external interrupt		
	SEL RB0	C5	1/1	select register bank 0	(RBS)←0	5
	SEL RB1	D5	1/1	select register bank 1	(RBS)←1	5
	SEL MB0	E5	1/1	select program memory bank 0	(MBFF0)←0, (MBFF1)←0	
	SEL MB1	F5	1/1	select program memory bank 1	(MBFF0)←1, (MBFF1)←0	
	SEL MB2	A5	1/1	select program memory bank 2	(MBFF0)←0, (MBFF1)←1	
	SEL MB3	B5	1/1	select program memory bank 3	(MBFF0)←1, (MBFF1)←1	
SUBROUTINE	CALL addr	▲ 4 address	2/2	jump to subroutine	((SP))←(PC), ($PSW_{4, 6, 7}$) (SP)←(SP) + 1 (PC_{8-10})←$addr_{8-10}$ (PC_{0-7})←$addr_{0-7}$ (PC_{11-12})←MBFF 0–1	6
	RET	83	1/2	return from subroutine	(SP)←(SP) −1 (PC)←((SP))	6
	RETR	93	1/2	return from interrupt and restore bits 4, 6, 7 of PSW	(SP)←(SP) −1 ($PSW_{4, 6, 7}$) + (PC)←((SP))	6

Table 2 Instruction set (continued)

PARALLEL INPUT/OUTPUT	IN A, Pp	08 09 0A	1/2	input port p data to accumulator	(A)←(P0) (A)←(P1) (A)←(P2)	7
	OUTL Pp, A	38 39 3A	1/2	output accumulator data to port p	(P0)←(A) (P1)←(A) (P2)←(A)	
	ANL Pp, #data	98 data 99 data 9A data	2/2	AND port p data with immediate data	(P0)←(P0) AND data (P1)←(P1) AND data (P2)←(P2) AND data	
	ORL Pp, #data	88 data 89 data 8A data	2/2	OR port p data with immediate data	(P0)←(P0) OR data (P1)←(P1) OR data (P2)←(P2) OR data	
	OUTL PO,A	90	1/2	Output accumulator data to port ϕ	(P0)←(A)	8
SERIAL INPUT/OUTPUT	EN SI	85	1/1	enable serial I/O interrupt		
	DIS SI	95	1/1	disable serial I/O interrupt		
	NOP	00	1/1	no operation		

Notes to Table 2.

1. PSW CY, AC affected
2. PSW CY affected
3. PSW PS affected
4. Execution of JTF and JNTF instructions resets the Timer Flag (TF).
5. PSW RBS affected
6. PSW SP_0, SP_1, SP_2 affected
7. (A) = 1111 P23, P22, P21, P20.
8. Only for software-transfer from the MAB8021.

* : 8, 9, A, B, C, D, E, F
● : 0, 2, 4, 6, 8, A, C, E
▲ : 1, 3, 5, 7, 9, B, D, F

TABLE 3 MAB8422/8442 INSTRUCTION MAP

first hexadecimal character of opcode ↓

second hexadecimal character of opcode →

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NOP			ADD A,#data	JMP page 0	EN I	JNTF addr	DEC A	IN A,Pp							
1	INC @Rr		JB0 addr	ADDC A,#data	CALL page 0	DIS I	JTF addr	INC A	INC Rr							
2	XCH A,@Rr			MOV A,#data	JMP page 1	EN TCNTI	JNTO addr	CLR A	XCH A,Rr							
3	XCHD A,@Rr		JB1 addr		CALL page 1	DIS TCNTI	JTO addr	CPL A	OUTL Pp,A							
4	ORL A,@Rr		MOV A,T	ORL A,#data	JMP page 2	STRT CNT	JNT1 addr	SWAP A	ORL A,Rr							
5	ANL A,@Rr		JB2 addr	ANL A,#data	CALL page 2	STRT T	JT1 addr	DA A	ANL A,Rr							
6	ADD A,@Rr		MOV T,A		JMP page 3	STOP TCNT		RRC A	ADD A,Rr							
7	ADDC A,@Rr		JB3 addr		CALL page 3			RR A	ADDC A,Rr							
8				RET	JMP page 4	EN SI			ORL Pp,#data							
9	OUTL PO,A		JB4 addr	RETR	CALL page 4	DIS SI	JNZ addr	CLR C	ANL Pp,#data							
A	MOV @Rr,A			MOVP A,@A	JMP page 5	SEL MB2		CPL C	MOV Rr,A							
B	MOV @Rr,#data		JB5 addr	JMPP @A	CALL page 5	SEL MB3			MOV Rr,#data							
C	DEC @Rr				JMP page 6	SEL RB0	JZ addr	MOV A,PSW	DEC Rr							
D	XRL A,@Rr		JB6 addr	XRL A,#data	CALL page 6	SEL RB1		MOV PSW,A	XRL A,Rr							
E	DJNZ @Rr,addr				JMP page 7	SEL MB0	JNC addr	RL A	DJNZ Rr, addr							
F	MOV A,@Rr		JB7 addr		CALL page 7	SEL MB1	JC addr	RLC A	MOV A,Rr							

ABSOLUTE MAXIMUM RATINGS

Limiting values in accordance with the Absolute Maximum System (IEC 134)

Input voltage on any pin with respect to ground (V_{SS})		V_I	−0,5 to +7	V
Total power dissipation		P_{tot}	1	W
Input/output current for all pins except port 0		I_I, I_O	max. 10	mA
Input/output current for port 0		I_I, I_O	max. 20	mA
Storage temperature		T_{stg}	−65 to +150	°C
Operating temperature	standard	T_{amb}	0 to +70	°C
	extended	T_{amb}	−40 to +85	°C
	automotive	T_{amb}	−40 to +110	°C

D.C. CHARACTERISTICS

V_{CC} = 5 V (± 10%); V_{SS} = 0 V

parameter	symbol	min.	max.	unit	conditions
Supply current					
MAB8422/42	I_{CC}	−	70	mA	at 0 °C
MAF8422/42	I_{CC}	−	90	mA	at −40 °C
MAF84A22/42	I_{CC}	−	90	mA	at −40 °C
Inputs					
Input voltage LOW (except P10/SDA and P11/SCL)	V_{IL}	−0,5	0,8	V	
Input voltage LOW (P10/SDA and P11/SCL)	V_{IL1}	−0,5	1,5	V	
Input voltage HIGH all inputs except XTAL 1, P10/SDA and P11/SCL	V_{IH}	2,0	V_{CC} + 0,5	V	
Input voltage HIGH to XTAL 1, P10/SDA and P11/SCL	V_{IH1}	3,0	V_{CC} + 0,5	V	
Outputs					
Output voltage LOW (P0 only)	V_{OL}		1,0	V	I_{OL} = 10 mA
Output voltage LOW (P10/SDA and P11/SCL)	V_{OL1}		0,45	V	I_{OL} = 5 mA
Output voltage LOW (P0 and P2)	V_{OL2}		0,45	V	I_{OL} = 1,6 mA
Output voltage HIGH (all outputs unless open-drain)	V_{OH}	2,4		V	I_{OH} = −50 μA
Output leakage current	$\pm I_{OL}$		10	μA	$V_{CC} > V_I > V_{SS}$

A.C. CHARACTERISTICS

parameter	symbol	min.	max.	unit
Frequency				
MAB/MAF8422/42	f_{XTAL}	1	6	MHz
MAF84A22/42	f_{XTAL}	1	5	MHz
Cycle time				
MAB/MAF8422/42	t_{CY}	5	30	μs
MAF84A22/42	t_{CY}	6	30	μs

T1 ZERO-CROSS CHARACTERISTICS

T_{amb} = 0 to + 70 °C; V_{CC} = 5 V ± 10%; V_{SS} = 0 V; C_L = 80 pF

parameter	symbol	min.	max.	unit	conditions
Zero-cross detection input (T1) peak-to-peak	$V_{ZX(p-p)}$	1	3	V	a.c. coupled, C = 1 μF
Zero-cross accuracy	A_{ZX}	–	± 135	mV	50 Hz sine wave
Zero-cross detection input frequency (T1)	F_{ZX}	0,05	1	kHz	

DEVELOPMENT DATA

Purchase of Philips' I²C components conveys a license under the Philips' I²C patent to use the components in the I²C-system provided the system conforms to the I²C specification defined by Philips.

SINGLE-CHIP 8-BIT MICROCONTROLLER

DESCRIPTION

The PCB80C51 family of single-chip 8-bit microcontrollers is manufactured in an advanced CMOS process. The family consists of the following members:

- PCB80C31BH: ROM-less version of the PCB80C51BH
- PCB80C51BH: 4 K bytes mask-programmable ROM, 128 bytes RAM

In the following, the generic term "PCB80C51BH" is used to refer to both family members.

The device provides hardware features, architectural enhancements and new instructions to function as a controller for applications requiring up to 64 K bytes of program memory and/or up to 64 K bytes of data storage.

The PCB80C51BH contains a non-volatile 4 K x 8 read-only program memory (not ROM-less version); a volatile 128 x 8 read/write data memory; 32 I/O lines; two 16-bit timer/event counters; a five-source, two-priority-level, nested interrupt structure; a serial I/O port for either multi-processor communications, I/O expansion, or full duplex UART; and on-chip oscillator and timing circuits. For systems that require extra capability, the PCB80C51BH can be expanded using standard TTL compatible memories and logic.

The PCB80C31BH/80C51BH has two software selectable modes of reduced activity for further power reduction — Idle and Power Down.
The Idle modes freezes the CPU while allowing the RAM, timers, serial port and interrupt system to continue functioning.
The Power Down mode saves the RAM contents but freezes the oscillator causing all other chip functions to be inoperative.

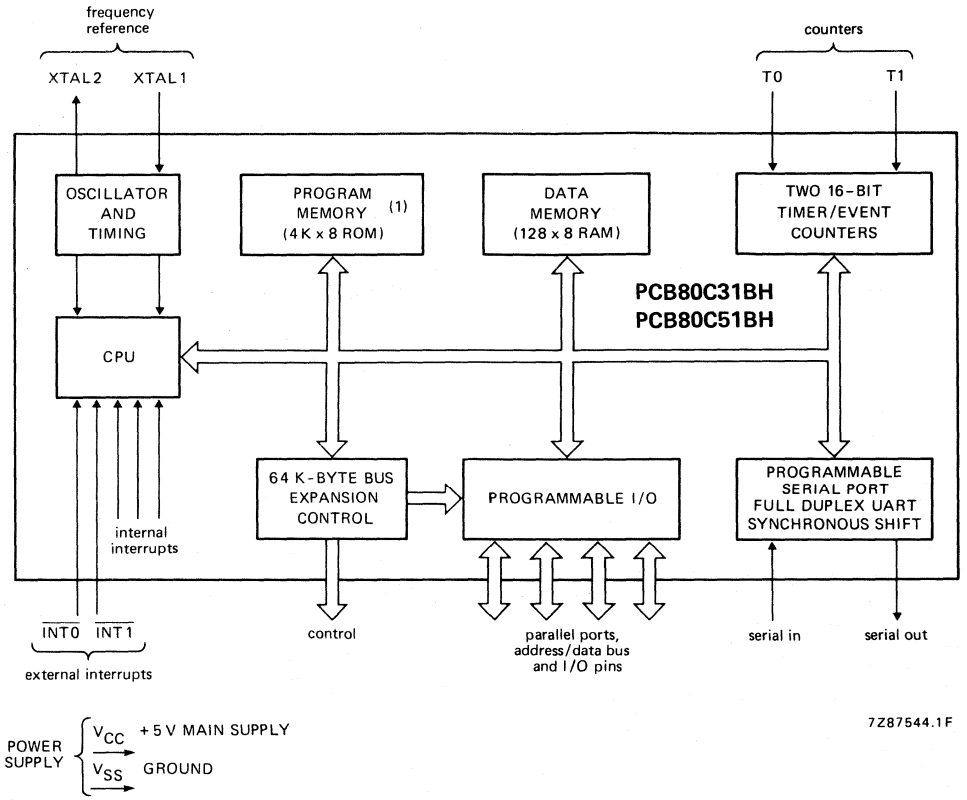
The device also functions as an arithmetic processor having facilities for both binary and BCD arithmetic plus bit-handling capabilities. The instruction set consists of 255 instructions; 44% one-byte, 41% two-byte and 15% three-byte. With a 12 MHz crystal for example, 58% of the instructions are executed in 1 μs and 40% in 2 μs. Multiply and divide instructions require 4 μs. Multiply, divide, subtract and compare are among the many instructions added to the standard PCB80C48 instruction set.
Software development to be announced: PCB85C51 in piggy-back.

Features

- 4 K x 8 ROM (80C51BH only), 128 x 8 RAM
- Four 8-bit ports, 32 I/O lines
- Two 16-bit timer/event counters
- Full-duplex serial port
- External memory expandable to 128 K, external ROM up to 64 K and/or external RAM up to 64 K
- Boolean processing
- 218 bit-addressable locations
- On-chip oscillator
- Five-source interrupt structure with two priority levels
- 58% of instructions executed in 1 μs; multiply and divide in 4 μs; all others executed in 2 μs (at 12 MHz clock)
- Enhanced architecture with:
  non-page-oriented-instructions
  direct addressing
  four 8-byte + 1-byte register banks
  stack depth up to 128-bytes
  multiply, divide, subtract and compare instructions.
- Available as
  PCB80C31/51BH with 1,2 to 16 MHz
  PCF80C31/51BH with 1,2 to 12 MHz

PACKAGE OUTLINES

PCB/PCF80C31/51BHP: 40-lead DIL; plastic (SOT-129).
PCB/PCF80C31/51BHWP: 44-lead PLCC; plastic, leaded-chip-carrier (SOT-187A).



(1) PCB80C51BH only.

Fig. 1 Block diagram.

type	temp. range	frequency range	$I_{CC\,max}$ at 5,5 V
PCB80C51BH PCB80C31BH	0 – 70 °C	1,2 – 16 MHz	23 mA*
PCF80C51BH PCF80C31BH	–40 – + 85 °C	1,2 – 12 MHz	18 mA*

* Preliminary value.

DEVELOPMENT DATA

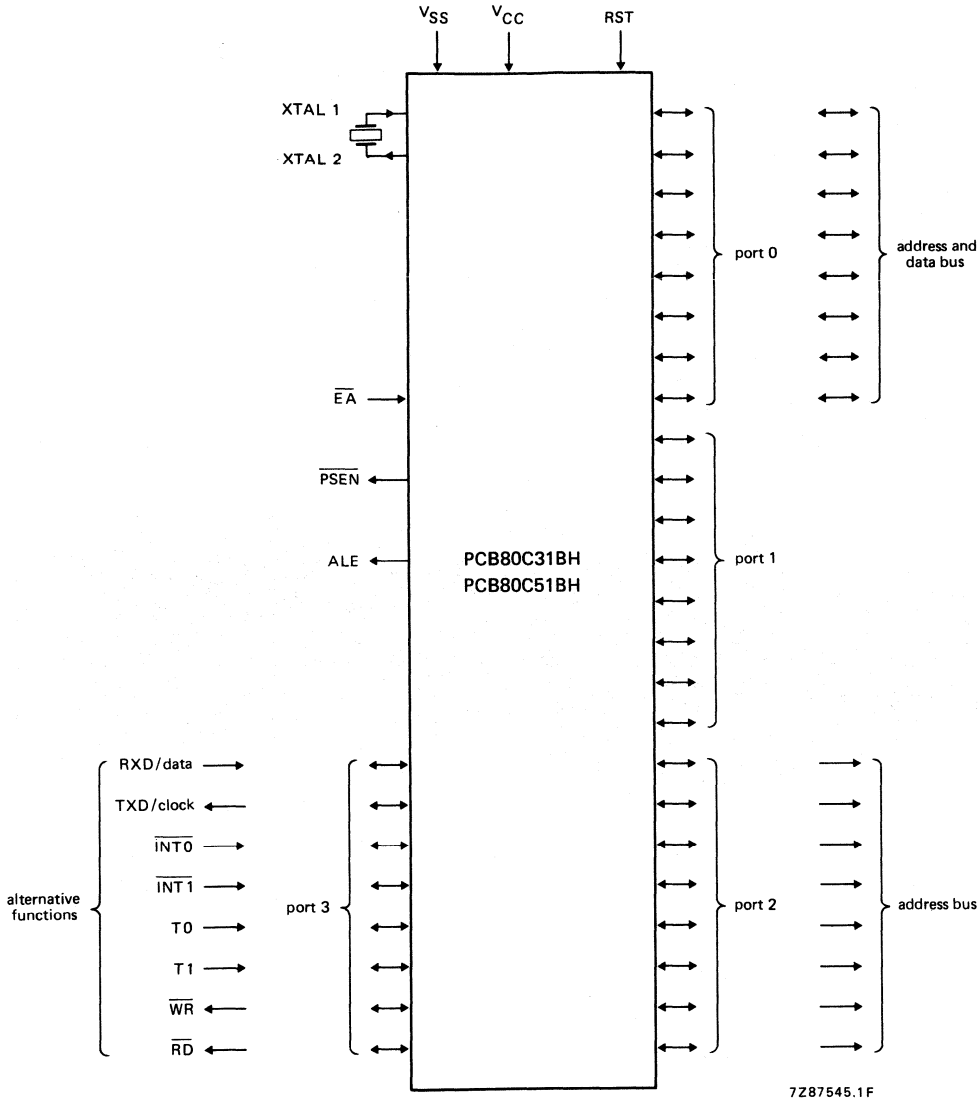


Fig. 2 Functional diagram.

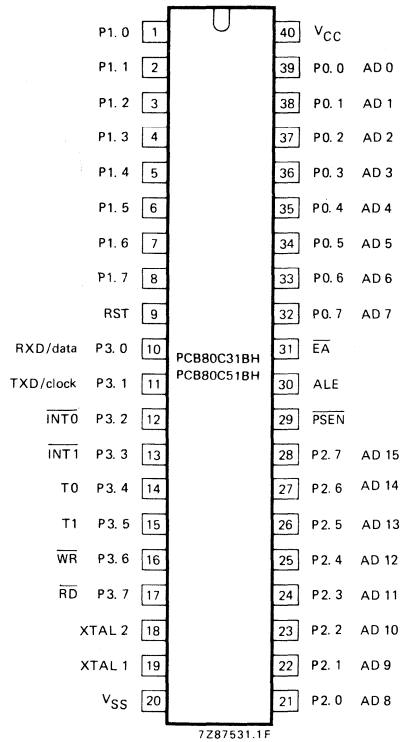


Fig. 3a Pinning diagram for PCB80C31BH/51BHP.

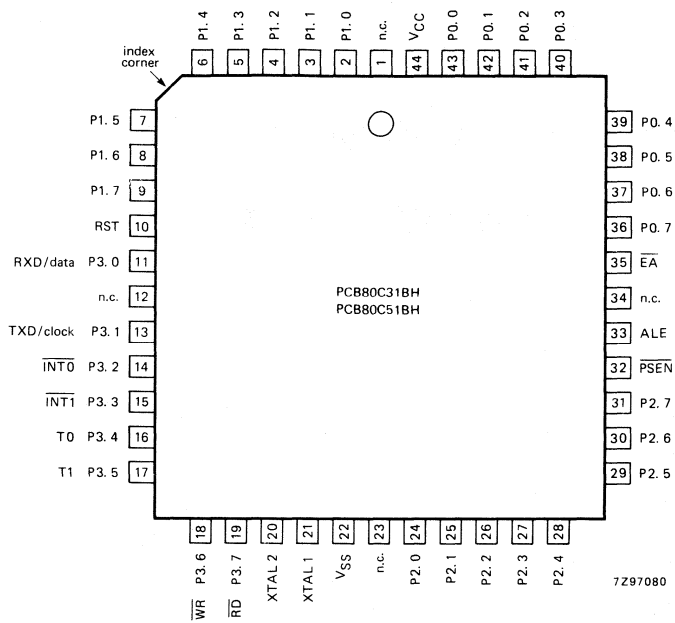


Fig. 3b Pinning diagram for PCB80C31BH/51BHWP.

PINNING (PCB80C31BH/51BHP)

1–8	P1.0–P1.7	**Port 1**: 8-bit quasi-bidirectional I/O port. Port 1 can sink/source one TTL (= 4 LS TTL) input. It can drive CMOS inputs without external pull-ups.
9	RST	**RESET**: a high level on this pin for two machine cycles while the oscillator is running resets the device. An internal pull-down permits Power-On reset using only a capacitor connected to the V_{CC}.
10–17	P3.0–P3.7	**Port 3**: 8-bit quasi-bidirectional I/O port with internal pull-ups. It also serves the following alternative functions:

Port pin	*Alternative function*
P3.0	**RXD/data**: serial port receiver data input (asynchronous) or data input/output (synchronous)
P3.1	**TXD/clock**: serial port transmitter data output (asynchronous) or clock output (synchronous)
P3.2	**$\overline{\text{INT0}}$**: external interrupt 0 or gate control input for timer/event counter 0
P3.3	**$\overline{\text{INT1}}$**: external interrupt 1 or gate control input for timer/event counter 1

		P3.4 **T0** : external input for timer/event counter 0
		P3.5 **T1** : external input for timer/event counter 1
		P3.6 **$\overline{WR}$**: external data memory write strobe
		P3.7 **$\overline{RD}$** : external data memory read strobe
		Operation of an alternative function is determined by the relevant output latch programmed to logic 1. Port 3 can sink/source one TTL input. It can drive CMOS inputs without external pull-ups.
18	XTAL 2	**Crystal input 2**: output of the inverting amplifier that forms the oscillator. Left open-circuit when an external oscillator is used (see figures 8 and 9).
19	XTAL 1	**Crystal input 1**: input to the inverting amplifier that forms the oscillator, and input to the internal clock generator. Receives the external oscillator signal when an external oscillator is used (see figures 8 and 9).
20	V_{SS}	**Ground**: circuit ground potential.
21–28	P2.0–P2.7	**Port 2**: 8-bit quasi-bidirectional I/O port with internal pull-ups. Port 2 can sink/source one TTL input. It can drive CMOS inputs without external pull-ups. During the access to external memories (RAM/ROM) that uses 16-bit addresses (MOVX @DPTR) Port 2 emits the high-order address byte. When used for an external RAM with an 8-bit address (MOVX @Ri) Port 2 emits the contents of the P2 Special Function Register.
29	$\overline{PSEN}$	**Program Store Enable output**: read strobe to the external Program Memory. It is activated twice each machine cycle during fetches from external Program Memory. When executing out of external Program Memory two activations of $\overline{PSEN}$ are skipped during each access to external Data Memory. $\overline{PSEN}$ is not activated (remains HIGH) during fetches from internal Program Memory. $\overline{PSEN}$ can sink/source 8 LS TTL inputs. It can drive CMOS inputs without an external pull-up.
30	ALE	**Address Latch Enable output**: latches the low byte of the address during accesses to external memory in normal operation. It is activated every six oscillator periods except during an external data memory access. ALE can sink/source 8 LS TTL inputs. It can drive CMOS inputs without an external pull-up.
31	$\overline{EA}$	**External Access input**: When $\overline{EA}$ is held at a TTL high level the CPU executes out of the internal Program Memory (ROM), provided the Program Counter is less than 4096. When $\overline{EA}$ is held at a TTL low level, the CPU executes out of external Program Memory. $\overline{EA}$ is not allowed to float.
32–39	P0.7–P0.0	**Port 0**: 8-bit open drain bidirectional I/O port. It is also the multiplexed low-order address and data bus during accesses to external memory (during these accesses it activates internal pull-ups). Port 0 can sink/source eight TTL inputs.
40	V_{CC}	**Power supply**: + 5 V power supply pin during normal operation, Idle mode and Power Down mode.

DEVELOPMENT DATA

FUNCTIONAL DESCRIPTION

General

The PCB80C51BH is a stand-alone high-performance microcontroller designed for use in real-time applications such as instrumentation, industrial control and intelligent computer peripherals.

The device provides hardware features, architectural enhancements and new instructions to function as a controller for applications requiring up to 64 K bytes of program memory and/or up to 64 K bytes of data storage.

The PCB80C31BH is a control-oriented CPU without on-chip program memory. It can address 64 K bytes of external program memory in addition to 64 K bytes of external data memory. The PCB80C51BH is a PCB80C31BH with the lower 4 K bytes of program memory filled with on-chip mask programmable ROM. For systems requiring extra capability, the PCB80C51BH can be expanded using standard memories and peripherals.

The two pin-compatible versions of this component reduce development problems to a minimum and provide maximum flexibility. The PCB80C51BH is for low-cost, high volume production and the PCB80C31BH for applications requiring the flexibility of external program memory which can be easily modified and updated in the field.

The PCB80C51BH contains a non-volatile 4 K x 8 read-only program memory; a volatile 128 x 8 read/write data memory; 32 I/O lines; two 16-bit timer/event counters; a five-source, two-priority-level, nested interrupt structure; a serial I/O port for either multi-processor communications, I/O expansion, or full duplex UART; and on-chip oscillator and timing circuits.

Central processing unit

The central processing unit (CPU) manipulates operands in four memory spaces. These are the 64 K-byte external data memory, 384-byte internal data memory, the 64 K-byte internal and external program memory and 16-bit program counter spaces. The internal data memory address space is sub-divided into the 256-byte internal data RAM and 128-byte special function register (SFR) address spaces, as shown in Fig. 4.

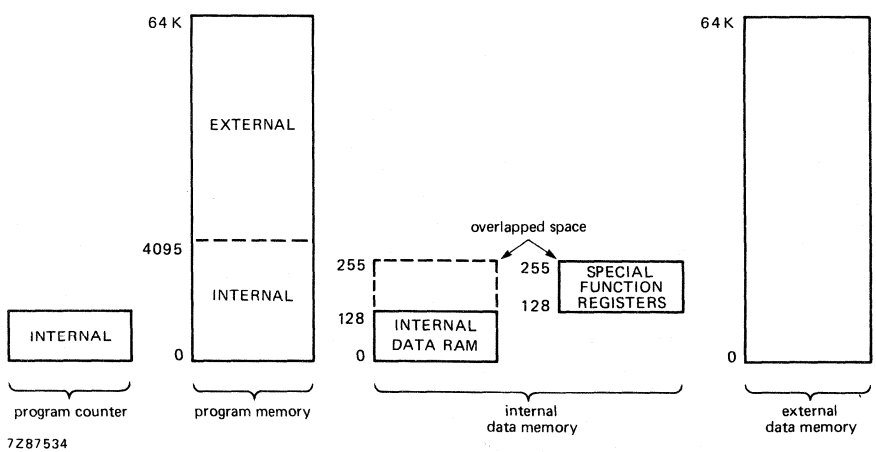


Fig. 4 Memory map.

The internal data RAM contains four register banks (each with eight registers), 128 addressable bits, and the stack. The stack depth is limited by the available internal data RAM and its location is determined by the 8-bit stack pointer. All registers except the program counter and the four 8-register banks reside in the special function register address space. These memory mapped registers include arithmetic registers, pointers, I/O ports, interrupt system registers, timers and serial port. There are 128 addressable bite locations in the SFR address space.

The PCB80C51BH contains 128 bytes of internal data RAM and 20 special function registers. It provides a non-paged program memory address space to accommodate relocatable code. Conditional branches are performed relative to the program counter. The register-indirect jump permits branching relative to a 16-bit base register with an offset provided by an 8-bit index register. 16-bit jumps and calls permit branching to any location in the contiguous 64 K program memory address space.

The PCB80C51BH has five methods for addressing source operands:

- Register.
- Direct.
- Register-Indirect.
- Immediate.
- Base-Register-plus Index-Register-Indirect.

The first three methods can be used for addressing destination operands. Most instructions have a "destination/source" field that specifies the data type, addressing methods and operands involved. For operations other than moves, the destination operand is also a source operand.

Access addressing is as follows:

- Registers in the four 8-register banks through Register, Direct, or Register-Indirect.
- 128 bytes of internal data RAM through Direct or Register-Indirect.
- Special function registers through Direct.
- External data memory through Register-Indirect.
- Program memory look-up tables through Base-Register-plus Index-Register-Indirect.

The PCB80C51BH is classified as an 8-bit device since the internal ROM, RAM, Special Function Registers (SFR), Arithmetic Logic Unit (ALU), and external data bus are each 8-bits wide. It performs operations on bit, nibble, byte and double-byte data types.

Facilities are available for byte transfer, logic, and integer arithmetic operations, Data transfer, logic, and conditional branch operations can be performed directly on Boolean variables to provide excellent bit handling.

DEVELOPMENT DATA

I/O facilities

The PCB80C51BH has 32 I/O lines treated as 32 individual addressable bits and as four parallel 8-bit addressable ports. Ports 0, 1, 2 and 3 perform the following alternate functions:

- Port 0; provides the multiplexed low-order address and data bus used for expanding the PCB80C51BH with standard memories and peripherals.
- Port 2; provides the high-order address bus when expanding the PCB80C51BH with external program memory or more than 256 bytes of external data memory.
- Port 3; pins can be configured individually to provide:

  external interrupt requests inputs
  counter inputs
  serial port receiver input and transmitter output
  control signals to READ and WRITE to external data memory

The generation or use of a Port 3 pin as an alternate function is carried out automatically by the PCB80C51BH provided the pin is loaded with a HIGH content.

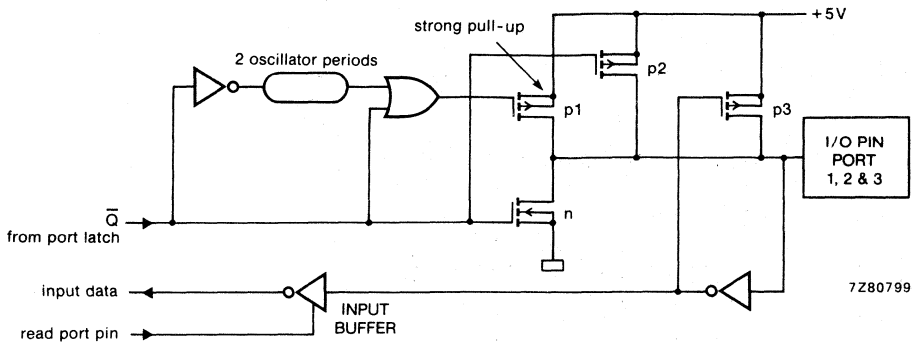


Fig. 5 I/O buffers in the PCB80C51BH (Ports 1, 2 and 3).

Timer/event counters

The PCB80C51BH contains two 16-bit registers, Timer 0 and Timer 1, that can be used as timers or event counters to carry out the following functions:

- Measure time intervals and pulse durations.
- Count events.
- Generate interrupt requests.

Each timer/event counter can be programmed independently to operate in three modes:

- Mode 0; 8-bit timer or 8-bit counter each with divide by 32 prescaler.
- Mode 1; 16-bit time-interval or event counter.
- Mode 2; 8-bit time-interval or event counter with automatic reload upon overflow.

Counter 0 can be programmed to operate in an additional mode as follows:

- Mode 3; one 8-bit time-interval or event counter and one 8-bit time-interval counter.

When counter 0 is in Mode 3, counter 1 can be programmed to operate in Modes 0, 1 or 2 but cannot set an interrupt request flag or generate an interrupt. However the overflow from counter 1 can be used to pulse the serial Port transmission-rate generator.

The frequency handling range of these counters with a 12 MHz crystal for example is as follows:

- Up to 1 MHz when programmed for an input that is a division by 12 of the oscillator frequency.
- 0 Hz to an upper limit of 150 kHz to 0,5 MHz when programmed for external inputs.

Both internal and external inputs can be gated to the counter by a second external source for directly measuring pulse durations.

The counters are started and stopped under software control. Each one sets its interrupt request flag when it overflows from all 1's to all 0's (or automatic reload value).

On-chip peripheral functions

In addition to the CPU and memories, an interrupt system, extensive I/O facilities, and several peripheral functions are integrated on-chip to relieve the CPU of repetitious, complicated or time-critical tasks and to permit stringent real-time control of external system interfaces. The I/O facilities include the I/O pins, parallel ports, bidirectional address/data bus and the serial port for I/O expansion. The CPU peripheral functions integrated on-chip are the two 16-bit timer/event counters and the serial port.

Idle and Power-down operation (see Fig. 6)

The Power-down operation freezed the oscillator. The Idle mode operation allows the interrupt, serial port and timer blocks to continue to function while the clock to the CPU is halted.

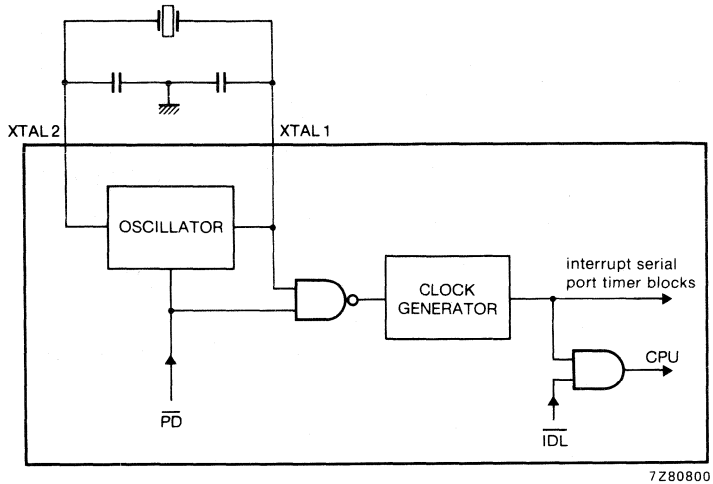


Fig. 6 Internal Idle and Power-down clock configuration.

Power control register (PCON)

These special modes are activated by software via the Special Function Register PCON. Its hardware address is 87H. PCON is not bit addressable.

MSB							LSB
SMOD	–	–	–	GF1	GF0	PD	IDL

symbol	position	name and function
SMOD	PCON.7	Double Baud rate bit when set to logic 1 the baud rate is doubled when the serial port is being used in either modes 1, 2 or 3
–	PCON.6	(reserved)
–	PCON.5	(reserved)
–	PCON.4	(reserved)
GF1	PCON.3	general-purpose flag bit
GF0	PCON.2	general-purpose flag bit
PD	PCON.1	Power-down bit setting this bit activates power-down operation
IDL	PCON.0	Idle mode bit setting this bit activates the idle mode operation

If logic1s are written to PD and IDL at the same time, PD takes precedence. The reset value of PCON is (0XXX0000).

DEVELOPMENT DATA

Idle mode

The instruction that sets PCON.0 is the last instruction executed in the normal operating mode before Idle mode is activated. Once in the Idle mode, the CPU status is preserved in its entirety: the Stack Pointer, Program Counter, Program Status Word, Accumulator, RAM and all other registers maintain their data during Idle mode. The status of the external pins during Idle mode is shown in Table 1.

There are two ways to terminate the Idle mode:—

Activation of any enabled interrupt will cause PCON.0 to be cleared by hardware terminating Idle mode. The interrupt is serviced, and following return from interrupt instruction RETI, the next instruction to be executed will be the one which follows the instruction that wrote a logic 1 to PCON.0. The flag bits GF0 and GF1 may be used to determine whether the interrupt was received during normal execution or during the Idle mode. For example, the instruction that writes the PCON.0 can also set or clear one or both flag bits. When Idle mode is terminated by an interrupt, the service routine can examine the status of the flags bits.

The second way of terminating the Idle mode is with a hardware reset. Since the oscillator is still running, the hardware reset is required to be active for two machine cycles (24 oscillator periods) to complete the reset operation.

Power-down mode

The instruction that sets PCON.1 is the last executed prior to going into the Power-down mode. Once in Power-down mode, the oscillator is stopped. Only the contents of the on-chip RAM are preserved. The Special Function Registers are not saved. A hardware reset is the only way of exiting the Power-down mode.

In the Power-down mode, V_{CC} may be reduced to minimize circuit power consumption. The voltage must not be reduced until the Power-down mode is entered, but must be restored before the hardware reset is applied which will free the oscillator. Reset should not be released until the oscillator has restarted and stabilized.

The status of the external pins during Power-down mode is shown in Table 1. If the Power-down mode is activated while in external program memory, the port data that is held in the Special Function Register P2 is restored to Port 2. If the data is a logic 1, the port pin is held HIGH during the Power-down mode by the strong pull-up transistor p1(see Fig. 5).

Table 1 Status of the external pins during Idle and Power-down modes.

mode	memory	ALE	$\overline{PSEN}$	Port 0	Port 1	Port 2	Port 3
Idle	internal	1	1	port data	port data	port data	port data
Idle	external	1	1	floating	port data	address	port data
Power-down	internal	0	0	port data	port data	port data	port data
Power-down	external	0	0	floating	port data	port data	port data

DEVELOPMENT DATA

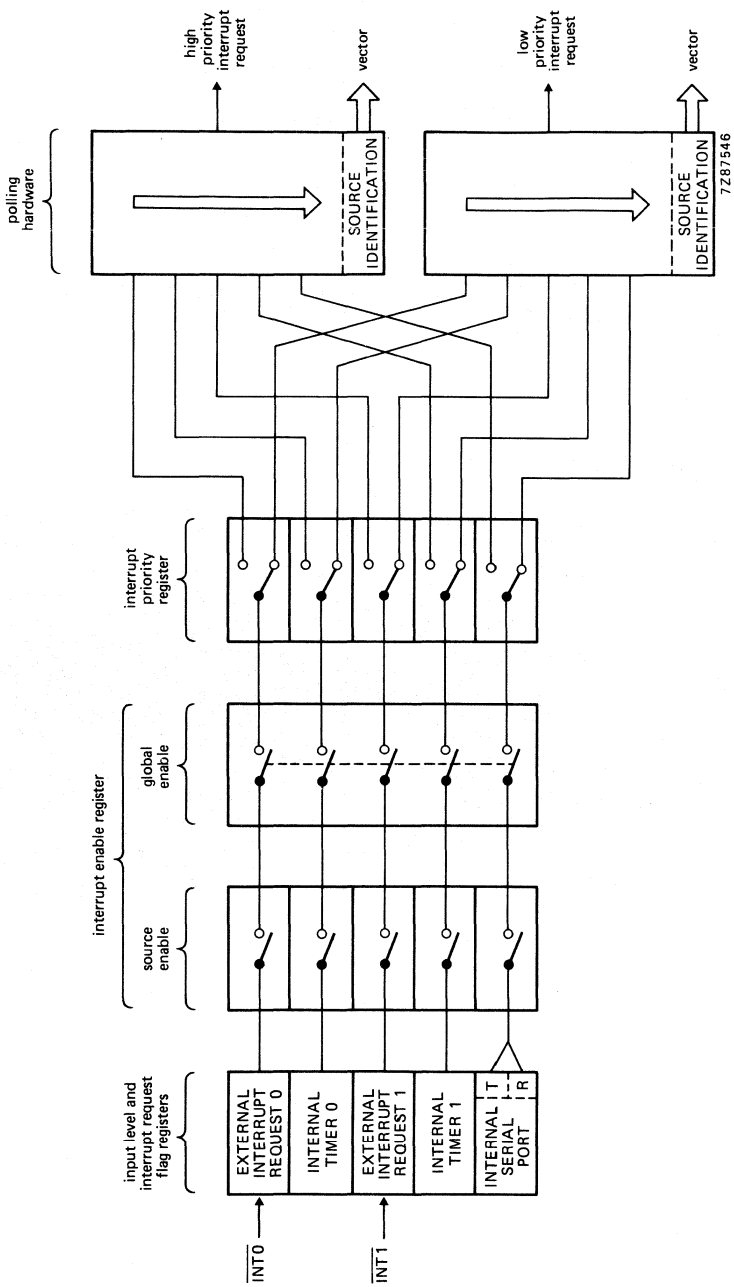


Fig. 7 Interrupt system.

Interrupt system (see Fig. 7)

External events and the real-time-driven on-chip peripherals require service by the CPU asynchronous to the execution of any particular section of code. To tie the asynchronous activities of these functions to normal program execution a multiple-source, two-priority-level, nested interrupt system is provided. Interrupt response latency is from 3 μs to 7 μs when using a 12 MHz crystal.

The PCB80C51BH acknowledges interrupt requests from five sources as follows:

- $\overline{\text{INT0}}$ and $\overline{\text{INT1}}$; externally via pins 12 and 13 respectively.
- Timer 0 and Timer 1; from the two internal counters.
- Serial Port; from the internal serial I/O port.

Each interrupt vectors to a separate location in program memory for its service program.

Each source can be individually enabled or disabled and can be programmed to a high or low priority level. Also all enabled sources can be globally disabled or enabled. Both external interrupts can be programmed to be level-activated or transition-activated and is active LOW to allow "wire-ORing" of several interrupt sources to the input pin.

Oscillator circuitry

The oscillator circuitry of the PCB80C51BH is a single-stage inverting amplifier in a Pierce oscillator configuration. The circuitry between XTAL 1 and XTAL 2 is basically an inverter biased to the transfer point. Either a crystal or ceramic resonator can be used as the feedback element to complete the oscillator circuitry. Both are operated in parallel resonance. XTAL 1 (pin 19) is the high gain amplifier input, and XTAL 2 (pin 18) is the output (see Fig. 8).

To drive the PCB80C51BH externally, XTAL 1 is driven from an external source and XTAL 2 left open-circuit (see Fig. 9).

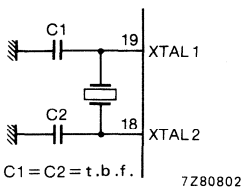


Fig. 8 PCB80C51BH oscillator circuit.

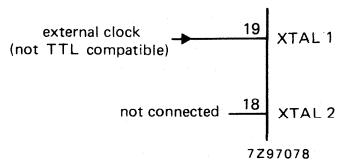


Fig. 9 Driving the PCB80C51BH from an external source.

Reset circuitry

The reset circuitry for the PCB80C51BH is connected to the reset pin, RST, as shown in Fig. 10. A Schmitt trigger is used at the input for noise rejection. The output of the Schmitt trigger is sampled by the reset circuitry every machine cycle.

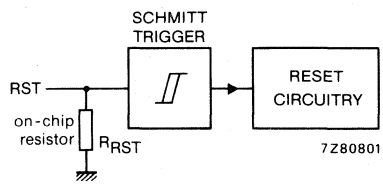


Fig. 10 Reset configuration at RST.

A reset is accomplished by holding the RST pin HIGH for at least two machine cycles (24 oscillator periods), while the oscillator is running. The CPU responds by executing an internal reset. It also configures the ALE and $\overline{\text{PSEN}}$ pins as inputs. (They are quasi-bidirectional.)

The internal reset is executed during the second cycle in which RST is HIGH and is repeated every cycle until RST goes LOW. It leaves the internal registers as follows:

Register	Content
PC	000H
ACC	00H
B	00H
PSW	00H
SP	07H
DPTR	0000H
P0 - P3	0FFH
IP	(XX000000)
IE	(0X000000)
TMOD	00H
TCON	00H
TH0	00H
TL0	00H
TH1	00H
TL1	00H
SCON	00H
SBUF	Intermediate
PCON	(0XXX0000)

The internal RAM is not affected by reset. When V_{CC} is turned on, the RAM content is indeterminate.

Power-on reset (see Fig. 11)

When V_{CC} is turned on, and provided its rise-time does not exceed 10 ms, an automatic reset can be obtained by connecting the RST pin to V_{CC} via a 10 μF capacitor. When the power is switched on, the current drawn by RST is the difference between V_{CC} and the capacitor voltage, and decreases from V_{CC} as the capacitor charges through the internal resistor (R_{RST}) to ground. The larger the capacitor, the more slowly V_{RST} decreases. V_{RST} must remain above the lower threshold of the Schmitt trigger long enough to effect a complete reset. The time required is the oscillator start-up time, plus 2 machine cycles.

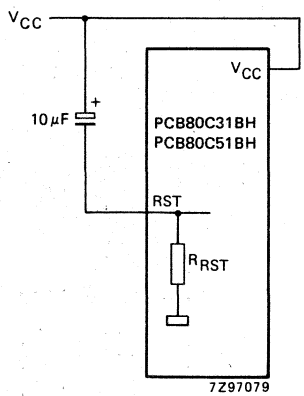


Fig. 11 Power-on reset.

To ensure a correct start-up at power-on, the voltage V_{CC} and RST must come up at the same time.

DEVELOPMENT DATA

INSTRUCTION SET

The PCB80C51BH uses a powerful instruction set to allow expansion of on-chip CPU peripherals and to optimize byte efficiency and execution speed. Reassigned opcodes add new high-power operations and permit new addressing modes to make old operations more orthogonal when compared to the 8048 family.

The instruction set consists of 49 single-byte, 45 two-byte and 17 three-byte instructions. When using a 16 MHz oscillator, 64 instructions execute in 750 ns and 45 instructions execute in 15 μs. Multiply and divide instructions execute in 3 μs.

Table 2 Instruction set description.

mnemonic		description	bytes/ cycles	opcode (hex.)
Arithmetic operation				
ADD	A,Rr	Add register to A	1 1	2*
ADD	A,direct	Add direct byte to A	2 1	25
ADD	A,@Ri	Add indirect RAM to A	1 1	26, 27
ADD	A,#data	Add immediate data to A	2 1	24
ADDC	A,Rr	Add register to A with carry flag	1 1	3*
ADDC	A,direct	Add direct byte to A with carry flag	2 1	35
ADDC	A,@Ri	Add indirect RAM to A with carry flag	1 1	36, 37
ADDC	A,#data	Add immediate data to A with carry flag	2 1	34
SUBB	A,Rr	Subtract register from A with borrow	1 1	9*
SUBB	A,direct	Subtract direct byte from A with borrow	2 1	95
SUBB	A,@Ri	Subtract indirect RAM from A with borrow	1 1	96, 97
SUBB	A,#data	Subtract immediate data from A with borrow	2 1	94
INC	A	Increment A	1 1	04
INC	Rr	Increment register	1 1	0*
INC	direct	Increment direct byte	2 1	05
INC	@Ri	Increment indirect RAM	1 1	06, 07
DEC	A	Decrement A	1 1	14
DEC	Rr	Decrement register	1 1	1*
DEC	direct	Decrement direct byte	2 1	15
DEC	@Ri	Decrement indirect RAM	1 1	16, 17
INC	DPTR	Increment data pointer	1 2	A3
MUL	AB	Multiply A & B	1 4	A4
DIV	AB	Divide A by B	1 4	84
DA	A	Decimal adjust A	1 1	D4

mnemonic		description	bytes/ cycles	opcode (hex.)
Logic operations				
ANL	A,Rr	AND register to A	1 1	5*
ANL	A,direct	AND direct byte to A	2 1	55
ANL	A,@Ri	AND indirect RAM to A	1 1	56, 57
ANL	A,#data	AND immediate data to A	2 1	54
ANL	direct,A	AND A to direct byte	2 1	52
ANL	direct,#data	AND immediate data to direct byte	3 2	53
ORL	A,Rr	OR register to A	1 1	4*
ORL	A,direct	OR direct byte to A	2 1	45
ORL	A,@Ri	OR indirect RAM to A	1 1	46, 47
ORL	A,#data	OR immediate data to A	2 1	44
ORL	direct,A	OR A to direct byte	2 1	42
ORL	direct,#data	OR immediate data to direct byte	3 2	43
XRL	A,Rr	Exclusive-OR register to A	1 1	6*
XRL	A,direct	Exclusive-OR direct byte to A	2 1	65
XRL	A,@Ri	Exclusive-OR indirect RAM to A	1 1	66, 67
XRL	A,#data	Exclusive-OR immediate data to A	2 1	64
XRL	direct, A	Exclusive-OR to direct byte	2 1	62
XRL	direct,#data	Exclusive-OR immediate data to direct byte	3 2	63
CLR	A	Clear A	1 1	E4
CPL	A	Complement A	1 1	F4
RL	A	Rotate A left	1 1	23
RLC	A	Rotate A left through the carry flag	1 1	33
RR	A	Rotate A right	1 1	03
RRC	A	Rotate A right through the carry flag	1 1	13
SWAP	A	Swap nibbles within A	1 1	C4

INSTRUCTION SET (continued)

mnemonic		description	bytes/ cycles	opcode (hex.)
Data transfer				
MOV	A,Rr	Move register to A	1 1	E*
MOV	A,direct **	Move direct byte to A	2 1	E5
MOV	A,@Ri	Move indirect RAM to A	1 1	E6, E7
MOV	A,#data	Move immediate data to A	2 1	74
MOV	Rr,A	Move A to register	1 1	F*
MOV	Rr,direct	Move direct byte to register	2 2	A*
MOV	Rr,#data	Move immediate data to register	2 1	7*
MOV	direct,A	Move A to direct byte	2 1	F5
MOV	direct,Rr	Move register to direct byte	2 2	8*
MOV	direct,direct	Move direct byte to direct	3 2	85
MOV	direct,@Ri	Move indirect RAM to direct byte	2 2	86, 87
MOV	direct,#data	Move immediate data to direct byte	3 2	75
MOV	@Ri,A	Move A to indirect RAM	1 1	F6, F7
MOV	@Ri,direct	Move direct byte to indirect RAM	2 2	A6, A7
MOV	@Ri,#data	Move immediate data to indirect RAM	2 1	76, 77
MOV	DPTR,#data16	Load data pointer with a 16-bit constant	3 2	90
MOVC	A,@A+DPTR	Move code byte relative to DPTR to A	1 2	93
MOVC	A,@A+PC	Move code byte relative to PC to A	1 2	83
MOVX	A,@Ri	Move external RAM (8-bit address) to A	1 2	E2, E3
MOVX	A,@DPTR	Move external RAM (16-bit address) to A	1 2	E0
MOVX	@Ri,A	Move A to external RAM (8-bit address)	1 2	F2, F3
MOVX	@DPTR,A	Move A to external RAM (16-bit address)	1 2	F0
PUSH	direct	Push direct byte onto stack	2 2	C0
POP	direct	Pop direct byte from stack	2 2	D0
XCH	A,Rr	Exchange register with A	1 1	C*
XCH	A,direct	Exchange direct byte with A	2 1	C5
XCH	A,@Ri	Exchange indirect RAM with A	1 1	C6, C7
XCHD	A,@Ri	Exchange LOW-order digit indirect RAM with A	1 1	D6, D7

** MOV A, ACC is not a valid instruction.

mnemonic		description	bytes/ cycles	opcode (hex.)
Boolean variable manipulation				
CLR	C	Clear carry flag	1 1	C3
CLR	bit	Clear direct bit	2 1	C2
SETB	C	Set carry flag	1 1	D3
SETB	bit	Set direct bit	2 1	D2
CPL	C	Complement carry flag	1 1	B3
CPL	bit	Complement direct bit	2 1	B2
ANL	C,bit	AND direct bit to carry flag	2 2	82
ANL	C,/bit	AND complement of direct bit to carry flag	2 2	B0
ORL	C,bit	OR direct bit to carry flag	2 2	72
ORL	C,/bit	OR complement of direct bit to carry flag	2 2	A0
MOV	C,bit	Move direct bit to carry flag	2 1	A2
MOV	bit,C	Move carry flag to direct bit	2 2	92
Program and machine control				
ACALL	addr11	Absolute subroutine call	2 2	●1addr
LCALL	addr16	Long subroutine call	3 2	12
RET		Return from subroutine	1 2	22
RETI		Return from interrupt	1 2	32
AJMP	addr11	Absolute jump	2 2	▲1addr
LJMP	addr16	Long jump	3 2	02
SJMP	rel	Short jump (relative address)	2 2	80
JMP	@A+DPTR	Jump indirect relative to the DPTR	1 2	73
JZ	rel	Jump if A is zero	2 2	60
JNZ	rel	Jump if A is not zero	2 2	70
JC	rel	Jump if carry flag is set	2 2	40
JNC	rel	Jump if no carry flag	2 2	50
JB	bit,rel	Jump if direct bit is set	3 2	20
JNB	bit,rel	Jump if direct bit is not set	3 2	30
JBC	bit,rel	Jump if direct bit is set and clear bit	3 2	10
CJNE	A,direct,rel	Compare direct to A and jump if not equal	3 2	B5
CJNE	A,#data,rel	Compare immediate to A and jump if not equal	3 2	B4
CJNE	Rr,#data,rel	Compare immed. to reg. and jump if not equal	3 2	B*
CJNE	@Ri,#data,rel	Compare immed. to ind. and jump if not equal	3 2	B6, B7
DJNZ	Rr,rel	Decrement register and jump if not zero	2 2	D*
DJNZ	direct,rel	Decrement direct and jump if not zero	3 2	D5
NOP		No operation	1 1	00

Notes to Table 2

Data addressing modes

Rr	Working register R0-R7.
direct	128 internal RAM locations and any special function register (SFR).
@Ri	Indirect internal RAM location addressed by register R0 or R1 of the actual register bank.
#data	8-bit constant included in instruction.
#data16	16-bit constant included as bytes 2 and 3 of instruction.
bit	direct addressed bit in internal RAM or SFR.
addr16	16-bit destination address. Used by LCALL and LJMP. The branch will be anywhere within the 64 K-byte program memory address space.
addr11	11-bit destination address. Used by ACALL and AJMP. The branch will be within the same 2 K-byte page of program memory as the first byte of the following instruction.
rel	Signed (two's complement) 8-bit offset byte. Used by SJMP and all conditional jumps. Range is −128 to + 127 bytes relative to first byte of the following instruction.

Hexadecimal opcode cross-reference to Table 3

★ : 8, 9, A, B, C, D, E, F.

● : 11, 31, 51, 71, 91, B1, D1, F1.

▲ : 01, 21, 41, 61, 81, A1, C1, E1.

Table 3 Instruction map

first hexadecimal chacter of opcode ↓ — second hexadecimal character of opcode →

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NOP	AJMP addr11	LJMP addr16	RR A	INC A	INC dir	INC @Ri 0	1	INC Rr 0	1	2	3	4	5	6	7
1	JBC bit,rel	ACALL addr11	LCALL addr16	RRC A	DEC A	DEC dir	DEC @Ri 0	1	DEC Rr 0	1	2	3	4	5	6	7
2	JB bit,rel	AJMP addr11	RET	RL A	ADD A,#data	ADD A,dir	ADD A,@Ri 0	1	ADD A,Rr 0	1	2	3	4	5	6	7
3	JNB bit,rel	ACALL addr11	RETI	RLC A	ADDC A,#data	ADDC A,dir	ADDC A,@Ri 0	1	ADDC A,Rr 0	1	2	3	4	5	6	7
4	JC rel	AJMP addr11	ORL dir,A	ORL dir,#data	ORL A,#data	ORL A,dir	ORL A,@Ri 0	1	ORL A,Rr 0	1	2	3	4	5	6	7
5	JNC rel	ACALL addr11	ANL dir,A	ANL dir,#data	ANL A,#data	ANL A,dir	ANL A,@Ri 0	1	ANL A,Rr 0	1	2	3	4	5	6	7
6	JZ rel	AJMP addr11	XRL dir,A	XRL dir,#data	XRL A,#data	XRL A,dir	XRL A,@Ri 0	1	XRL A,Rr 0	1	2	3	4	5	6	7
7	JNZ rel	ACALL addr11	ORL C,bit	JMP @A+DPTR	MOV A,#data	MOV dir,#data	MOV @Ri,#data 0	1	MOV Rr,#data 0	1	2	3	4	5	6	7
8	SJMP rel	AJMP addr11	ANL C,bit	MOVC A,@A+PC	DIV AB	MOV dir,dir	MOV dir,@Ri 0	1	MOV dir,Rr 0	1	2	3	4	5	6	7
9	MOV DPTR, #data	ACALL addr11	MOV bit,C	MOVC A,@A+DPTR	SUBB A,#data	SUBB A,dir	SUBB A,@Ri 0	1	SUBB A,Rr 0	1	2	3	4	5	6	7
A	ORL C,/bit	AJMP addr11	MOV C,bit	INC DPTR	MUL AB		MOV @Ri,dir 0	1	MOV Rr,dir 0	1	2	3	4	5	6	7
B	ANL C,/bit	ACALL addr11	CPL bit	CPL C	CJNE A, #data,rel	CJNE A,dir,rel	CJNE @Ri,#data,rel 0	1	CJNE Rr,#data,rel 0	1	2	3	4	5	6	7
C	PUSH dir	AJMP addr11	CLR bit	CLR C	SWAP A	XCH A,dir	XCH A,@Ri 0	1	XCH A,Rr 0	1	2	3	4	5	6	7
D	POP dir	ACALL addr11	SETB bit	SETB C	DA A	DJNZ dir,rel	XCHD A,@Ri 0	1	DJNZ Rr,rel 0	1	2	3	4	5	6	7
E	MOVX A,@DPTR	AJMP addr11	MOVX A,@Ri 0	1	CLR A	MOV * A,dir	MOV A,@Ri 0	1	MOV A,Rr 0	1	2	3	4	5	6	7
F	MOVX @DPTR,A	ACALL addr11	MOVX @Ri,A 0	1	CPL A	MOV dir,A	MOV @Ri,A 0	1	MOV Rr,A 0	1	2	3	4	5	6	7

RATINGS

Limiting values in accordance with the Absolute Maximum System (IEC 134)

Input voltage on any pin with respect to ground (V_{SS})	V_I	−0,5 to +6,5	V
Input, output current	$\pm I_I$, I_O	max. 5	mA
Total power dissipation	P_{tot}	max. 1	W
Storage temperature range	T_{stg}	−65 to + 150	°C
Operating ambient temperature range	T_{amb}	0 to + 70	°C

D.C. CHARACTERISTICS

V_{CC} = 5 V (±10%); V_{SS} = 0 V; T_{amb} = −40 to + 85 °C, 0 to + 70 °C; all voltages with respect to V_{SS} unless otherwise specified

parameter	symbol	min.	max.	unit	conditions
Supply voltage	V_{CC}	4,5	5,5	V	
Supply current					
operating (note 1)	I_{CC}	–	18*	mA	f_{CLK} = 12 MHz PCF80C51BH
	I_{CC}	–	23*	mA	f_{CLK} = 16 MHz PCB80C51BH
idle mode (note 2)	I_{CC}	–	10*	mA	f_{CLK} = 12 MHz
Power-down current	I_{PD}	–	100*	µA	V_{CC} = 2 V (note 3)
Inputs					
LOW level input voltage (except $\overline{EA}$)	V_{IL}	−0,5	$0{,}2V_{CC}-0{,}1$	V	
LOW level input voltage ($\overline{EA}$)	V_{IL1}	−0,5	$0{,}2V_{CC}-0{,}3$	V	
HIGH level input voltage (except XTAL 1, RST	V_{IH}	$0{,}2V_{CC}+0{,}9$	V_{CC} +0,5	V	
HIGH level input voltage (XTAL 1, RST)	V_{IH1}	$0{,}7V_{CC}$	V_{CC} +0,5	V	
Input current logic 0 (Ports 1, 2 and 3)	$-I_{IL}$	–	50	µA	V_I = 0,45 V
Input current logic 1 to 0 transition (Ports 1, 2 and 3)	$-I_{TL}$	–	650	µA	V_I = 2 V
Input leakage current (Port 0, $\overline{EA}$)	$\pm I_{LI}$	–	10	µA	$0{,}45\ V < V_I < V_{CC}$

* Preliminary value. Table showing I_{CC} as function of frequency will be available at a later date.

parameter	symbol	min.	max.	unit	conditions
Outputs					
LOW level output voltage (note 4) (Ports 1, 2 and 3)	V_{OL}	–	0,45	V	I_{OL} = 1,6 mA
LOW level output voltage (note 4) (Port 0, ALE, $\overline{PSEN}$)	V_{OL1}	–	0,45	V	I_{OL} = 3,2 mA
HIGH level output voltage (Ports 1, 2 and 3)	V_{OH}	2,4	–	V	$-I_{OH}$ = 60 μA; V_{CC} = 5 V ± 10%
		$0{,}75V_{CC}$	–	V	$-I_{OH}$ = 25 μA
		$0{,}9V_{CC}$	–	V	$-I_{OH}$ = 10 μA
HIGH level output voltage (note 5) (Port 0 in external Bus mode, ALE, $\overline{PSEN}$)	V_{OH1}	2,4	–	V	$-I_{OH}$ = 400 μA; V_{CC} = 5 V ± 10%
		$0{,}75V_{CC}$	–	V	$-I_{OH}$ = 150 μA
		$0{,}9V_{CC}$	–	V	$-I_{OH}$ = 40 μA
RST pull-down resistor	R_{RST}	50	150	kΩ	
I/O pin capacitance	$C_{I/O}$	–	10	pF	test freq. = 1 MHz; T_{amb} = 25 °C

DEVELOPMENT DATA

Notes to the d.c. characteristics

1. The operating supply current is measured with all output pins disconnected; XTAL 1 driven with $t_r = t_f$ = 10 ns, $V_{IL} = V_{SS}$ + 0,5 V, $V_{IH} = V_{CC}$ −0,5 V; XTAL 2 not connected; $\overline{EA}$ = RST = Port 0 = V_{CC}.
2. The idle mode supply current is measured with all output pins disconnected; XTAL 1 driven with $t_r = t_f$ = 10 ns, $V_{IL} = V_{SS}$ + 0,5 V, $V_{IH} = V_{CC}$ −0,5 V; XTAL 2 not connected; $\overline{EA}$ = Port 0 = V_{CC}; RST = V_{SS}.
3. The power-down current is measured with all output pins disconnected; XTAL 2 not connected; $\overline{EA}$ = Port 0 = V_{CC}; RST = V_{SS}.
4. Capacitive loading on Port 0 and Port 2 may cause spurious noise pulses to be superimposed on the LOW level output voltage of ALE, Port 1 and Port 3. The noise is due to external Bus capacitance discharging into the Port 0 and Port 2 pins when these pins make a 1-to-0 transition during Bus operations. In the most adverse condition (capacitive loading > 100 pF) the noise pulse on ALE line may exceed 0,8 V. In this event it may be required to qualify ALE with a Schmitt trigger, or use an address latch with a Schmitt trigger STROBE input.
5. Capacitive loading on Port 0 and Port 2 may cause the HIGH level output voltage on ALE and $\overline{PSEN}$ to momentarily fall below the $0{,}9V_{CC}$ specification when the address bits are stabilizing.

A.C. CHARACTERISTICS

V_{CC} = 5 V ± 10%; V_{SS} = 0 V; T_{amb} = 0 to + 70 °C; C_L = 100 pF (Port 0, ALE and $\overline{PSEN}$); C_L = 80 pF (all other outputs); unless otherwise specified (see waveforms Figs 14, 15 and 16)

parameter	symbol	16 MHz		12 MHz		variable clock		unit
		min.	max.	min.	max.	min.	max.	
Program memory								
ALE pulse duration	t_{LL}	85	–	127	–	$2t_{CK}-40$	–	ns
Address set-up time to ALE	t_{AL}	23	–	43	–	$t_{CK}-40$	–	ns
Address hold time after ALE	t_{LA}	28	–	48	–	$t_{CK}-35$	–	ns
Time from ALE to valid instruction input	t_{LIV}	–	150	–	233	–	$4t_{CK}-100$	ns
Time from ALE to control pulse $\overline{PSEN}$	t_{LC}	38	–	58	–	$t_{CK}-25$	–	ns
Control pulse duration $\overline{PSEN}$	t_{CC}	153	–	215	–	$3t_{CK}-35$	–	ns
Time from $\overline{PSEN}$ to valid instruction input	t_{CIV}	–	63	–	125	–	$3t_{CK}-125$	ns
Input instruction hold time after $\overline{PSEN}$	t_{CI}	0	–	0	–	0	–	ns
Input instruction float delay after $\overline{PSEN}$*	t_{CIF}	–	43	–	63	–	$t_{CK}-20$	ns
Address to valid instruction input	t_{AIV}	–	198	–	302	–	$5t_{CK}-115$	ns
Address float time to $\overline{PSEN}$	t_{AFC}	0	–	0	–	0	–	ns

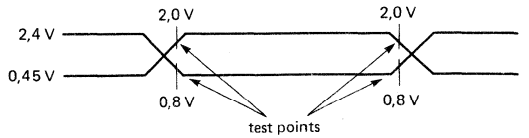
* Interfacing the PCB80C51BH to devices with float times up to 75 ns is permitted. This limited bus contention will not cause damage to port 0 drivers.

parameter	symbol	16 MHz		12 MHz		variable clock		unit
		min.	max.	min.	max.	min.	max.	
External data memory								
$\overline{RD}$ pulse duration	t_{RR}	275	–	400	–	$6t_{CK}-100$	–	ns
$\overline{WR}$ pulse duration	t_{WW}	275	–	400	–	$6t_{CK}-100$	–	ns
Address hold time after ALE	t_{LA}	28	–	48	–	$t_{CK}-35$	–	ns
$\overline{RD}$ to valid data input	t_{RD}	–	148	–	250	–	$5t_{CK}-165$	ns
Data hold time after $\overline{RD}$	t_{DR}	0	–	0	–	0	–	ns
Data float delay after $\overline{RD}$	t_{DFR}	–	55	–	97	–	$2t_{CK}-70$	ns
Time from ALE to valid data input	t_{LD}	–	350	–	517	–	$8t_{CK}-150$	ns
Address to valid data input	t_{AD}	–	398	–	585	–	$9t_{CK}-165$	ns
Time from ALE to $\overline{RD}$ or $\overline{WR}$	t_{LW}	138	238	200	300	$3t_{CK}-50$	$3t_{CK}+50$	ns
Time from address to $\overline{RD}$ or $\overline{WR}$	t_{AW}	120	–	203	–	$4t_{CK}-130$	–	ns
Time from $\overline{RD}$ or $\overline{WR}$ HIGH to ALE HIGH	t_{WHLH}	23	103	43	123	$t_{CK}-40$	$t_{CK}+40$	ns
Data valid to $\overline{WR}$ transition	t_{DWX}	3	–	23	–	$t_{CK}-60$	–	ns
Data set-up time before $\overline{WR}$	t_{DW}	288	–	433	–	$7t_{CK}-150$	–	ns
Data hold time after $\overline{WR}$	t_{WD}	13	–	33	–	$t_{CK}-50$	–	ns
Address float delay after $\overline{RD}$	t_{AFR}	–	0	–	0	–	0	ns

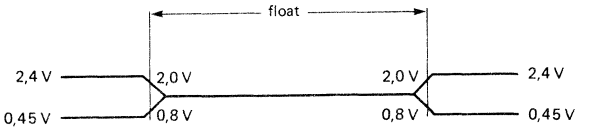
Where:

$1/t_{CK}$ = 1,2 to 16 MHz (see Fig. 13 and Table 4).

A.C. CHARACTERISTICS (continued)



(a)



(b)

A.C. testing inputs are driven at 2,4 V for a logic 1 and 0,45 V for a logic 0.
Timing measurements are taken at 2,0 V for a logic 1 and 0,8 V for logic 0.
The float state is defined as the point at which a Port 0 pin sinks 3,2 mA or sources 400 μA at the voltage test levels.

Fig. 12 A.C. testing input, output waveform (a) and float waveform (b).

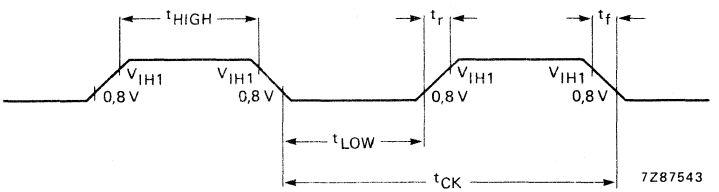


Fig. 13 External clock drive XTAL 1 (see Table 4).

Table 4 External clock drive XTAL 1 (see Fig. 13)

parameter	symbol	variable clock (f = 1,2 to 16 MHz)*		unit
		min.	max.	
oscillator clock period	t_{CK}	63	833,3	ns
HIGH time	t_{HIGH}	20	$t_{CK}-t_{LOW}$	ns
LOW time	t_{LOW}	20	$t_{CK}-t_{HIGH}$	ns
rise time	t_r	–	20	ns
fall time	t_f	–	20	ns

* PCF80C51BH: 1,2 – 12 MHz only.

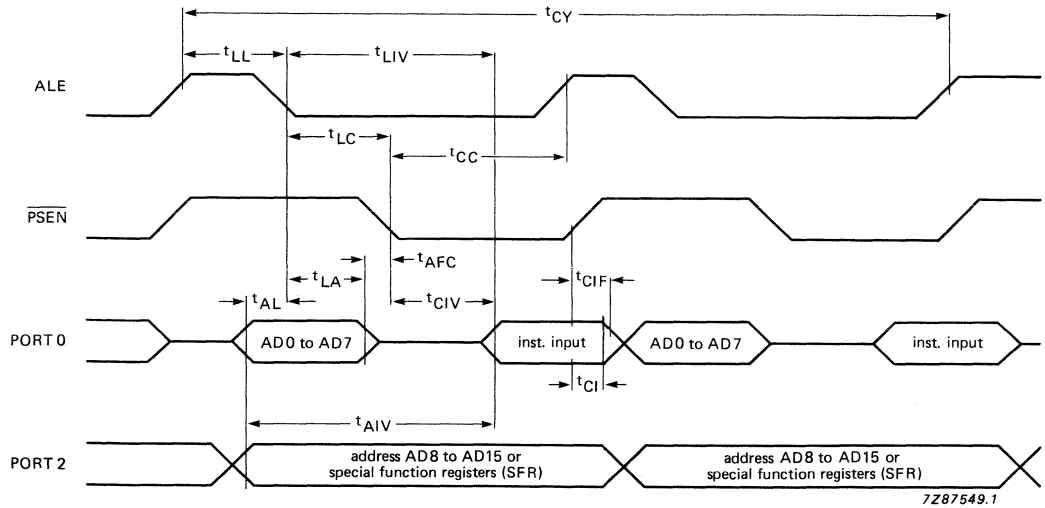


Fig. 14 Read from program memory.

DEVELOPMENT DATA

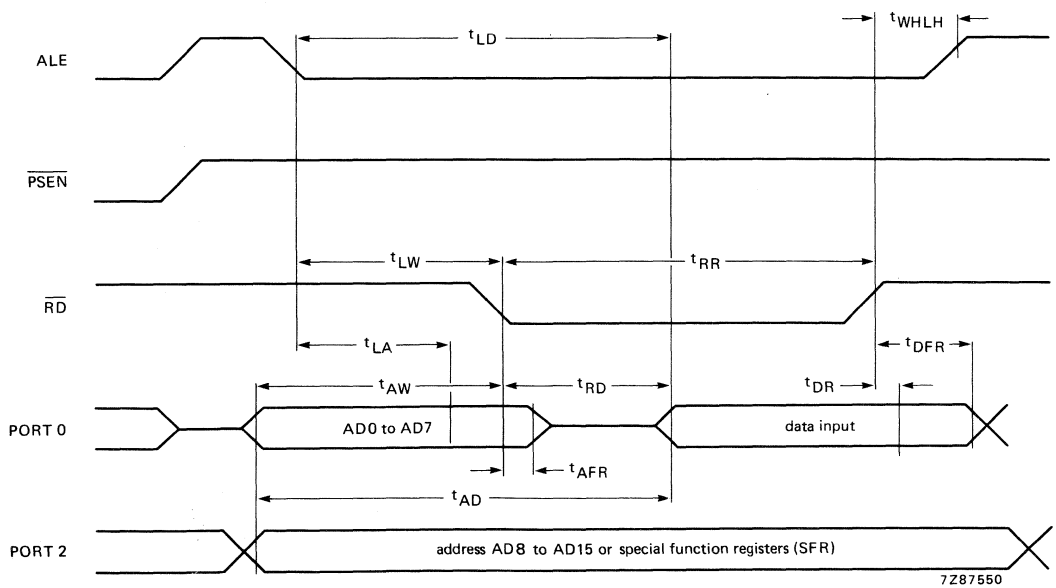


Fig. 15 Read from data memory.

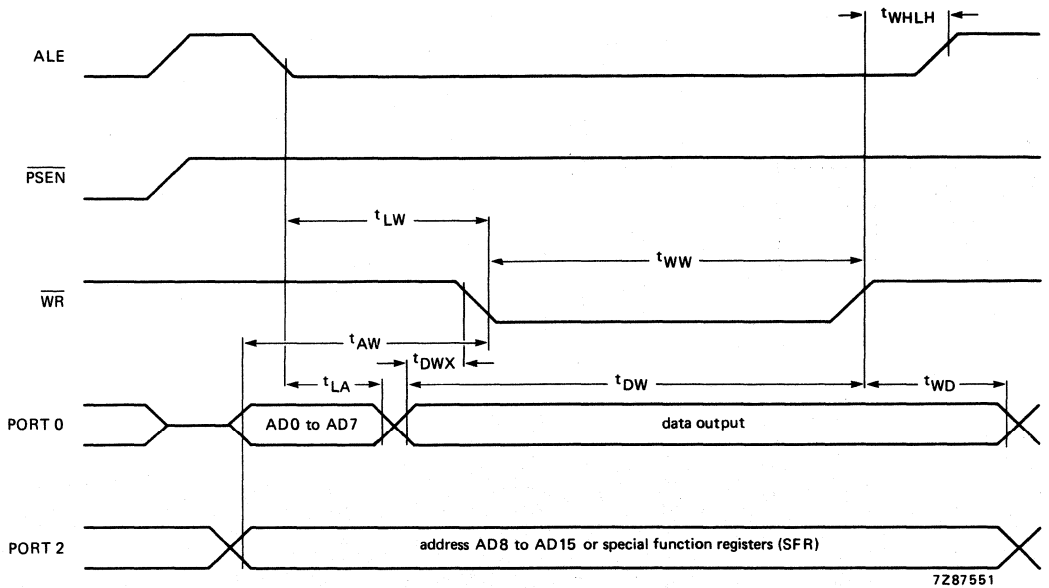


Fig. 16 Write to data memory.

DEVELOPMENT DATA

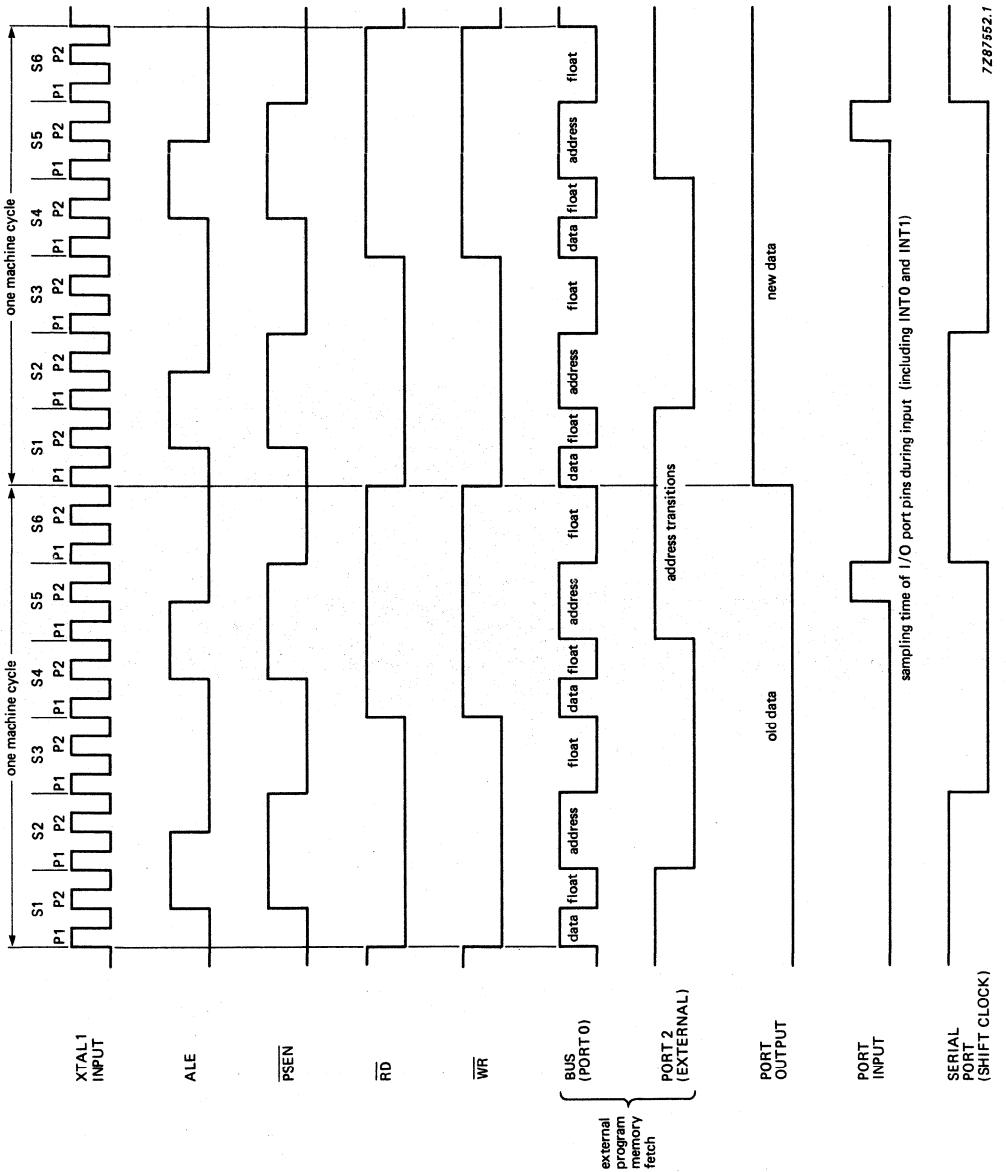


Fig. 17 Instruction cycle timing.

SINGLE-CHIP 8-BIT CMOS MICROCONTROLLER

DESCRIPTION

The PCB80CXX family of single-chip 8-bit CMOS microcontrollers consists of:

- The PCB80C49 with resident mask programmed 2K x 8 ROM, 128 x 8 RAM.
- The PCB80C39 without resident program memory for use with external EPROM/ROM,128 x 8 RAM.

All versions are pin and function compatible to their NMOS counter parts but with additional features and high performance.

The PCB80CXX family are desgined to be both efficient control and arithmetic processors.
The instruction set allows the user to directly set and reset individual I/O lines, and to test individual bits within the accumulator. A large variety of branch and table look-up instructions enable efficient implementation of standard logic functions. Code efficiency is high; over 70% of the instructions are single byte; all others are two byte.
An on-chip 8-bit counter is provided, which can count either machine cycles (÷ 32) or external events. The counter can be programmed to cause an interrupt to the processor.
Program and data memories can be expanded using standard devices. Input/output capabilities can be expanded using standard devices.
The family has low power consumption and in addition a power down mode is provided.
For further detailed information see users manual 'single-chip 8-bit microcontrollers'.

FEATURES

- 8-bit CPU, ROM, RAM, I/O in a single 40-pin package
- PCB80C49: 2K x 8 ROM, 128 x 8 RAM
- PCB80C39: 128 x 8 RAM
- Internal counter/timer
- Internal oscillator, clock driver
- Single-level interrupts: external and counter/timer
- 17 internal registers: accumulator, 16 addressable registers
- Over 90 instructions: 70% single byte
- All instructions: 1 or 2 cycles
- Easily expandable memory and I/O
- TTL compatible inputs and outputs
- Single 5 V supply
- Wide frequency operation range
- Low current consumption
- Also available with extended temperature range; −40 °C to + 85 °C

APPLICATIONS

- Peripheral interfaces and controllers
- Test and measurement instruments
- Sequencers
- Audio/video systems
- Environmental control systems
- Modems and data enciphering

PACKAGES

PCB/F80C39/C49P: 40-lead DIL; plastic (SOT-129).
PCB/F80C39/C49WP: 44-lead plastic leaded chip carrier (PLCC); SOT-187A.

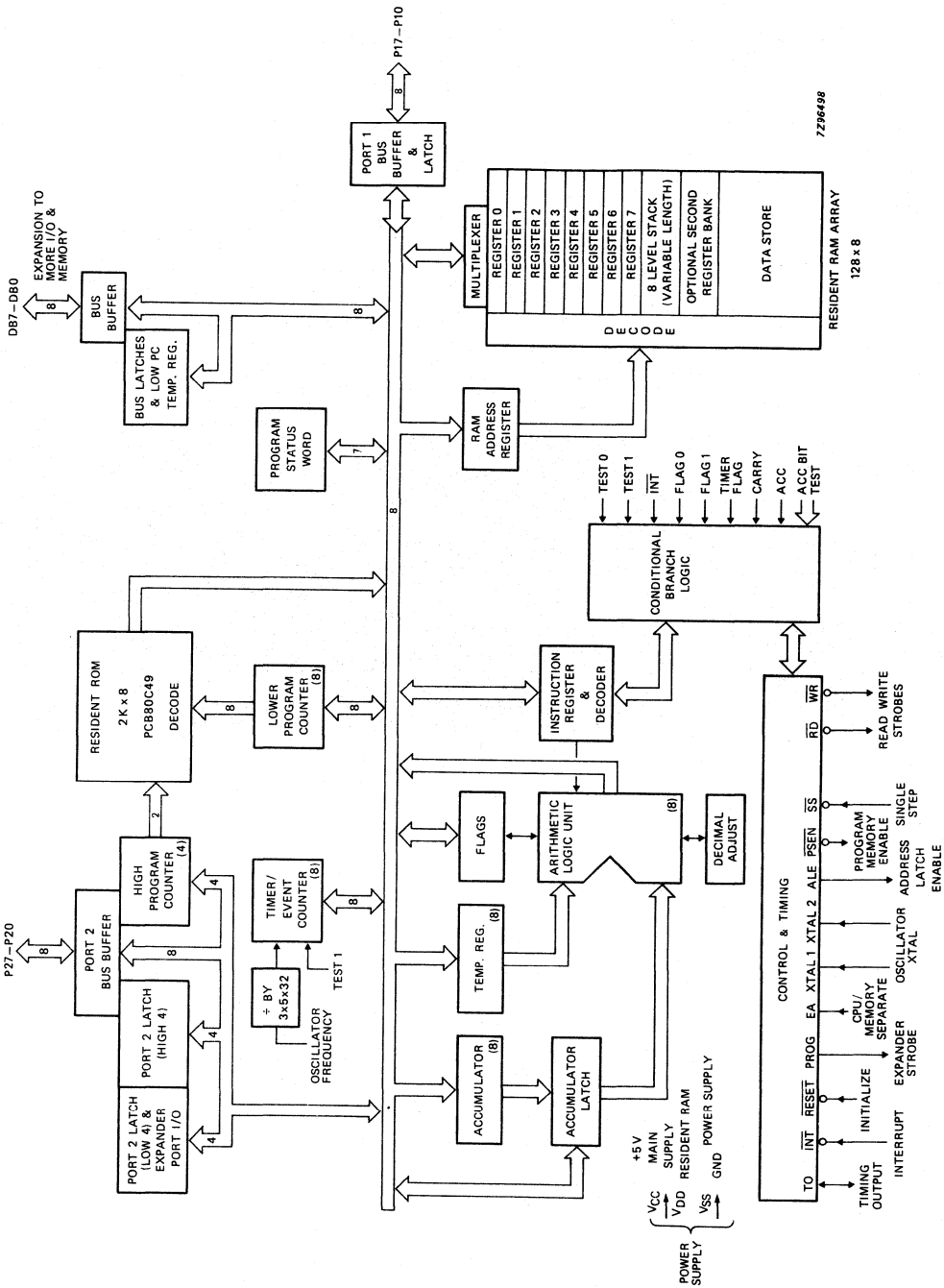


Fig. 1 Block diagram.

DEVELOPMENT DATA

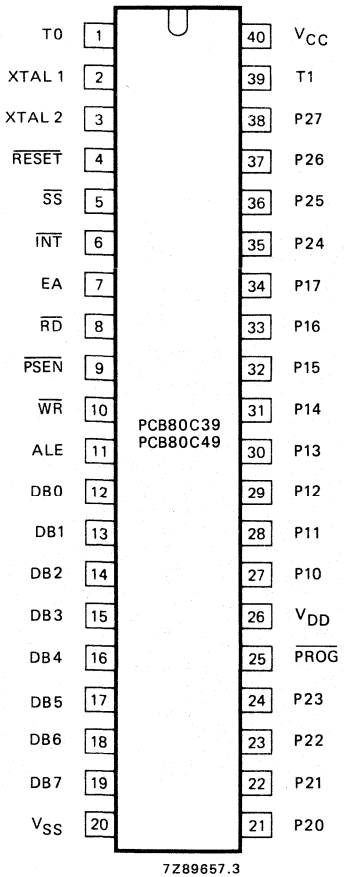


Fig. 2(a) Pinning diagram, DIL; for pin designation see next page.

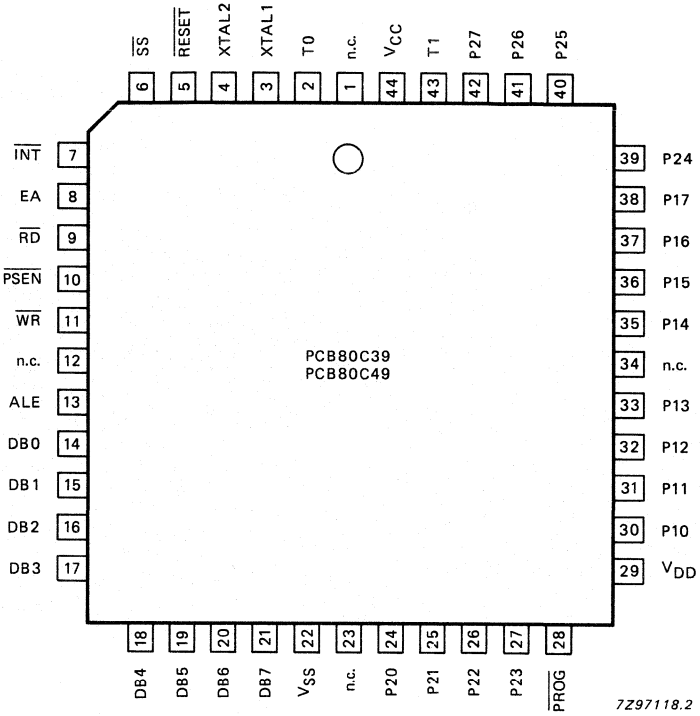


Fig. 2(b) Pinning diagram, PLCC; for pin designation see next page.

Where: n.c. = not connected.

PIN DESIGNATION

designation	pin no.	function
DB0–DB7	12–19	**BUS.** Bidirectional I/O port that can be read or written using the $\overline{RD}$ and $\overline{WR}$ strobes. This port can also be statically latched. Contains the 8 lower order address bits during external memory access and receives the addressed instruction under control of $\overline{PSEN}$. $\overline{PSEN}$, ALE, $\overline{RD}$ and $\overline{WR}$ determine whether the access is an instruction fetch or a read/write access to external RAM.
P10–P17	27–34	**Port 1.** 8-bit quasi-bidirectional I/O port (note 1).
P20–P27	21–24, 35–38	**Port 2.** 8-bit quasi-bidirectional I/O port (note 1). P20–P23 contain the 4 higher order address bits during an access of external program memory.
$\overline{PROG}$	25	**Output strobe** (active LOW) for I/O expander.
T0	1	**Input pin** sensed using the JT0 and JNT0 instructions. **Clock output pin** when designated as such by the ENT0 CLK instruction.
T1	39	**Input pin** sensed using the JT1 and JNT1 instructions. Can be designated as the timer/counter input by the STRT CNT instruction.
$\overline{INT}$	6	**Interrupt input pin.** When LOW causes an interrupt in the current program if external interrupt is enabled. Can also be used as an input, testable using the JNI instruction. Interrupt is disabled during and after a RESET.
$\overline{RESET}$	4	**Reset input pin** used to initialize the microcontroller. Active LOW. During program verification the address is latched by a '0' to '1' transition on $\overline{RESET}$ and the data at the addressed location is output on BUS (note 2).
ALE	11	**Address latch enable.** Occurs once each machine cycle and is useful for timing and sampling. During external program or data memory access, ALE is used to strobe the address information multiplexed on the DB0 to DB7 outputs.
$\overline{RD}$	8	**Read BUS.** Active LOW strobe used to gate data onto BUS lines when reading from an external source.
$\overline{WR}$	10	**Write BUS.** Active LOW strobe used to write data from BUS lines to an external designation.
EA	7	**External access input.** When HIGH, forces instruction fetch from external memory.
$\overline{PSEN}$	9	**Program store enable.** Active LOW strobe that occurs only during a fetch from external program memory.
$\overline{SS}$	5	**Single step input.** Active LOW which is used with ALE to cause the microcontroller to execute a single instruction.
V_{DD}	26	**RAM power supply,** + 5 V during normal operation and power-down mode.

designation	pin no.	function
XTAL1	2	**One side of crystal** (or inductor) input for internal oscillator. Can also be used as an input for an external timing source (note 2).
XTAL2	3	**Other side of crystal.**
V_{SS}	20	**Ground.**
V_{CC}	40	**Main power supply**, + 5 V during normal operation.

Notes

1. Each port line can be designated as an input or an output. A line is designated as an input by first writing a logic 1 to the line. $\overline{\text{RESET}}$ sets all lines to logic 1.
2. Non-standard TTL V_{IH}.

FUNCTIONAL DESCRIPTION

Program memory (see Fig. 3)

The resident program memory is:
2048 byte ROM
The PCB80C39 has no resident program memory.
The total addressing capability is 4096 bytes.
The program memory address space is divided into two 2048-bytes banks MB0 and MB1.
The program memory is also divided into pages of 256 bytes for conditional branches.

There are three locations in program memory of special importance. These locations contain the first instruction to be executed after one of three events.
The three locations and their contents are:
location 0 — activation, then deactivation of the $\overline{\text{RESET}}$ line,
location 3 — activation of the $\overline{\text{INT}}$ line when the external interrupt is enabled,
location 7 — an overflow of the timer/counter if the T/C interrupt is enabled.

Data memory (see Fig. 4)

The resident data memory is: 128 byte RAM.

All locations are indirectly addressable by either of two RAM pointer registers at locations 0 and 1.
The first eight locations of RAM (0 to 7) are designated as working registers and are directly addressable by several instructions. By selecting register bank 1, RAM locations 24 to 31 become the working registers, replacing those in register bank 0 (0 to 7).
RAM locations 8 to 23 are designated as the stack. Two locations (bytes) are used per CALL, allowing up to eight levels of subroutine nesting.
If additional RAM is required, up to 256 bytes may be added and addressed directly using the MOVX instructions. If more RAM is required, an I/O port can be used to select one (256 byte) bank of external memory at a time.

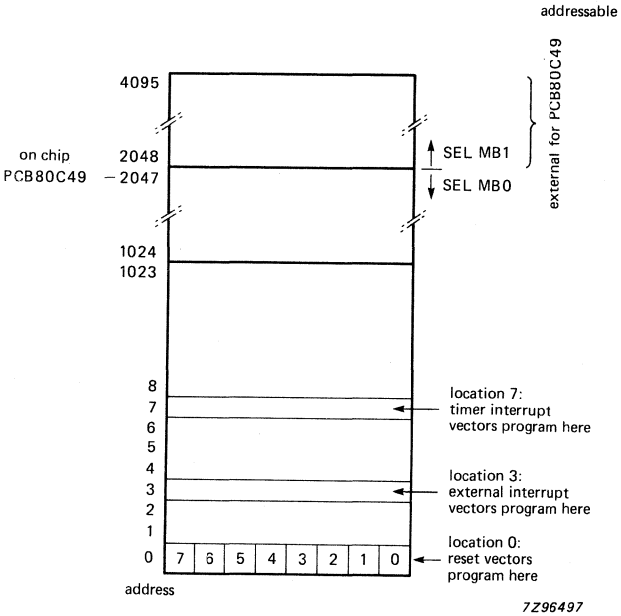


Fig. 3 Program memory map.

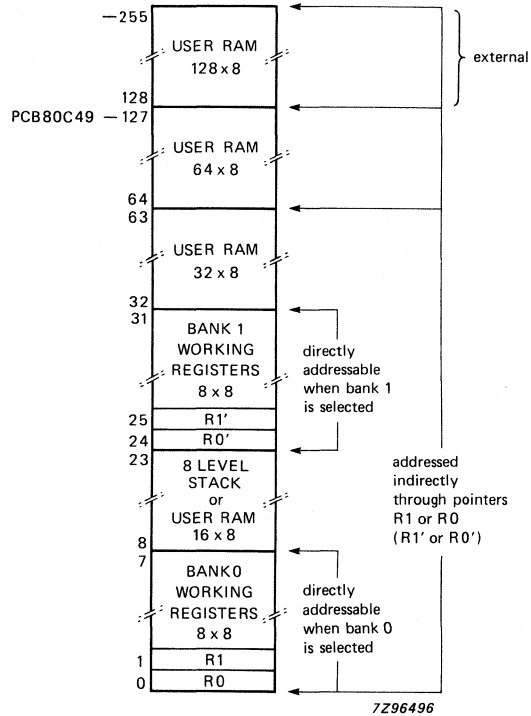


Fig. 4 Data memory map. In addition R0 or R1 (R0′ or R1′) may be used to address 256 words of an external RAM.

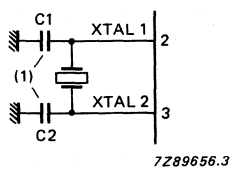
Program counter and stack

The program counter (PC) is a 12-bit counter/register that points to the location from which the next instruction is to be fetched. When EA is '0', the PC addresses an internal program memory. At the boundary of the internal program memory an automatic switch over to external memory is made. When EA is '1', all the program is fetched from external ROM/EPROM. The total address space is 4K bytes. An interrupt or CALL to a subroutine causes the contents of the program counter to be stored in one of the 8 register pairs of the program counter stack. The pair to be used is determined by a 3-bit stack pointer which is part of the program status word (PSW). Data RAM locations 8 to 23 are available as stack registers and are used to store the program counter and 4 bits of PSW. The stack pointer, when initialized to 000B, points to RAM locations 8 and 9. The first subroutine jump or interrupt results in the program counter contents being transferred to locations 8 and 9 of the RAM array. The stack pointer is then incremented by one to point to locations 10 and 11 in anticipation of another CALL. Nesting of subroutines within subroutines can continue up to 8 times without overflowing the stack. If overflow does occur the deepest address stored (location 8 and 9) will be overwritten and lost since the stack pointer overflows from 111 to 000. It also underflows from 000 to 111.
The end of a subroutine, which is signalled by a return instruction (RET or RETR), causes the stack pointer to be decremented and the contents of the resulting register pair to be transferred to the program counter.

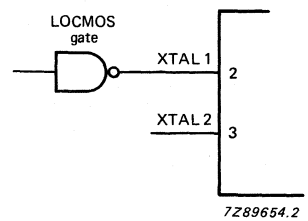
Oscillator and clock

The PCB80C49 contains its own internal oscillator and clock driver.
A crystal, inductor or external pulse generator determines the oscillator frequency (see Figs 5, 6 and 7). The output of the oscillator is divided-by-three and is available at T0 (pin 1) by executing the ENT0 CLK instruction. This CLK signal is divided-by-five to define a machine (instruction) cycle. It is available at ALE (pin 11).

DEVELOPMENT DATA



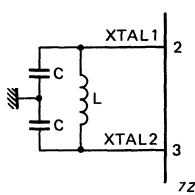
For quartz crystal
1 to 11 MHz: C1 = C2 = 15 to 25 pF
For ceramic resonators
1 to 11 MHz: C1 = C2 = 30^{+5}_{-10} pF



(1) Including crystal-socket stray capacitance.

Fig. 5 Crystal oscillator mode. Typical values are given. Crystal serial impedance should be $< 75\ \Omega$ at 6 MHz and $< 180\ \Omega$ at 3,6 MHz.

Fig. 6 Driving from an external source. Test conditions at XTAL1; Minimum HIGH ($> 0{,}7$ of V_{CC}) and LOW ($< 0{,}13$ of V_{CC}) times, should be at least 45% of a clock period.



L (μH)	C (pF)	f_{nom} (MHz)
45	20	5,2
120	20	3,2

Fig. 7 LC oscillator mode.

FUNCTIONAL DESCRIPTION (continued)

Timer/event counter

An internal counter is available which can count either external events or machine cycles (÷ 32). The machine cycles are divided-by-32 before they are applied to the input of the 8-bit counter. External events are applied directly to the input of the counter. The maximum frequency that can be counted is one third of the machine cycle frequency. The minimum positive duty cycle that can be detected is 0,2 times the cycle period. The counter is under program control and can be made to generate an interrupt to the processor when it overflows.

Interrupt

An interrupt may be generated by either an external input ($\overline{INT}$, pin 6) or the overflow of the internal timer/event counter, when enabled. In either case, the processor completes execution of the present instruction and then does a CALL to the interrupt service routine, After service, a RETR instruction restores the machine to the state it was prior to the interrupt. The external interrupt has priority over the internal interrupt.

Input/output

The PCB80CXX family has 27 I/O lines. These lines are arranged as three 8-line ports, which serve individually as either inputs, outputs or together as bidirectional ports, plus 3 'test' inputs which can alter program sequences when tested by conditional jump instructions.

Ports 1 and 2

Ports 1 and 2 are each 8-bits wide and have identical characteristics. Data written to these ports is statically latched and remains unchanged until rewritten. As input ports these lines are non-latching, e.g., inputs must be present until read by an input instruction. Inputs are fully TTL compatible and outputs will drive one standard TTL load.

The lines of ports 1 and 2 are called quasi-bidirectional because of a special output circuit structure which allows each line to serve as an input, an output, or both even though outputs are statically latched. The circuit configuration is shown in Fig. 8. Each line has a unique high-impedance pull-up transistor TR3, this is turned on when the line is pulled above 2 V by an external source or by writing a logic 1 to the port. This pull-up is sufficient to provide the source current for a TTL HIGH level, yet can be pulled LOW by a standard TTL gate, thus allowing the same pin to used for both input and output. When a logic 1 is written to a line, a second high impedance transistor TR2, pulls the line up to 5 V. To provide fast switching during a '0' to '1' transition, a relatively low-impedance transistor TR1 (approx. 750 Ω) is switched on for 1/5 of a machine cycle whenever a '1' is written to the line. Whenever a '0' is written to the line, a low-impedance (approx. 250 Ω) transistor TR4, overcomes the weak light pull-up and provides TTL current sinking capability.

Since the pull-down transistor TR4 is a low-impedance device, a '1' must first be written to any line which is to be used as an input. RESET initializes all lines to the high impedance '1' state. This structure allows input and output on the same pin and also allows a mixture of input lines and output lines on the same port. The quasi-bidirectional port in combination with the ANL and ORL logical instructions provide an efficient means for handling single line inputs and outputs within an 8-bit processor.

BUS

BUS is also an 8-bit port which is a true bidirectional port with associated input and output strobes. If the bidirectional feature is not needed, BUS can serve as either a statically latched output port or non-latching input port. Input and output lines on this port cannot be mixed.
As a static port, data is written and latched using the OUTL instruction and input using the INS instruction. The INS and OUTL instructions generate pulses on the corresponding $\overline{RD}$ and $\overline{WR}$ output strobe lines; however, in the static port mode they are generally not used. As a bidirectional port, the MOVX instructions are used to read and write the port. A write to the port generates a pulse on the $\overline{WR}$ output line and output data is valid at the trailing-edge of $\overline{WR}$. A read of the port generates a pulse on the $\overline{RD}$ output line and input data must be valid at the trailing-edge of $\overline{RD}$. When not being written or read, the BUS lines are high impedance.

Test (T0, T1) and $\overline{INT}$ inputs

Three pins serve as inputs and are testable with the conditional jump instruction. These pins are T0, T1, and $\overline{INT}$ and they allow inputs to cause program branches without the necessity to load an input port into the accumulator. The T0, T1 and $\overline{INT}$ pins have other possible functions as well.

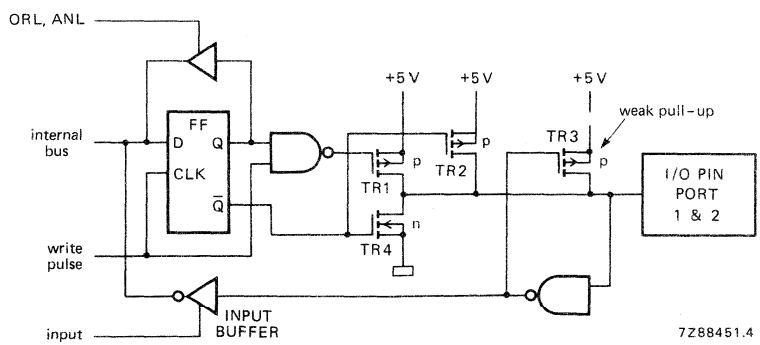


Fig. 8 Quasi-bidirectional port structure.

$\overline{RESET}$ input (see Fig. 10)

The $\overline{RESET}$ input provides a means to initialize the processor. This Schmitt-trigger input has an internal pull-up resistor. The combination of an external 47 kΩ resistor and a 1 μF capacitor provides a reset pulse of sufficient duration to guarantee that all circuitry is reset. If the reset pulse is generated otherwise, the $\overline{RESET}$ pin must be held at ground for at least 10 ms after the (0,13 V_{CC}) power supply is within tolerance. Only five machine cycles (2,5 μs at 6 MHz) are required if power is already on and the oscillator has stabilized.

Single step input ($\overline{SS}$)

Under control of the $\overline{SS}$ line, the processor can be forced to execute one instruction and then to wait until the single step switch is activated again.

FUNCTIONAL DESCRIPTION (continued)

IDLE mode

The PCB80CXX family is provided with a IDLE mode in which the internal oscillator, the internal timer and the external interrupt and counter are still functioning, while the status of following parts is maintained: RAM and register/Port 1 and 2/Bus. The IDLE mode is entered after execution of the IDLE instruction (opcode 1H). The IDLE mode is terminated by one of the two possible interrupts, if enabled, or a RESET signal. If an external interrupt terminates the IDLE mode, the next instruction that is executed is at location 3 of the program store. If a timer/counter interrupt terminates the IDLE mode, the next instruction is at location 7. The reset signal will terminate the IDLE mode, and also initialize the processor.

Power-down mode

In the PCB80CXX family, power can be removed from all but the 64 x 8 bit and 128 x 8 bit data RAM array, for low power standby operation. In the power-down mode the contents of the data RAM are maintained. V_{CC} serves as the + 5 V supply pin for most of the circuitry, while the V_{DD} pin supplies only the RAM array. In normal operation, both pins are at + 5 V. In standby, V_{CC} is at ground and V_{DD} is maintained at + 5 V.

Applying $\overline{\text{RESET}}$ to the processor through the $\overline{\text{RESET}}$ pin inhibits any access to the RAM by the processor and guarantees that the RAM cannot be inadvertently altered when power is removed from V_{CC}. Fig. 9 shows a typical power-down sequence.

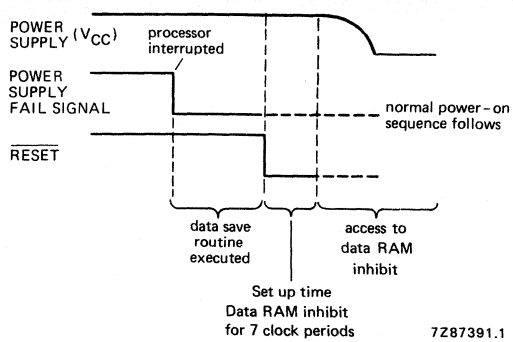


Fig. 9 Power-down sequence.

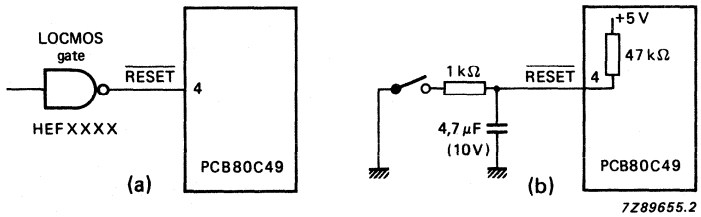


Fig. 10(a) External reset circuit; (b) power-on reset.

DEVELOPMENT DATA

Instruction set

The PCB80CXX instruction set consists of over 90 one and two-byte instructions (see Table 2). Program code efficiency is high because:

- working registers and program variables are stored in the RAM, which require only a single byte to address,
- program memory is divided into pages of 256 bytes, which means that branch destination addresses require one byte.

In addition to performing logical and arithmetic operations, the instruction set manipulates and tests both bits and bytes. A set of MOVE instructions operates indirectly upon either RAM or ROM, which permits efficient access of pointers and data tables. The indirect jump instruction performs a multi-way branch (up to 256) upon the content of the accumulator to addresses stored in a look-up table. The 'decrement register and jump if not zero' DJNZ instruction saves a byte every time it is used as opposed to using separate increment and test instructions.

The on-chip counter enables either external events or time to be counted off-line from the main program. The PCB80CXX can either test the counter (under program control) or cause its overflow to generate an interrupt. These features are required for real-time applications. Instruction timing is shown in Table 3.

Table 1

Symbol definitions used in Table 2.

symbol	description
A	the accumulator
AC	the auxiliary carry flag
addr	program memory address (11-bits)
Bb	bit designation (b = 0–7)
BS	the bank switch
C	carry flag
CLK	clock signal
CNT	event counter
D	nibble designation (4-bits)
DBF	program memory bank flip-flop
data	number or expression (8-bits)
F0, F1	flags 0 and 1
$\overline{\text{I}}$	interrupt
$\overline{\text{INT}}$	external interrupt
P	'in-page' operation designation
Pp	port designation (p = 1, 2 or 4–7)
PSW	program status word
Rr	register designation (r = 0, 1 or 0–7)
SP	stack pointer
T	timer
TF	timer flag
T0, T1,	test 0 and 1 inputs
#	immediate data prefix
@	indirect address prefix
$	current value of program counter
←	is replaced by
↔	is exchanged with

Notes to Table 2.

1. Instruction code designations r and p form the binary representation of the registers and ports involved.
2. The dot under the appropriate flag bit indicates that its content is subject to change by the instruction it appears in.
3. Numerical subscripts appearing in the FUNCTION column reference the specific bits affected.

Table 2 Instruction set

	mnemonic	function	description	instruction code D_7 D_6 D_5 D_4 D_3 D_2 D_1 D_0	cycles	bytes	flags C	AC	F0	F1	BS
ACCUMULATOR	ADD A,Rr	(A)←(A) + (Rr) for r = 0–7	Add contents of designated register to A	0 1 1 0 1 r r r	1	1	•	•			
	ADD A,@Rr	(A)←(A) + ((Rr)) for r = 0–1	Add indirect the contents of the data memory location to A	0 1 1 0 0 0 0 r	1	1	•	•			
	ADD A,#data	(A)←(A) + data	Add immediate data to A	0 0 0 0 0 0 1 1 d_7 d_6 d_5 d_4 d_3 d_2 d_1 d_0	2	2	•	•			
	ADDC A,Rr	(A)←(A) + (C) + (Rr) for r = 0–7	Add with carry the contents of designated register to A	0 1 1 1 1 r r r	1	1	•	•			
	ADDC A,@Rr	(A)←(A) + (C) + ((Rr)) for r = 0–1	Add indirect with carry the contents of the data memory location to A	0 1 1 1 0 0 0 r	1	1	•	•			
	ADDC A,#data	(A)←(A) + (C) + data	Add immediate data with carry to A	0 0 0 1 0 0 1 1 d_7 d_6 d_5 d_4 d_3 d_2 d_1 d_0	2	2	•	•			
	ANL A,Rr	(A)←(A) AND (Rr) for r = 0–7	Logical AND contents of designated register with A	0 1 0 1 1 r r r	1	1					
	ANL A,@Rr	(A)←(A) AND ((Rr)) for r = 0–1	Logical AND indirect the contents of data memory with A	0 1 0 1 0 0 0 r	1	1					
	ANL A,#data	(A)←(A) AND data	Logical AND immediate data with A	0 1 0 1 0 0 1 1 d_7 d_6 d_5 d_4 d_3 d_2 d_1 d_0	2	2					
	CLR A	(A)←0	Clear the contents of A	0 0 1 0 0 1 1 1	1	1					
	CPL A	(A)←NOT(A)	Complement the contents of A	0 0 1 1 0 1 1 1	1	1					
	DA A		Decimal adjust the contents of A	0 1 0 1 0 1 1 1	1	1	•	•			
	DEC A	(A)←(A) −1	Decrement the contents of A by 1	0 0 0 0 0 1 1 1	1	1					
	INC A	(A)←(A) + 1	Increment the contents of A by 1	0 0 0 1 0 1 1 1	1	1					
	ORL A,Rr	(A)←(A) OR (Rr) for r = 0–7	Logical OR contents of designated register with A	0 1 0 0 1 r r r	1	1					
	ORL A,@Rr	(A)←(A) OR ((Rr)) for r = 0–1	Logical OR indirect the contents of data memory location with A	0 1 0 0 0 0 0 r	1	1					

ACCUMULATOR (continued)	ORL A,#data	(A)←(A) OR data	Logical OR immediate data with A	0 d_7	1 d_6	0 d_5	0 d_4	0 d_3	0 d_2	1 d_1	1 d_0	2	2			
	RL A	$(A_{n+1})\leftarrow(A_n)$ $(A_0)\leftarrow(A_7)$ for n = 0–6	Rotate A left by 1-bit without carry	1	1	1	0	0	1	1	1	1	1			
	RLC A	$(A_{n+1})\leftarrow(A_n)$ $(A_0)\leftarrow(C)$ $(C)\leftarrow(A_7)$ for n = 0–6	Rotate A left by 1-bit through carry	1	1	1	1	0	1	1	1	1	1	•		
	RR A	$(A_n)\leftarrow(A_{n+1})$ $(A_7)\leftarrow(A_0)$ for n = 0–6	Rotate A right by 1-bit without carry	0	1	1	1	0	1	1	1	1	1			
	RRC A	$(A_n)\leftarrow(A_{n+1})$ $(A_7)\leftarrow(C)$ $(C)\leftarrow(A_0)$ n = 0–6	Rotate A right by 1-bit through carry	0	1	1	0	0	1	1	1	1	1	•		
	SWAP A	$(A_{4-7})\leftrightarrow(A_{0-3})$	Swap the two 4-bit nibbles in A	0	1	0	0	0	1	1	1	1	1			
	XRL A,Rr	(A)←(A) XOR (Rr)	Logical XOR contents of designated register with A	1	1	0	1	1	r	r	r	1	1			
	XRL A,@Rr	(A)←(A) XOR ((Rr))	Logical XOR indirect the contents of data memory location with A	1	1	0	1	0	0	0	r	1	1			
	XRL A,#data	(A)←(A) XOR data	Logical XOR immediate data with A	1 d_7	1 d_6	0 d_5	1 d_4	0 d_3	0 d_2	1 d_1	1 d_0	2	2			

Table 2 (continued)

	mnemonic	function	description	instruction code D_7 D_6 D_5 D_4 D_3 D_2 D_1 D_0	cycles	bytes	flags C	AC	F0	F1	BS
BRANCH	DJNZ Rr, addr	(Rr)←(Rr) −1 if (Rr) not zero: (PC_{0-7})←addr	Decrement the specified register and test contents	1 1 1 0 1 r r r a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0	2	2					
	JBb addr	(PC_{0-7})←addr if Bb = 1; (PC)←(PC) + 2 if Bb = 0	Jump to specified address if accumulator bit is set	b_2 b_1 b_0 1 0 0 1 0 a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0	2	2					
	JC addr	(PC_{0-7})←addr if C = 1; (PC)←(PC) + 2 if C = 0	Jump to specified address if carry flag is set	1 1 1 1 0 1 1 0 a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0	2	2					
	JF0 addr	(PC_{0-7})←addr if (F0 = 1; (PC)←(PC) + 2 if F0 = 0	Jump to specified address if flag F0 is set	1 0 1 1 0 1 1 0 a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0	2	2					
	JF1 addr	(PC_{0-7})←addr if F1 = 1; (PC)←(PC) + 2 if F1 = 0	Jump to specified address if flag F1 is set	0 1 1 1 0 1 1 0 a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0	2	2					
	JMP addr	(PC_{8-10})←$addr_{8-10}$ (PC_{0-7})←$addr_{0-7}$ (PC_{11})←(DBF)	Direct jump to specified address within the 2K address block	a_{10} a_9 a_8 0 0 1 0 0 a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0	2	2					
	JMPP @A	(PC_{0-7})←((A))	Jump indirect to specified address within address page	1 0 1 1 0 0 1 1	2	1					
	JNC addr	(PC_{0-7})←addr if C = 0; (PC)←(PC) + 2 if C = 1	Jump to specified address if carry flag is LOW	1 1 1 0 0 1 1 0 a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0	2	2					
	JNI	(PC_{0-7})←addr if $\overline{INT}$ = 0; (PC)←(PC) + 2 if $\overline{INT}$ = 1	Jump to specified address if $\overline{INT}$ input is LOW	1 0 0 0 0 1 1 0 a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0	2	2					
	JNT0 addr	(PC_{0-7})←addr if T0 = 0; (PC)←(PC) + 2 if T0 = 1	Jump to specified address if T0 is LOW	0 0 1 0 0 1 1 0 a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0	2	2					
	JNT1 addr	(PC_{0-7})←addr if T1 = 0; (PC)←(PC) + 2 if T1 = 1	Jump to specified address if T1 is LOW	0 1 0 0 0 1 1 0 a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0	2	2					

BRANCH (continued)	JNZ addr	$(PC_{0-7}) \leftarrow$addr if $A \neq 0$; $(PC) \leftarrow (PC) + 2$ if $A = 0$	Jump to specified address if A is non-zero	1 a_7	0 a_6	0 a_5	1 a_4	0 a_3	1 a_2	1 a_1	0 a_0	2	2					
	JTF addr	$(PC_{0-7}) \leftarrow$addr if TF = 1; $(PC) \leftarrow (PC) + 2$ if TF = 0	Jump to specified address if timer flag is set to 1	0 a_7	0 a_6	0 a_5	1 a_4	0 a_3	1 a_2	1 a_1	0 a_0	2	2					
	JT0 addr	$(PC_{0-7}) \leftarrow$addr if T0 = 1; $(PC) \leftarrow (PC) + 2$ if T0 = 0	Jump to specified address if T0 = 1	0 a_7	0 a_6	1 a_5	1 a_4	0 a_3	1 a_2	1 a_1	0 a_0	2	2					
	JT1 addr	$(PC_{0-7}) \leftarrow$addr if T1 = 1; $(PC) \leftarrow (PC) + 2$ if T1 = 0	Jump to specified address if T1 = 1	0 a_7	1 a_6	0 a_5	1 a_4	0 a_3	1 a_2	1 a_1	0 a_0	2	2					
	JZ addr	$(PC_{0-7}) \leftarrow$addr if $A = 0$; $(PC) \leftarrow (PC) + 2$ if $A \neq 0$	Jump to specified address if A is zero	1 a_7	1 a_6	0 a_5	0 a_4	0 a_3	1 a_2	1 a_1	0 a_0	2	2					
CONTROL	EN I		Enable external ($\overline{INT}$) interrupt	0	0	0	0	0	1	0	1	1	1					
	DIS I		Disable external ($\overline{INT}$) interrupt	0	0	0	1	0	1	0	1	1	1					
	SEL RB0	(BS)←0	Select bank 0 (locations 0–7) of data memory	1	1	0	0	0	1	0	1	1	1					•
	SEL RB1	(BS)←1	Select bank 1 (locations 24–31) of data memory	1	1	0	1	0	1	0	1	1	1					•
	SEL MB0	(DBF)←0	Select program memory bank 0; addresses 0–2047	1	1	1	0	0	1	0	1	1	1					
	SEL MB1	(DBF)←1	Select program memory bank 1; addresses 2048–4095	1	1	1	1	0	1	0	1	1	1					
	ENT0 CLK		Enable clock output onto T0	0	1	1	1	0	1	0	1	1	1					

Table 2 (continued)

	mnemonic	function	description	instruction code D_7 D_6 D_5 D_4 D_3 D_2 D_1 D_0	cycles	bytes	flags C	AC	F0	F1	BS
DATA MOVES	MOV A,#data	(A)←data	Move immediate data into A	0 0 1 0 0 0 1 1 d_7 d_6 d_5 d_4 d_3 d_2 d_1 d_0	2	2					
	MOV A,Rr	(A)←(Rr) for r = 0–7	Move the contents of the designated register into A	1 1 1 1 1 r r r	1	1					
	MOV A,@Rr	(A)←((R)) for r = 0–1	Move indirect the contents of data memory into A	1 1 1 1 0 0 0 r	1	1					
	MOV A,PSW	(A)←(PSW)	Move contents of the program status word into A	1 1 0 0 0 1 1 1	1	1					
	MOV Rr,#data	(Rr)←data for r = 0–7	Move immediate data into the designated register	1 0 1 1 1 r r r d_7 d_6 d_5 d_4 d_3 d_2 d_1 d_0	2	2					
	MOV Rr,A	(Rr)←(A) for r = 0–7	Move A contents into the designated register	1 0 1 0 1 r r r	1	1					
	MOV @Rr,A	((Rr))←(A) for r = 0–1	Move indirect A contents into data memory location	1 0 1 0 0 0 0 r	1	1					
	MOV @Rr,#data	((Rr))←data for r = 0–1	Move indirect the specified data into data memory	1 0 1 1 0 0 0 r d_7 d_6 d_5 d_4 d_3 d_2 d_1 d_0	2	2					
	MOV PSW,A	(PSW)←A	Move contents of A into the program status word	1 1 0 1 0 1 1 1	1	1	•	•	•		•
	MOVP A,@A	(A)←((A))	Move data in the current page into A	1 0 1 0 0 0 1 1	2	1					
	MOVP3 A,@A	(A)←((A)) in page 3	Move data in page 3 of memory bank Ø into A	1 1 1 0 0 0 1 1	2	1					
	MOVX A,@Rr	(A)←((R)) for r = 0–1	Move indirect the contents of external memory location into A	1 0 0 0 0 0 0 r	2	1					
	MOVX @Rr,A	((Rr))←(A) for r = 0–1	Move indirect the contents of A into external memory	1 0 0 1 0 0 0 r	2	1					
	XCH A,Rr	(A)↔(Rr) for r = 0–7	Exchange A with designated register contents	0 0 1 0 1 r r r	1	1					
	XCH A,@Rr	(A)↔((R)) for r = 0–1	Exchange indirect A contents with location in data memory	0 0 1 0 0 0 0 r	1	1					

DEVELOPMENT DATA

D.M.	XCHD A,@Rr	$(A_{0-3}) \leftrightarrow (Rr_{0-3})$ for r = 0–1	Exchange indirect 4-bit contents of A with data memory	0	0	1	1	0	0	0	r	1	1					
FLAGS	CPL C	(C)←NOT (C)	Complement content of carry bit	1	0	1	0	0	1	1	1	1	1	•				
	CPL F0	(F0)←NOT (F0)	Complement content of flag F0	1	0	0	1	0	1	0	1	1	1			•		
	CPL F1	(F1)←NOT (F1)	Complement content of flag F1	1	0	1	1	0	1	0	1	1	1				•	
	CLR C	(C)←0	Clear content of carry bit to 0	1	0	0	1	0	1	1	1	1	1	•				
	CLR F0	(F0)←0	Clear content of flag F0 to 0	1	0	0	0	0	1	0	1	1	1			•		
	CLR F1	(F1)←0	Clear content of flag F1 to 0	1	0	1	0	0	1	0	1	1	1				•	
INPUT/OUTPUT	ANL BUS,#data	(BUS)←(BUS) AND data	Logical AND immediate data with BUS	1 d_7	0 d_6	0 d_5	1 d_4	1 d_3	0 d_2	0 d_1	0 d_0	2	2					
	ANL Pp,#data	(Pp)←(Pp) AND data; p = 1–2	Logical AND immediate data with designated port (1 or 2)	1 d_7	0 d_6	0 d_5	1 d_4	1 d_3	0 d_2	p d_1	p d_0	2	2					
	ANLD Pp,A	(Pp)←(Pp) AND (A_{0-3}); p = 4–7	Logical AND contents of A with designated port (4–7)	1	0	0	1	1	1	p	p	2	1					
	IN A,Pp	(A)←(Pp) p = 1–2	Input data from designated port (1–2) into A	0	0	0	0	1	0	p	p	2	1					
	INS A,BUS	(A)←(BUS)	Input strobed BUS data into A	0	0	0	0	1	0	0	0	2	1					
	MOVD A,Pp	(A_{0-3})←(Pp); p = 4–7 (A_{4-7})←0	Move contents of designated port (4–7) into A	0	0	0	0	1	1	p	p	2	1					
	MOVD Pp,A	(Pp)←(A_{0-3}) p = 4–7	Move contents of A to designated port (4–7)	0	0	1	1	1	1	p	p	1	1					
	ORLD Pp,A	(Pp)←(Pp) OR (A_{0-3}); p = 4–7	Logical OR contents of A with designated port (4–7)	1	0	0	0	1	1	p	p	1	1					
	ORL BUS,#data	(BUS)←(BUS) OR data	Logical OR immediate data with BUS	1 d_7	0 d_6	0 d_5	0 d_4	1 d_3	0 d_2	0 d_1	0 d_0	2	2					
	ORL Pp,#data	(Pp)←(Pp) OR data p = 1–2	Logical OR immediate data with designated port (1–2)	1 d_7	0 d_6	0 d_5	0 d_4	1 d_3	0 d_2	p d_1	p d_0	2	2					
	OUTL BUS,A	(BUS)←(A)	Output contents A onto BUS	0	0	0	0	0	0	1	0	2	1					
	OUTL Pp,A	(Pp)←(A) p = 1–2	Output contents A to designated port (1–2)	0	0	1	1	1	0	p	p	1	1					

Table 2 (continued)

	mnemonic	function	description	instruction code D_7 D_6 D_5 D_4 D_3 D_2 D_1 D_0	cycles	bytes	flags C	AC	F0	F1	BS
REGISTER	DEC Rr	(Rr)←(Rr) −1 for r = 0–7	Decrement contents of designated register by 1	1 1 0 0 1 r r r	1	1					
	INC Rr	(Rr)←(Rr) + 1 for r = 0–7	Increment contents of designated register by 1	0 0 0 1 1 r r r	1	1					
	INC @Rr	((R))←((Rr)) + 1 for r = 0–1	Increment indirect the contents of data memory location by 1	0 0 0 1 0 0 0 r	1	1					
SUBROUTINE	CALL addr	((SP))←(PC), (PSW_{4-7}) (SP)←(SP) + 1 (PC_{8-10})←$addr_{8-10}$ (PC_{0-7})←$addr_{0-7}$	Call designated subroutine	a_{10} a_9 a_8 1 0 1 0 0 a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0	2	2					
	RET	(SP)←(SP) −1 (PC)←((SP))	Return from subroutine without restoring program status word	1 0 0 0 0 0 1 1	2	1					
	RETR	(SP)←(SP) −1 (PC)←((SP)) (PSW_{4-7})←((SP))	Return from subroutine restoring program status word	1 0 0 1 0 0 1 1	2	1					
TIMER/COUNTER	EN TCNTI		Enable timer/counter interrupt	0 0 1 0 0 1 0 1	1	1					
	DIS TCNTI		Disable timer/counter interrupt	0 0 1 1 0 1 0 1	1	1					
	MOV A,T	(A)←(T)	Move contents of timer/counter into A	0 1 0 0 0 0 1 0	1	1					
	MOV T,A	(T)←(A)	Move contents of A into timer/counter	0 1 1 0 0 0 1 0	1	1					
	STOP TCNT		Stop count for event counter or timer	0 1 1 0 0 1 0 1	1	1					
	STRT CNT		Start count for event counter	0 1 0 0 0 1 0 1	1	1					
	STRT T		Start count for timer	0 1 0 1 0 1 0 1	1	1					
	IDLE		Entering IDLE mode	0 0 0 0 0 0 0 1	1	1					
	NOP		No operation	0 0 0 0 0 0 0 0	1	1					

Table 3 Instruction timing (see also Figs 11 and 12)

	cycle 1					cycle 2				
instruction	S1	S2	S3	S4	S5	S1	S2	S3	S4	S5
IN A,P	fetch instruction	increment program counter	–	increment timer	–	–	read port	* –	–	–
OUTL P,A	|	|	–	|	output to port	–	–	* –	–	–
ANL P,#data	|	* |	–	|	read port	fetch immediate data	–	*increment program counter	output to port	–
ORL P,#data	|	* |	–	|	read port	fetch immediate data	–	*increment program counter	output to port	–
INS A,BUS	|	|	–	|	–	–	read port	* –	–	–
OUTL BUS,A	|	|	–	|	output to port	–	–	* –	–	–
ANL BUS,#data	|	* |	–	|	read port	fetch immediate data	–	*increment program counter	output to port	–
ORL BUS,#data	|	* |	–	|	read port	fetch immediate data	–	*increment program counter	output to port	–
MOVX @R,A	|	|	output RAM address	|	output data to RAM	–	–	* –	–	–
MOVX A,@R	|	|	output RAM address	|	–	–	read data	* –	–	–
MOVD A,P	fetch instruction	increment program counter	output opcode/ address	increment timer	–	–	read P2 lower	* –	–	–

Table 3 Instruction timing (continued)

	cycle 1					cycle 2				
instruction	S1	S2	S3	S4	S5	S1	S2	S3	S4	S5
MOVD P,A	fetch instruction	increment program counter	output opcode/ address	increment timer	output data to P2 lower	–	–	* –	–	–
ANLD P,A	|		output opcode/ address	|	output data	–	–	* –	–	–
ORLD P,A	|		output opcode/ address	|	output data	–	–	* –	–	–
J (conditional)	|	*	sample condition	increment timer	–	fetch immediate data	–	*update program counter	–	–
STRT CNT STRT T	|	*	–	–	start counter					
STOP TCNT	|	*	–	–	stop counter					
EN I	|	*	–	enable interrupt	–					
DIS I	|	*	–	disable interrupt	–					
ENT0 CLK	fetch instruction	*increment program counter	–	enable clock	–					

* Valid instruction addresses are output at this time if external program memory is being accessed.

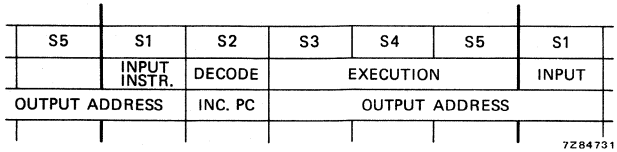


Fig. 11 Instruction cycle.

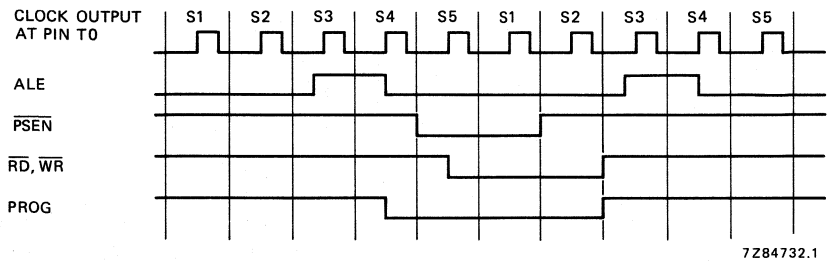


Fig. 12 Instruction cycle timing.

Table 4 Instruction map.

first hex ↓ / second hex →	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NOP	IDLE mode	OUTL BUS,A	ADD A, #data	JMP page 0	EN I		DEC A	INS A,BUS	IN A, Pp	2		MOVD A,Pp	5	6	7
1	INC @Rr	1	JB0 addr	ADDC A, #data	CALL page 0	DIS I	JTF addr	INC A	INC Rr	1	2	3	4	5	6	7
2	XCH A, @Rr	1		MOV A, #data	JMP page 1	EN TCNTI	JNT0 addr	CLR A	XCH A,Rr	1	2	3	4	5	6	7
3	XCHD A, @Rr	1	JB1 addr		CALL page 1	DIS TCNTI	JT0 addr	CPL A		OUTL Pp,A	2		MOVD Pp,A	5	6	7
4	ORL A, @Rr	1	MOV A, T	ORL A, #data	JMP page 2	STRT CNT	JNT1 addr	SWAP A	ORL A,Rr	1	2	3	4	5	6	7
5	ANL A, @Rr	1	JB2 addr	ANL A, #data	CALL page 2	STRT T	JT1 addr	DA, A	ANL A,Rr	1	2	3	4	5	6	7
6	ADD A, @Rr	1	MOV T,A		JMP page 3	STOP TCNT		RRC A	ADD A,Rr	1	2	3	4	5	6	7
7	ADDC A, @Rr	1	JB3 addr		CALL page 3	ENT0 CLK	JF1 addr	RR A	ADDC A,Rr	1	2	3	4	5	6	7
8	MOVX A, @Rr	1		RET	JMP page 4	CLR F0	JN1 addr		ORL BUS, #data	ORL Pp, #data	2		ORLD Pp,A	5	6	7
9	MOVX @Rr,A	1	JB4 addr	RETR	CALL page 4	CPL F0	JNZ addr	CLR C	ANL BUS, #data	ANP Pp, #data	2		ANLD Pp,A	5	6	7
A	MOV @Rr, A	1		MOVP A, @A	JMP page 5	CLR F1		CPL C	MOV Rr,A	1	2	3	4	5	6	7
B	MOV @Rr, #data	1	JB5 addr	JMPP @A	CALL page 5	CPL F1	JF0 addr		MOV R, #data	1	2	3	4	5	6	7
C					JMP page 6	SEL RB0	JZ addr	MOV A, PSW	DEC Rr	1	2	3	4	5	6	7
D	XRL A, @Rr	1	JB6 addr	XRL A, #data	CALL page 6	SEL RB1		MOV PSW, A	XRL A,Rr	1	2	3	4	5	6	7
E				MOVP3 A @A	JMP page 7	SEL MB0	JNC addr	RL A	DJNZ Rr, addr	1	2	3	4	5	6	7
F	MOV A, @Rr	1	JB7 addr		CALL page 7	SEL MB1	JC addr	RLC A	MOV A,Rr	1	2	3	4	5	6	7

DEVELOPMENT DATA

RATINGS

Limiting values in accordance with the Absolute Maximum System (IEC 134)

D.C. current into any input or output	$\pm I_I, \pm I_O$	max.	10 mA
Total power dissipation	P_{tot}	max.	0,5 W
Storage temperature range	T_{stg}		−65 to + 150 °C
Operating ambient temperature range	T_{amb}		0 to + 70 °C

D.C. CHARACTERISTICS

V_{SS} = 0 V; T_{amb} = 0 to + 70 °C; all voltages with respect to V_{SS}; unless otherwise specified

parameter	symbol	min.	typ.	max.	unit	conditions
Supply voltage	V_{CC} = V_{DD}	4,5	5,0	5,5	V	
Supply current total	I_{tot}	–	–	15	mA	f = 11 MHz; $I_{tot} = I_{CC} + I_{DD}$
		–	–	8,5	mA	f = 6 MHz; $I_{tot} = I_{CC} + I_{DD}$
		–	–	3	mA	f = 1 MHz; $I_{tot} = I_{CC} + I_{DD}$
IDLE mode	I_{idle}	–	–	6	mA	f = 11 MHz; $I_{idle} = I_{CC} + I_{DD}$
		–	–	4	mA	f = 6 MHz; $I_{idle} = I_{CC} + I_{DD}$
		–	–	1,2	mA	f = 1 MHz; $I_{idle} = I_{CC} + I_{DD}$
power down mode	I_{pd}	–	–	2	μA	V_{DD} = 2 V; $\overline{RESET}$ = LOW
Inputs						
Input voltage LOW all inputs except $\overline{RESET}$; XTAL 1; XTAL 2	V_{IL}	−0,5	–	0,18 x V_{CC}	V	
Input voltage LOW $\overline{RESET}$; XTAL 1; XTAL 2	V_{IL1}	−0,5	–	0,13 x V_{CC}	V	
Input voltage HIGH all inputs except $\overline{RESET}$; XTAL 1; XTAL 2	V_{IH}	0,4 x V_{CC}	–	V_{CC}	V	
Input voltage HIGH $\overline{RESET}$; XTAL 1; XTAL 2	V_{IH1}	0,7 x V_{CC}	–	V_{CC}	V	
Outputs						
Output voltage LOW BUS	V_{OL}	–	–	0,45	V	I_{OL} = 2 mA
Output voltage LOW $\overline{RD}$; $\overline{WR}$; $\overline{PSEN}$; ALE	V_{OL1}	–	–	0,45	V	I_{OL} = 1,8 mA
Output voltage LOW PROG	V_{OL2}	–	–	0,45	V	I_{OL} = 1 mA
Output voltage LOW all other outputs	V_{OL3}	–	–	0,45	V	I_{OL} = 1,6 mA

D.C. CHARACTERISTICS (continued)

parameter	symbol	min.	typ.	max.	unit	conditions
Output voltage HIGH BUS	V_{OH}	0,75 x V_{CC}	–	–	V	$-I_{OH}$ = 400 μA
Output voltage HIGH $\overline{RD}$; $\overline{WR}$; $\overline{PSEN}$; ALE	V_{OH1}	0,75 x V_{CC}	–	–	V	$-I_{OH}$ = 100 μA
Output voltage HIGH all other outputs	V_{OH2}	0,75 x V_{CC}	–	–	V	$-I_{OH}$ = 40 μA
Input leakage current $\overline{INT}$; T1; EA	$\pm I_{IL}$	–	–	10	μA	without internal pull-up; $V_{SS} < V_I < V_{CC}$
Input leakage current P10 – P17; P20 – P27; $\overline{SS}$	$-I_{IL1}$	–	–	500	μA	with internal pull-up; $V_{SS} + 0{,}45 < V_I < V_{CC}$
Input leakage current $\overline{RESET}$	$-I_{ILR}$	20	–	300	μA	$V_{SS} < V_I \leqslant V_{IL1}$
Output leakage current BUS; T0 at high impedance state	$\pm I_{OL}$	–	–	10	μA	$V_{SS} + 0{,}45 < V_I < V_{CC}$

DEVELOPMENT DATA

A.C. CHARACTERISTICS

$V_{CC} = V_{DD} = 5\ V \pm 10\%$; $V_{SS} = 0\ V$; $T_{amb} = 0$ to $+70\ ^{o}C$; note 1

See waveforms Figs 13, 14, 15, 16 and 17

parameter	f (t_{CL}) (note 2)	symbol	11 MHz		unit
			min.	max.	
Clock period (note 2)	$1/(f_{XTAL})$	t_{CL}	90,9	1000	ns
ALE pulse width	$7/30t_{CL}-170$	t_{LL}	150	–	ns
Address set-up time to ALE	$1/5t_{CL}-110$	t_{AL}	160	–	ns
Address hold time from ALE	$1/15t_{CL}-40$	t_{LA}	50	–	ns
Control pulse width $\overline{RD}$, $\overline{WR}$	$1/2t_{CL}-200$	t_{CC1}	480	–	ns
Control pulse width $\overline{PSEN}$	$2/5t_{CL}-200$	t_{CC2}	350	–	ns
Data set-up time before $\overline{WR}$	$13/30t_{CL}-200$	t_{DW}	390	–	ns
Data hold time after $\overline{WR}$ (note 3)	$1/15t_{CL}-50$	t_{WD}	40	–	ns
Data hold time $\overline{RD}$, $\overline{PSEN}$	$1/10t_{CL}-30$	t_{DR}	0	110	ns
$\overline{RD}$ to data input	$2/5t_{CL}-170$	t_{RD1}	–	375	ns
$\overline{PSEN}$ to data input	$3/10t_{CL}-170$	t_{RD2}	–	240	ns
Address set-up time to $\overline{WR}$	$1/3t_{CL}-150$	t_{AW}	300	–	ns
Address set-up time to data input ($\overline{RD}$)	$7/10t_{CL}-250$	t_{AD1}	–	730	ns
Address set-up time to data input ($\overline{PSEN}$)	$1/2t_{CL}-220$	t_{AD2}	–	460	ns
Address floating to $\overline{RD}$, $\overline{WR}$	$2/15t_{CL}-40$	t_{AFC1}	140	–	ns
Address floating to $\overline{PSEN}$	$1/30t_{CL}-40$	t_{AFC2}	10	–	ns
ALE to control pulse $\overline{RD}$, $\overline{WR}$	$1/5t_{CL}-75$	t_{LAFC1}	200	–	ns
ALE to control pulse $\overline{PSEN}$	$1/10t_{CL}-75$	t_{LAFC2}	60	–	ns
Control pulse to ALE $\overline{RD}$, $\overline{WR}$, $\overline{PROG}$	$1/15t_{CL}-40$	t_{CA1}	50	–	ns
Control pulse to ALE $\overline{PSEN}$	$4/15t_{CL}-40$	t_{CA2}	320	–	ns
Port control set-up to $\overline{PROG}$	$1/10t_{CL}-80$	t_{CP}	50	–	ns

A.C. CHARACTERISTICS (continued)

parameter	f (t_{CL})	symbol	11 MHz		unit
			min.	max.	
Port control hold to $\overline{PROG}$	$4/15t_{CL}-260$	t_{PC}	100	–	ns
$\overline{PROG}$ to time port 2 input must be valid	$17/30t_{CL}-120$	t_{PR}	–	650	ns
Input data hold time from $\overline{PROG}$	$1/10t_{CL}$	t_{PF}	0	140	ns
Output data set-up time	$2/5t_{CL}-290$	t_{DP}	250	–	ns
Output data hold time	$1/10t_{CL}-90$	t_{PD}	40	–	ns
$\overline{PROG}$ pulse width	$7/10t_{CL}-250$	t_{PP}	700	–	ns
Port 2 I/O data set-up time to ALE	$4/15t_{CL}-200$	t_{PL}	160	–	ns
Port 2 I/O data hold time to ALE	$1/10t_{CL}-120$	t_{LP}	15	–	ns
Port output from ALE	$3/10t_{CL}+100$	t_{PV}	–	510	ns
Cycle time	$(1/f_{XTAL}) \times 15$	t_{CY}	1,36	15	μs
T0 repetition rate	$3/15t_{CL}$	t_{0PRR}	270	–	ns
Clock period (note 2)	$1/(f_{XTAL})$	t_{CL}	98,8	1000	ns

Notes to A.C. characteristics

1. Control outputs: C_L = 80 pF
   Bus outputs: C_L = 150 pF.
2. f(t_{CL}) assumes 50% duty cycle on XTAL 1 and XTAL 2; minimum frequency = 1 MHz.
3. Bus high-impedance load: 20 pF.

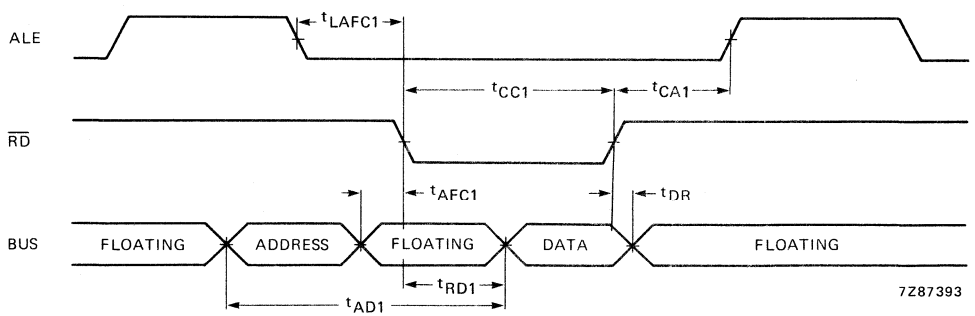


Fig. 13 Read from external data memory.

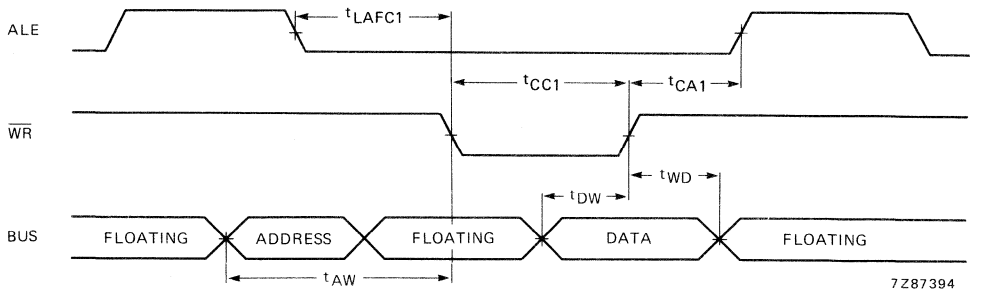


Fig. 14 Write to external data memory.

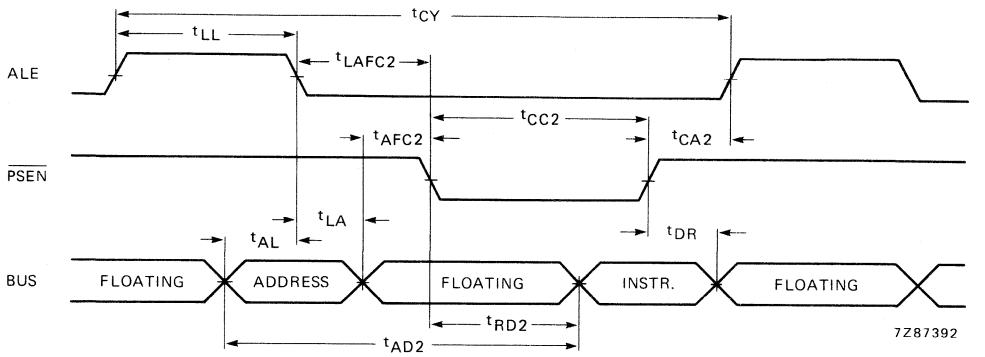


Fig. 15 Instruction fetch from external program memory.

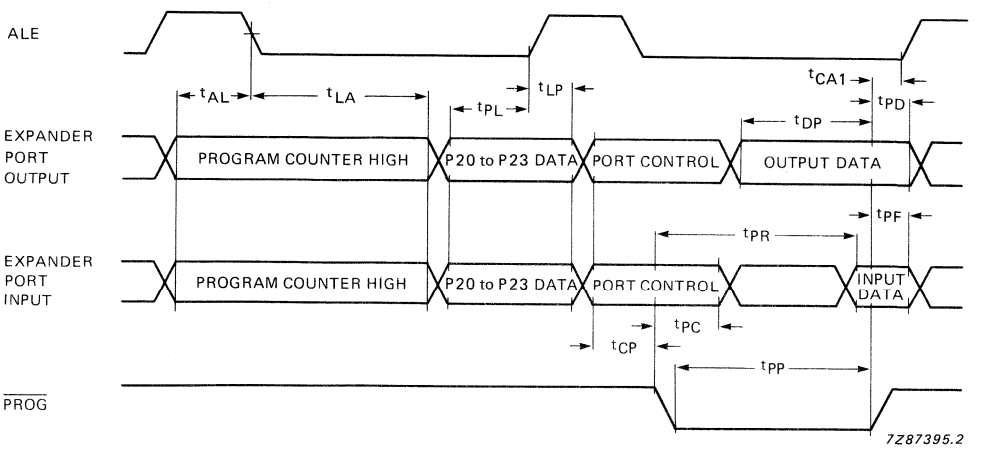


Fig. 16 Port 2 timing.

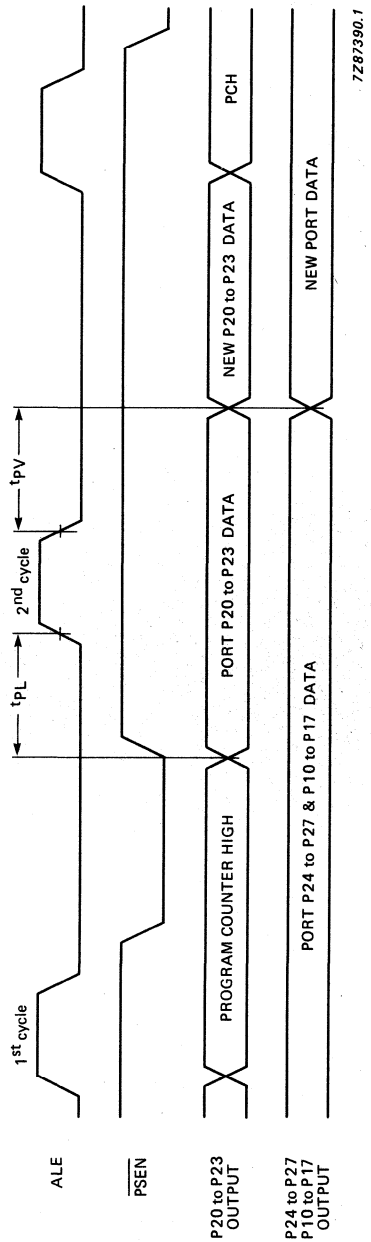


Fig. 17 I/O port timing.

SINGLE-CHIP 8-BIT MICROCONTROLLER

GENERAL DESCRIPTION

The PCB83C552 single-chip 8-bit microcontroller is manufactured in an advanced CMOS process and is a derivative of the PCB80C51 microcontroller family. PCF83C552 has the same instruction set as the PCB80C51. Two versions of the derivative exist although the generic term "PCB83C552" is used to refer to both family members:

- PCB80C552: ROM-less version of the PCB83C552
- PCB83C552: 8 K bytes mask programmable ROM, 256 bytes RAM

This I/O intensive device provides architectural enhancements to function as a controller in the field of automotive electronics, specifically engine management and gear box control.

The PCB83C552 contains a non-volatile 8 K x 8 read-only program memory (not ROM-less version), a volatile 256 x 8 read/write data memory; six 8-bit I/O ports; two 16-bit timer/event counters (identical to the timers of the 80C51); an additional 16-bit timer coupled to capture and compare latches; a fifteen-source, two-priority-level, nested interrupt structure; an ADC and DAC pulse width modulated interface, two serial interfaces; UART and I^2C-bus, a 'watchdog' timer and on-chip oscillator and timing circuits. For systems that require extra capability, the PCB83C552 can be expanded using standard TTL compatible memories and logic.

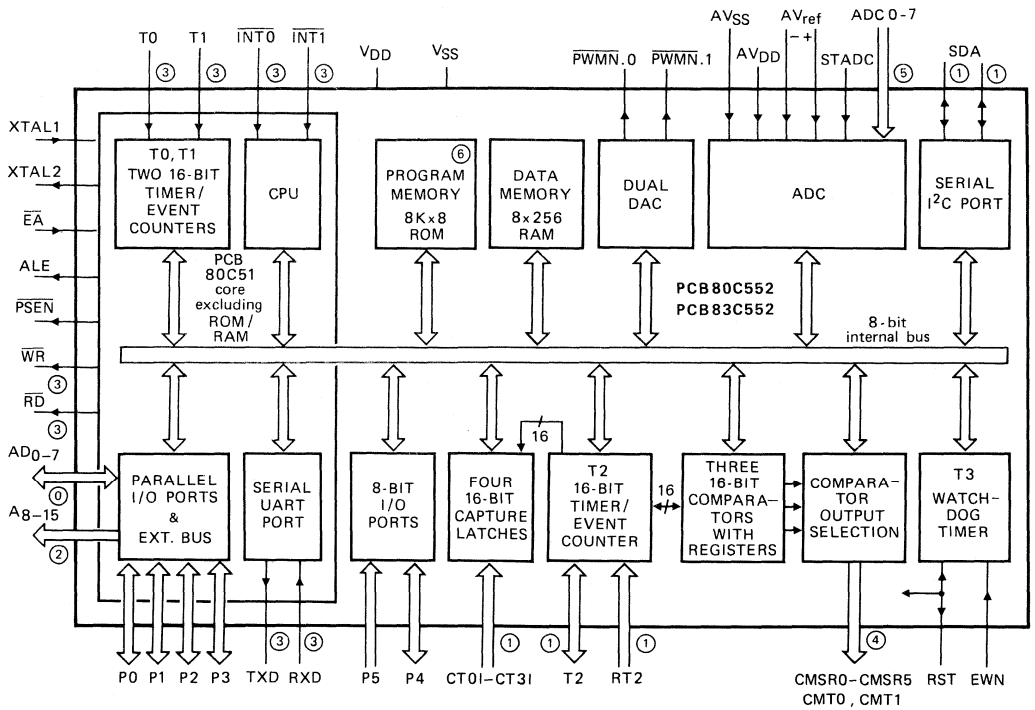
The device also functions as an arithmetic processor having facilities for both binary and BCD arithmetic plus bit-handling capabilities. The instruction set consists of over 100 instructions; 44% one-byte, 41% two-byte and 15% three-byte. With a 12 MHz crystal, 58% of the instructions are executed in 1 μs and 40% in 2 μs. Multiply and divide instructions require 4 μs.

Features

- 80C51 central processing unit
- 8 K x 8 ROM, expandable externally to 64 K bytes
- 256 x 8 RAM, expandable externally to 64 K bytes
- Two standard 16-bit timer/counters
- An additional 16-bit timer/counter coupled to four capture registers and three compare registers
- An A/D converter with 8 multiplexed analogue inputs and 10-bit resolution
- Two 8-bit resolution, Pulse Width Modulated analogue outputs
- Five 8-bit I/O ports plus one 8-bit input port shared with analogue inputs
- I^2C Bus serial I/O port with byte orientated master and slave functions
- Full-duplex UART compatible with the standard PCB80C51
- On-chip watchdog timer

PACKAGE OUTLINE

PCB83C552: 68-lead PLCC; plastic, leaded-chip-carrier (SOT-188A).



(0) alternative function of port 0
(1) alternative function of port 1
(2) alternative function of port 2
(3) alternative function of port 3
(4) alternative function of port 4
(5) alternative function of port 5
(6) not present in PCB80C552

POWER SUPPLY
V_{CC} + 5 V MAIN SUPPLY
V_{SS} GROUND

Fig. 1 Block diagram.

DEVELOPMENT DATA

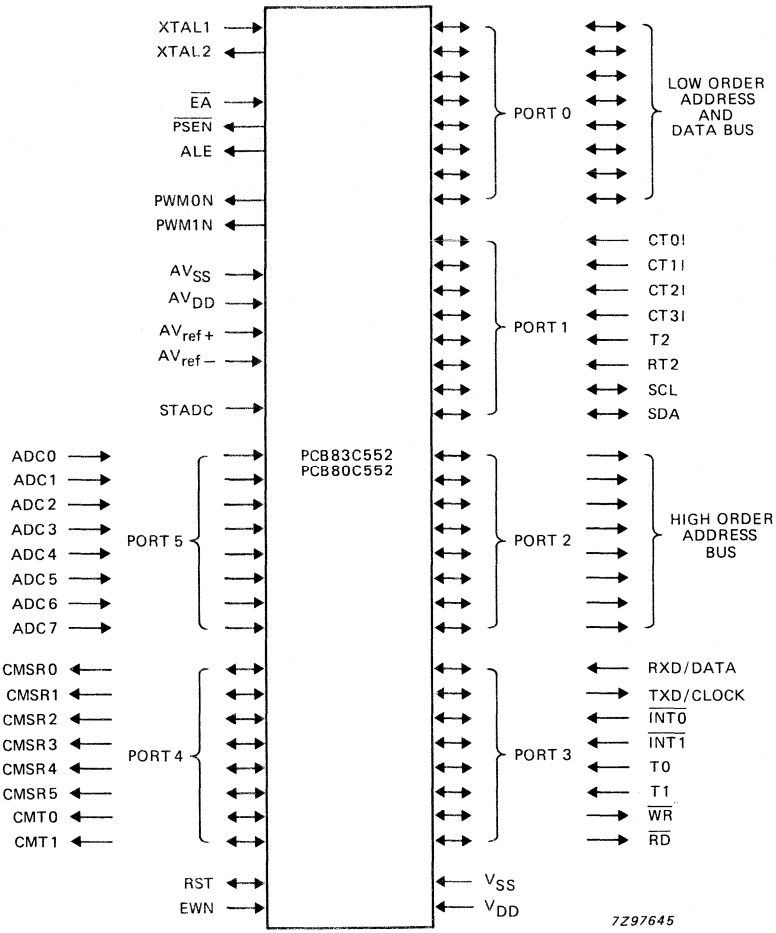


Fig. 2 Functional diagram.

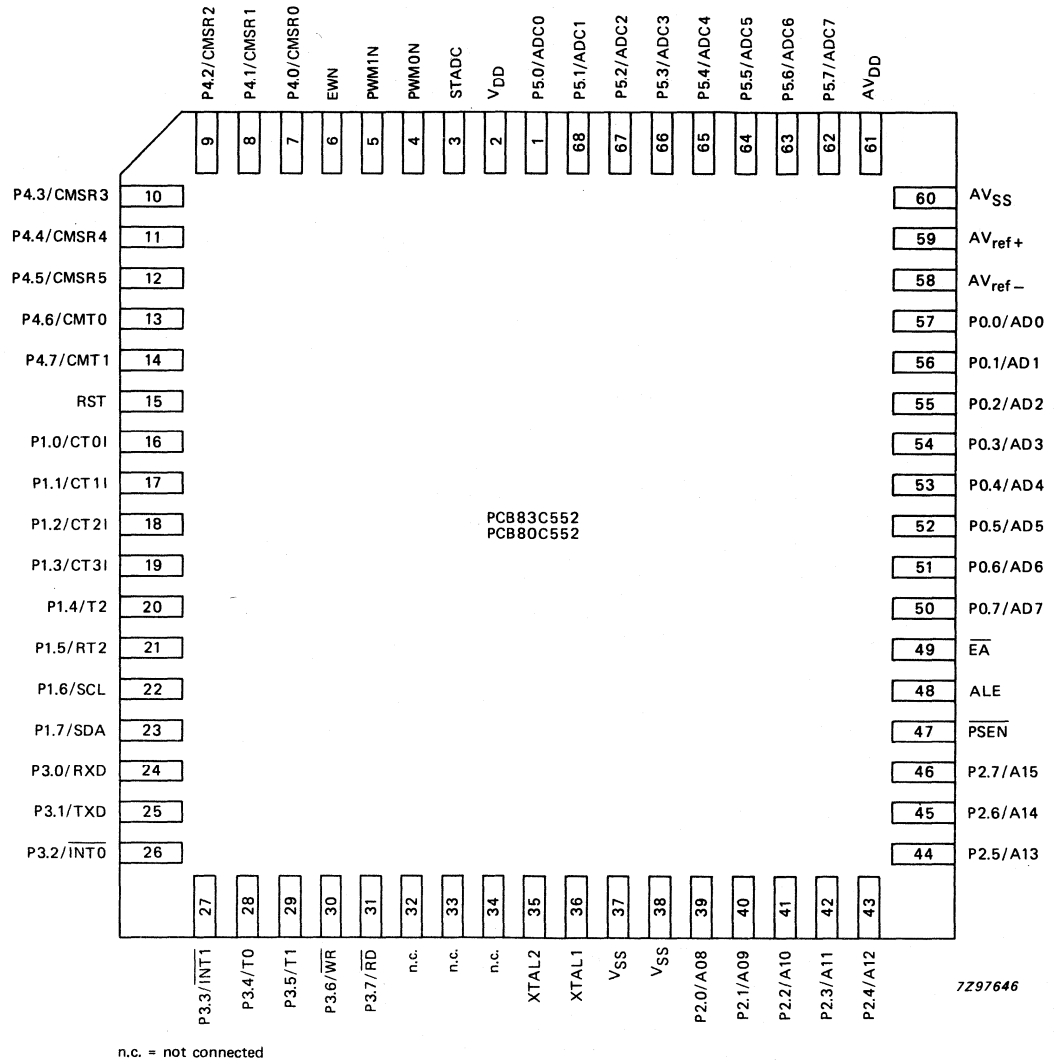


Fig. 3 Pinning diagram for PCB83C552.

DEVELOPMENT DATA

PINNING

2	V_{DD}	**Digital power supply**: + 5 V power supply pin during normal operation, idle mode and power down mode
3	STADC	**Start ADC operation**: Input starting analogue to digital conversion (ADC operation can also be started by software)
4	PWM0N	**Pulse width** modulation output 0
5	PWM1N	**Pulse width** modulation output 1
6	EWN	**Enable watchdog timer**: Enable for T3 watchdog timer and disable power down mode
7-14	P4.0-P4.7	**Port 4** 8-bit quasi-bidirectional I/O port

Port pin	Alternative function	
P4.0	CMSR0	: Timer T2; Compare and set/reset outputs on
P4.1	CMSR1	: a match with timer T2
P4.2	CMSR2	
P4.3	CMSR3	
P4.4	CMSR4	
P4.5	CMSR5	
P4.6	CMT0	: Timer T2; Compare and toggle outputs on a
P4.7	CMT1	match with timer T2

15	RST	**Reset**: Input to reset the PCF83C552. It also provides a reset pulse as output when timer T3 overflows.
16-23	P1.0-P1.7	**Port 1** 8-bit quasi-bidirectional I/O port

Port pin	Alternative function	
P1.0	CT0I	: Capture timer input signals for timer T2
P1.1	CT1I	
P1.2	CT2I	
P1.3	CT3I	
P1.4	T2	: T2 event input
P1.5	RT2	: T2 timer reset signal. Rising edge triggered
P1.6	SCL	: Serial port clock line I^2C bus
P1.7	SDA	: Serial port data line I^2C bus

24-31	P3.0-P3.7	**Port 3** 8-bit quasi-bidirectional I/O port

Port pin	Alternative function	
P3.0	RXD	: Serial input port
P3.1	TXD	: Serial output port
P3.2	$\overline{INT0}$	: External interrupt
P3.3	$\overline{INT1}$	: External interrupt
P3.4	T0	: Timer 0 external input
P3.5	T1	: Timer 1 external input
P3.6	$\overline{WR}$	: External data memory write strobe
P3.7	$\overline{RD}$	: External data memory read strobe

PINNING (continued)

36	XTAL1	**Connection** to timing component (crystal) that determines the frequency of the internal oscillator. It is also the input for an external clock source
35	XTAL2	**Connection** to other side of timing component
37, 38	V_{SS}	**Two digital** ground pins
		Port 2
39-46	P2.0-P2.7	8-bit quasi-bidirectional I/O port Pin — Alternative function P2.0-P2.7 — High-order address byte for external memory (A08-A15)
47	$\overline{PSEN}$	**Program store enable**: active LOW read strobe to external program memory
48	ALE	**Address latch enable**: latches the low byte of the address during accesses to external memory. It is activated every six oscillator periods. During a data memory access one ALE pulse is skipped. ALE can drive up to 8 LS TTL inputs and handles CMOS inputs without an external pull-up
49	$\overline{EA}$	**External access**: When $\overline{EA}$ is held at TTL level HIGH, the CPU executes out of the internal program ROM, provided the program counter is less than 8192. When $\overline{EA}$ is held at TTL LOW level, the CPU executes out of external program memory. $\overline{EA}$ is not allowed to float.
		Port 0
50-57	P0.7-P0.0	8-bit binary I/O port Pin — Alternative function P0.7-P0.0 — Multiplexed low-order address and data bus of external memory (AD7-AD0)
58	AV_{ref-}	**LOW end** of analogue to digital conversion reference resistor
59	AV_{ref+}	**High end** of analogue to digital conversion reference resistor
60	AV_{SS}	**Analogue** ground
61	AV_{DD}	**Analogue** power supply
62-68 1	P5.7-P5.0	8-bit input port Pin — Alternative function P5.0-P5.7 — Eight input channels to ADC (ADC0-ADC7)

At power-on, the voltage on any pin at any time must not be higher or lower than V_{CC} + 0,5 V or V_{SS} − 0,5 V respectively.

FUNCTIONAL DESCRIPTION

General

The PCB83C552 is a stand-alone high-performance microcontroller designed for use in real-time applications such as instrumentation, industrial control and specific automotive control applications.

The device provides in addition to the 80C51 standard functions, a number of dedicated hardware functions for these applications.

The PCB83C552 is a control-oriented CPU with on-chip program and data memory. It can be extended with external program memory up to 64 K bytes. For systems requiring extra capability, the PCB80C552 can be expanded using standard memories and peripherals.

The PCB83C552 has two software selectable modes of reduced activity for further power reduction – Idle and Power Down. The Idle mode freezes the CPU while allowing the RAM, timers, serial port and interrupt system to continue functioning. The Power Down mode saves the RAM contents but freezes the oscillator causing all other chip functions to be inoperative.

Memory organisation

The central processing unit (CPU) manipulates operands in three memory spaces; these are the 64 K-byte external data memory, 256-byte internal data memory and the 64 K-byte internal and external program memory. The internal data memory address space is sub-divided into the 256-byte internal data RAM and 128-byte Special Function Register (SFR) address spaces, as shown in Fig. 4. Figures 5(a) and (b) show the Special Function Register memory map. Internal RAM locations 0-127 are directly and indirectly addressable. Internal RAM locations 128-256 are only indirectly addressable as internal data RAM. The special function register locations 128-255 are only directly addressable.

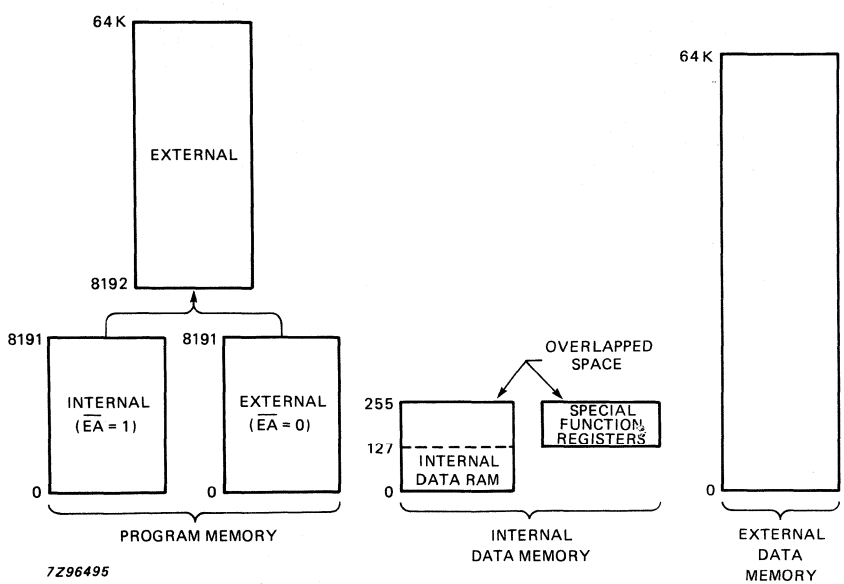


Fig. 4 Memory map.

FUNCTIONAL DESCRIPTION (continued)

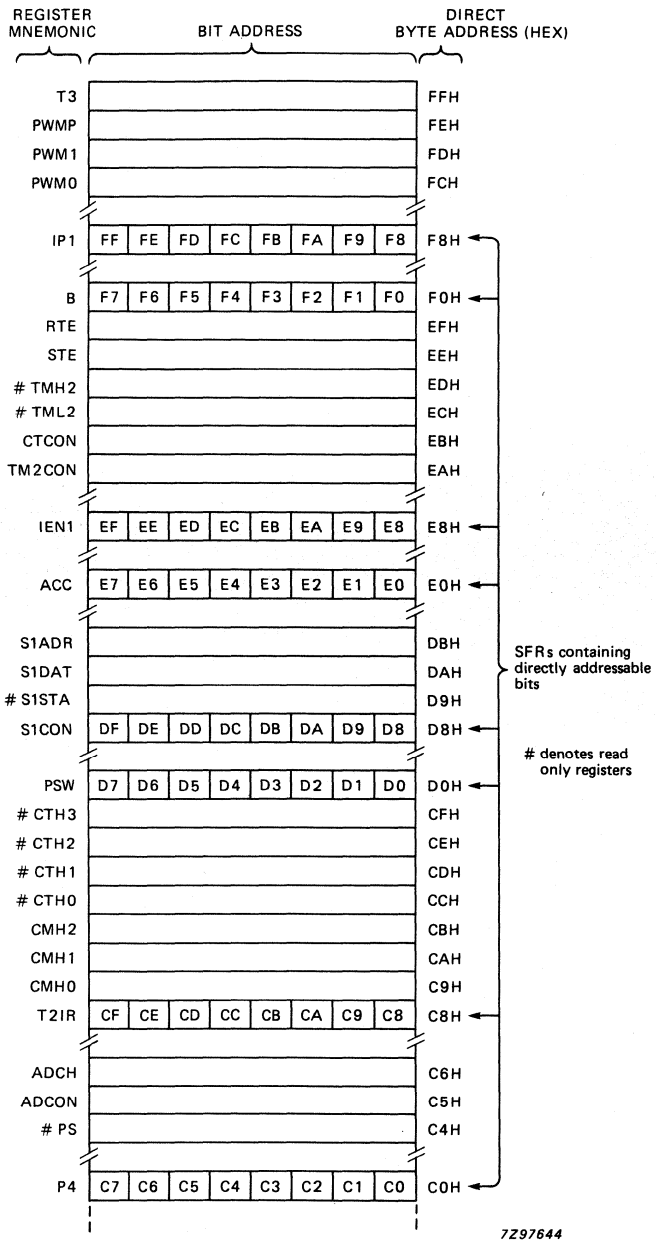


Fig. 5a.

Figures 5a and 5b show the special function registers memory map.

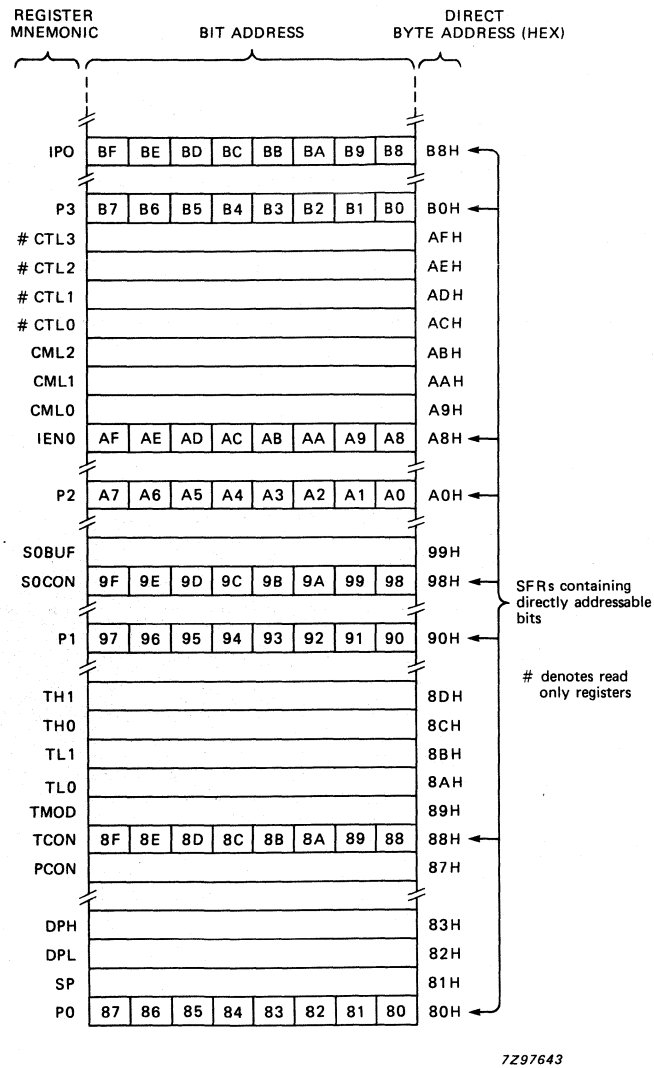


Fig. 5b.

Figures 5a and 5b show the special function registers memory map.

FUNCTIONAL DESCRIPTION (continued)

The internal data RAM contains four register banks (each with eight registers), 128 addressable bits, and the stack. The stack depth is limited by the available internal data RAM and its location is determined by the 8-bit stack pointer. All registers except the program counter and the four 8-register banks reside in the special function register address space. These memory mapped registers include arithmetic registers, pointers, I/O ports, interrupt system registers, ADC and PWM registers, timers and serial port registers. There are 128 addressable bit locations in the SFR address space.

The PCB83C552 contains 256 bytes of internal data RAM and 57 special function registers. It provides a non-paged program memory address space to accommodate relocatable code. Conditional branches are performed relative to the program counter. The register-indirect jump permits branching relative to a 16-bit base register with an offset provided by an 8-bit index register. 16-bit jumps and calls permit branching to any location in the contiguous 64 K program memory address space.

Addressing

The PCB83C552 has five methods for addressing source operands:

- Register
- Direct
- Register-Indirect
- Immediate
- Base-Register plus Index-Register-Indirect

The first three methods can be used for addressing destination operands. Most instructions have a "destination/source" field that specifies the data type, addressing methods and operands involved. For operations other than MOVs, the destination operand is also a source operand.

Access to memory addressing is as follows:

- Registers in one of the four 8-register banks through Register, Direct, or Register-Indirect
- 256 bytes of internal data RAM through Direct or Register-Indirect. Bytes 0-127 may be addressed directly/indirectly. Bytes 128-255 share their address locations with the SFR registers and so may only be addressed indirectly as data RAM
- Special function registers through Direct at address locations 128-255
- External data memory through Register-Indirect
- Program memory look-up tables through Base-Register-plus Index-Register-Indirect

Instruction set

The PCB83C552 uses a powerful instruction set to allow expansion of on-chip CPU peripherals and to optimize byte efficiency and execution speed. Assigned opcodes add new high-power operations and permit new addressing modes. The instruction set consists of 49 single-byte, 45 two-byte and 17 three-byte instructions. When using a 12 MHz oscillator, 64 instructions execute in 1 μs and 45 instructions execute in 2 μs. Multiply and divide instructions execute in 4 μs.

I/O facilities

The PCB83C552 has six 8-bit ports. Ports 0 - 3 are the same as in the 80C51, with the exception of the additional functions of port 1. The parallel I/O function of port 4 is equal to that of ports 1, 2 and 3. Port 5 has a parallel input port function, but has no function as an output port. Port lines P1.7 and P1.6 may be selected as the SDA and SCL lines of serial port SIO1 (I^2C). Because the I^2C bus may be active while the device is disconnected from V_{DD}, these pins are provided with open drain drivers.

N.B. Therefore these pins do not have pull-up devices when used as ports.

Ports 0, 1, 2, 3, 4 and 5 perform the following alternative functions:

- Port 0; provides the multiplexed low-order address and data bus used for expanding the PCB83C552 with standard memories and peripherals.
- Port 1; port 1 is used for a number of special functions;
  - 4 capture inputs
  - external counter input
  - external counter reset input
  - SCL and SDA for the I^2C interface

Bits whose alternate function is not used may be used as normal bidirectional I/O pins.

- Port 2; provides the high-order address bus when expanding the PCB83C552 with external program memory and/or external data memory.
- Port 3; pins can be configured individually to provide:
  - external interrupt request inputs
  - counter inputs
  - serial port receiver input and transmitter output
  - control signals to READ and WRITE external data memory

The generation or use of a Port 3 pin as an alternative function is carried out automatically by the PCB83C552 provided the pin is loaded with a HIGH content.

- Port 4; can be configured to provide signals indicating a match between timer counter T2 and its compare registers.
- Port 5; may be used in conjunction with the A/D converter interface. Unused analogue inputs can be used as digital inputs. As port 5 lines may be used as input to the A/D converter, these digital inputs have an inherent hysteresis to prevent the input logic from drawing too much current from the power lines when driven by analogue signals.

All ports are bidirectional with the exception of port 5 which is an input port.

DEVELOPMENT DATA

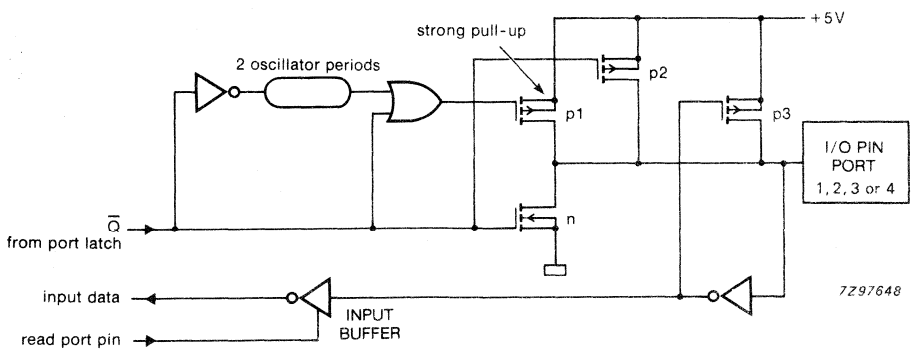


Fig. 6 I/O buffers in the PCB83C552 (Ports 1, 2, 3 and 4).

FUNCTIONAL DESCRIPTION (continued)

In addition to the standard 8-bit ports, the I/O facilities of the PCB83C552 also include a number of specialist I/O lines:

Pulse width modulated outputs

Two pulse width modulated output channels are provided with the PCB83C552. These channels output pulses of programmable length and interval. The interval length is defined by an 8-bit prescaler PWMP which generates the clock for the counter. Both the prescaler and counter are common to both PWM channels. The 8-bit counter counts modulo 255 i.e. from 0 to 254 inclusive. The value of the 8-bit counter is compared to the contents of two registers; PWM0 and PWM1. Provided the contents of either of these registers is greater than the counter value, the output of PWM0N or PWM1N is set LOW. If the contents of these registers are equal to, or less than the counter value, the output will be HIGH. The pulse-width-ratio is therefore defined by the contents of the registers PWM0 and PWM1. The pulse-width-ratio is in the range of 0 to 255/255 and may be programmed in increments of 1/255.

The repetition frequency f_{PWM}, at the PWMnN outputs is given by:

$$f_{PWM} = \frac{f_{osc}}{(2 + 2(PWMP)) \times 255}$$

This gives a repetition frequency range of 92 Hz to 23,5 kHz (f_{osc} = 12 MHz).

By loading the PWM registers with either OOH or FFH, the PWM outputs can be retained at a constant HIGH or LOW level respectively. When loading FFH to the PWM registers, the 8-bit counter will never actually reach this value.

Both PWMnN output pins are driven by push-pull drivers, and are not shared with any other function.

Prescaler frequency control register PWMP

PWMP	7	6	5	4	3	2	1	0
	MSB							LSB

Bit	Function
PWMP.0-7	Prescaler division factor = (PWMP) + 1

Pulse width registers PWM0 and PWM1

PWM0 PWM1	7	6	5	4	3	2	1	0
	MSB							LSB

Bit	Function
PWM0.0-7 PWM1.0-7	Low/high ratio of PWMnN signals = $\frac{(PWMn)}{255 - (PWMn)}$

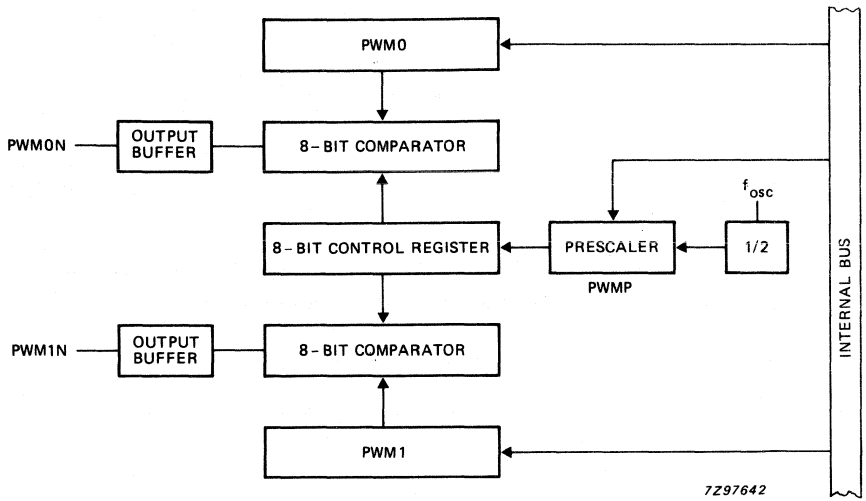


Fig. 7 Functional diagram of pulse width modulated outputs.

DEVELOPMENT DATA

Analogue input pins

The analogue input circuitry consists of an 8-input analogue multiplexer and an A/D converter with 10 bit resolution. The analogue reference voltage and analogue power supplies are connected via separate input pins. The conversion time takes 88 machine cycles i.e. 88 μs at 12 MHz oscillator frequency.

The A/D converter is controlled using the ADCON control register. Input channels are selected by the analogue multiplexer, care of ADCON register bits 0-2.

A/D control register ADCON

	ADC1	ADC0	ADEX	ADCI	ADCS	AADR2	AADR1	AADR0
ADCON	7	6	5	4	3	2	1	0

Bit	Function
ADCON.7 - ADC1	Bit 1 of ADC converted value
ADCON.6 - ADC0	Bit 0 of ADC converted value
ADCON.5 - ADEX	Enable external start of conversion by STADC
	0 = Conversion cannot be started externally by STADC
	1 = Conversion can be started externally by STADC
ADCON.4 - ADCI	ADC interrupt flag: This flag is set when an A/D conversion result is ready to be read. An interrupt is invoked if this is enabled. The flag must be cleared by software. It cannot be set by software.

FUNCTIONAL DESCRIPTION (continued)

ADCON.3 - ADCS — ADC start and status: Setting this bit starts an A/D conversion. It may be set by software or by the external signal STADC. The ADC logic ensures that this signal is HIGH while the ADC is busy. On completion of the conversion, ADCS is reset at the same time the interrupt flag ADCI is set. ADCS can not be reset by software.

ADCI	ADCS	OPERATION
0	0	ADC not busy, a conversion can be started
0	1	ADC busy, start of a new conversion is blocked
1	0	Conversion completed, start of a new conversion is blocked
1	1	Not possible

ADCON.2 - AADR2
ADCON.1 - AADR1
ADCON.0 - AADR0
} Analogue input select: This binary coded address selects one of the eight analogue port bits of P5 to be input to the converter. It can only be changed when ADCI and ADCS are both LOW.

The completion of the 10-bit A/D conversion is flagged by ADCI in the ADCON register and the result is stored in special function register ADCH (upper 8 bits) and the 2 LSBs in register ADCON.

An A/D conversion in progress is unaffected by an external or software ADC start. The result of a completed conversion remains unaffected provided ADSI = logic 1. During as ADCS = logic 1 or ADCI = logic 1, a new ADC START will be blocked and consequently lost. An A/D conversion already in progress is aborted when the IDLE or Power Down mode is entered. The result of a completed conversion (ADCI = logic 1) remains unaffected when entering the IDLE mode.

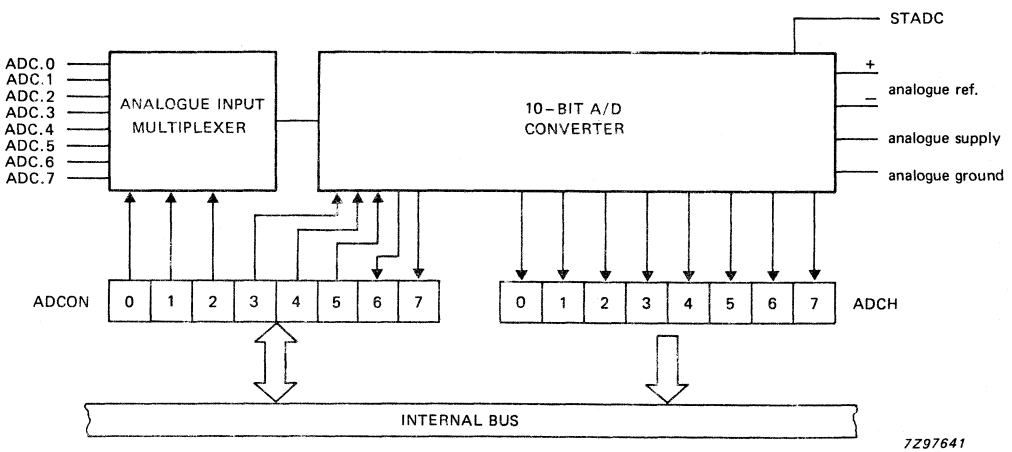


Fig. 8 Functional diagram of analogue input.

Timer/event counters

The PCB83C552 contains three 16-bit timer/event counters; Timer 0, Timer 1 and Timer T2 and one 8-bit timer, T3. Timer 0 and Timer 1 may be programmed to carry out the following functions:

- Measure time intervals and pulse durations
- Count events
- Generate interrupt requests

Timer 0 and Timer 1 can be programmed independently to operate in three modes:

- Mode 0; 8-bit timer or 8-bit counter each with divide by 32 prescaler
- Mode 1; 16-bit time-interval or event counter
- Mode 2; 8-bit time-interval or event counter with automatic reload upon overflow

Timer 0 can be programmed to operate in an additional mode as follows:

- Mode 3; one 8-bit time-interval or event counter and one 8-bit time-interval counter

When Timer 0 is in Mode 3, Timer 1 can be programmed to operate in Modes 0, 1 or 2 but cannot set an interrupt request flag or generate an interrupt. However the overflow from Timer 1 can be used to pulse the serial Port transmission-rate generator.

The frequency handling range of these counters with a 12 MHz crystal is as follows:

- In the timer function, the timer is incremented at a frequency of 1 MHz, that is, a division by 12 of the oscillator frequency
- 0 Hz to an upper limit of 0,5 MHz when programmed for external inputs

Both internal and external inputs can be gated to the counter by a second external source for directly measuring pulse durations.

The counters are started and stopped under software control. Each one sets its interrupt request flag when it overflows from all logic 1s to all logic 0s (or automatic reload value), with the exception of mode 3 as previously described.

Timer T2

Timer T2 is a 16 bit register which has, coupled to it, capture and compare facilities. The operational diagram is shown in Fig. 9.

The 16 bit timer/counter is clocked via a prescaler with a programmable division factor of 1, 2, 4 or 8. The input of the prescaler is clocked with 1/12 of the oscillator frequency, or with positive edges on the T2 input, or it is switched to the off position. This prescaler is cleared if its division factor or its input source is changed, or if the timer/counter is reset. T2 is readable 'on the fly', but posseses no extra read latches, this means that software precautions have to be taken against misinterpretation in overflow from least to most significant byte during read. T2 is not loadable and is reset by the RST signal or at the positive edge of the input signal RT2, is enabled. In the IDLE mode the timer/counter and prescaler are reset and halted.

T2 is connected to four 16-bit capture registers; (CT0), CT1, CT2 and CT3). These registers are loaded with the contents of T2 and an interrupt requested upon receipt of the input signals CT0I, CT1I, CT2I or CT3I. These input signals are shared with port 1. Using the capture register CTCON, an interrupt/capture on a positive, negative edge or both may be triggered.

FUNCTIONAL DESCRIPTION (continued)

The contents of the compare registers CM0, CM1 and CM2 are continually compared with the counter value of Timer T2. When a match is found an interrupt may be invoked. Using the match signal of CM0, the controller sets bits 0-5 of port 4, if the corresponding bits of the set enable register STE are logic 1.

Considering a match with CM1, if the corresponding bits of the reset/toggle enable register RTE are logic 1, then the controller will use the match signal to reset bits 0-5 of port 4. Bits 6 and 7 of port 4 may be 'toggled' by the signal that indicates a match between Timer T2 and CM2, again, if the corresponding bits of RTE are logic 1. CM0, CM1 and CM2 are reset by the RST signal.

Port 4 can be read and written by software without affecting the toggle, set and reset signals. At byte overflow of the least significant byte, or at a 16-bit overflow of the timer/counter, an interrupt sharing the same interrupt vector is requested. Either or both of these overflows can be programmed to request an interrupt.

All interupt flags must be reset by software.

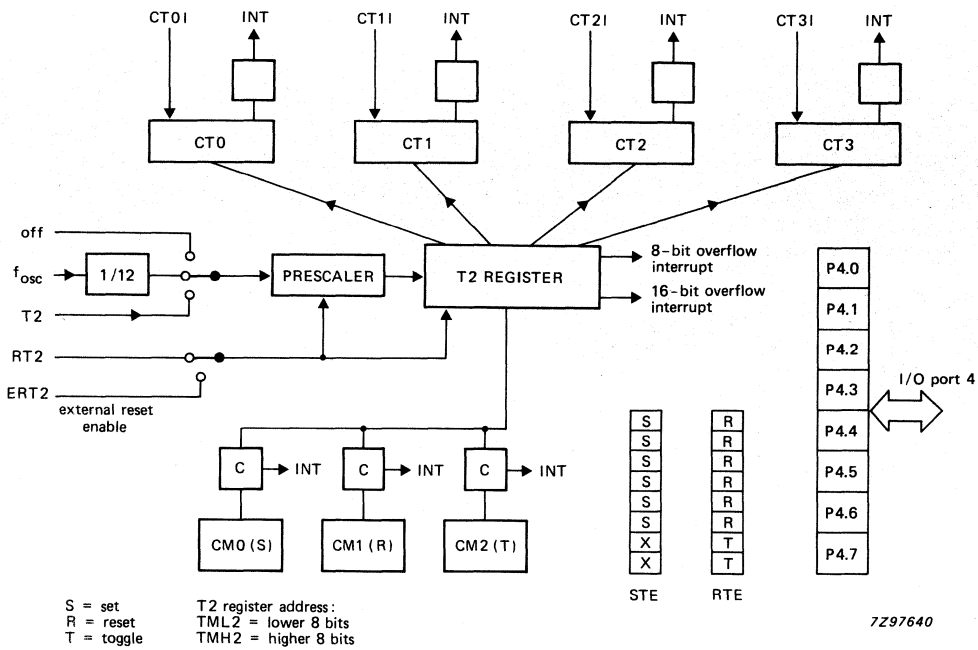


Fig. 9 Block diagram of Timer T2 configuration.

Capture control register CTCON

CTCON

7	6	5	4	3	2	1	0
MSB							LSB

Bit	Capture/interrupt at:
CTCON.0	CT0I positive edge
CTCON.1	CT0I negative edge
CTCON.2	CT1I positive edge
CTCON.3	CT1I negative edge
CTCON.4	CT2I positive edge
CTCON.5	CT2I negative edge
CTCON.6	CT3I positive edge
CTCON.7	CT3I negative edge

Counter control register TM2CON

TM2CON

7	6	5	4	3	2	1	0
MSB							LSB

Bit		Function
TM2CON.0	"x" MODE SELECT BIT	T2CON.0/T2CON.1
TM2CON.1	"x" MODE SELECT BIT	00 = Timer T2 is halted 01 = Timer T2 input source = $\frac{f_{osc}}{12}$ 10 = Test mode, do not use 11 = Timer T2 input source = pin T2
TM2CON.2	"x" PRESCALER BIT	T2CON.3/T2CON.2
TM2CON.3	"x" PRESCALER BIT	00 = DIVISION by 1 01 = DIVISION by 2 10 = DIVISION by 4 11 = DIVISION by 8
TM2CON.4	T2 Byte overflow INT flag	
TM2CON.5	ERT2 switch	
TM2CON.6	Select byte overflow interrupt	
TM2CON.7	Select 16 bit overflow interrupt	

Timer interrupt enable register IEN1

IEN1

7	6	5	4	3	2	1	0
MSB							LSB

Bit		Function
IEN1.7	"0/1"	T2 overflow interrupts disabled/enabled
IEN1.6	"0/1"	interrupt CM2 disabled/enabled
IEN1.5	"0/1"	interrupt CM1 disabled/enabled
IEN1.4	"0/1"	interrupt CM0 disabled/enabled
IEN1.3	"0/1"	interrupt CT3 disabled/enabled
IEN1.2	"0/1"	interrupt CT2 disabled/enabled
IEN1.1	"0/1"	interrupt CT1 disabled/enabled
IEN1.0	"0/1"	interrupt CT0 disabled/enabled

FUNCTIONAL DESCRIPTION (continued)

Timer interrupt flag register TM2IR

TM2IR

7	6	5	4	3	2	1	0
MSB							LSB

Bit	Function
TM2IR.7	T2 16-bit overflow interrupt flag
TM2IR.6	CM2 interrupt flag
TM2IR.5	CM1 interrupt flag
TM2IR.4	CM0 interrupt flag
TM2IR.3	CT3 interrupt flag
TM2IR.2	CT2 interrupt flag
TM2IR.1	CT1 interrupt flag
TM2IR.0	CT0 interrupt flag

Interrupt priority register IP1 is used to determine the timer interrupt priority.

The set enable register STE.

STE

7	6	5	4	3	2	1	0
MSB							LSB

Bit		Function
STE.0	"1"	:= P40 is set at match CM0 and T2
STE.1	"1"	:= P41 is set at match CM0 and T2
STE.2	"1"	:= P42 is set at match CM0 and T2
STE.3	"1"	:= P43 is set at match CM0 and T2
STE.4	"1"	:= P44 is set at match CM0 and T2
STE.5	"1"	:= P45 is set at match CM0 and T2
STE.6		:= not used
STE.7		:= not used
STE.n	"0"	:= P4n is not affected by a match

The reset/toggle enable register RTE.

RTE

7	6	5	4	3	2	1	0
MSB							LSB

Bit		Function
RTE.0	"1"	:= P40 is reset at match CM1 and T2
RTE.1	"1"	:= P41 is reset at match CM1 and T2
RTE.2	"1"	:= P42 is reset at match CM1 and T2
RTE.3	"1"	:= P43 is reset at match CM1 and T2
RTE.4	"1"	:= P44 is reset at match CM1 and T2
RTE.5	"1"	:= P45 is reset at match CM1 and T2
RTE.6	"1"	:= P46 toggles at match CM2 and T2
RTE.7	"1"	:= P47 toggles at match CM2 and T2
RTE.n	"0"	:= P4n is not affected by a match

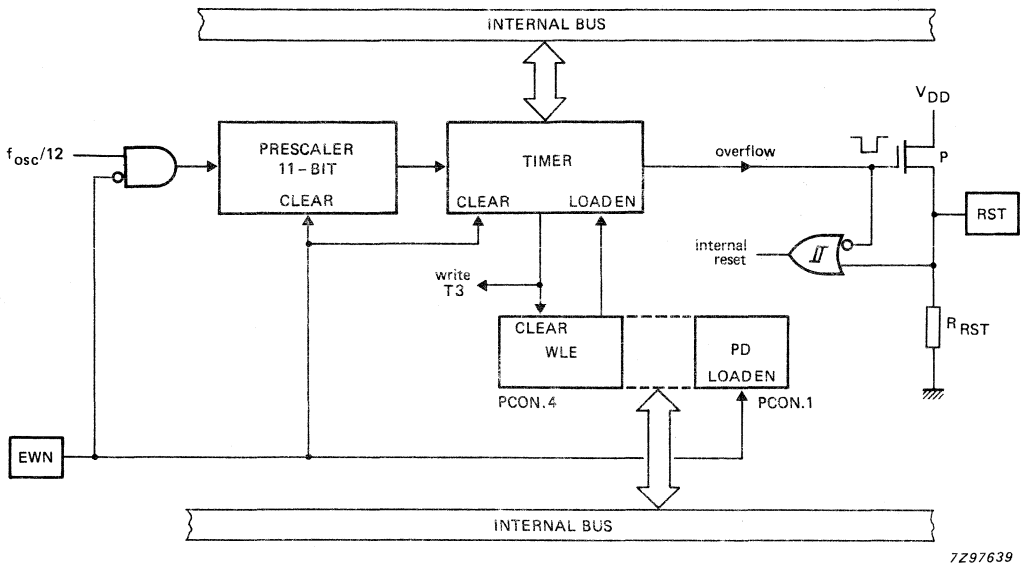


Fig. 10 Functional diagram of T3 watchdog timer.

T3 The watchdog timer (see Fig. 10)

In addition to Timer T2 and the standard timers, a watchdog timer consisting of an 11-bit prescaler and an 8-bit timer are also incorporated.

The timer is incremented every 2 ms, derived from the oscillator frequency of 12 MHz by the following:

$$f_{timer} = \frac{f_{osc}}{12 \times 2048}$$

When a timer overflow occurs, the microcontroller is reset and a reset output pulse is generated at pin RST. To prevent a system reset the timer must be reloaded in time by the application software. If the processor suffers a hardware/software malfunction, the software will fail to reload the timer. This failure will produce a reset upon overflow thus preventing the processor running out of control.

The watchdog timer can only be reloaded if the condition flag WLE = PCON.4 has been previously set by software.

At the moment the counter is loaded the condition flag is automatically cleared.

The time interval between the timer's reloading and occurance of a reset, is dependent upon the reloaded value. This may range from 2 ms to 0,5 s at an oscillator frequency of 12 MHz.

In the IDLE state the watchdog timer and reset circuitry remain active.

The watchdog timer is controlled by the watchdog enable pin (EWN). A logic 0 enables the watchdog timer and disables the Power Down mode. A logic 1 disables and resets the watchdog timer and enables the power-down mode.

DEVELOPMENT DATA

FUNCTIONAL DESCRIPTION (continued)

Serial I/O (see Fig. 11)

The PCB83C552 is equipped with two independent serial ports. SIO0 is the full duplex UART port and is identical to the serial port of the PCB80C51.

Serial port SIO1 supports the I^2C bus, the function of which is controlled by the S1CON register. S1STA is the status register whose contents may also be used as a vector to various service routines. S1DAT is the data shift register and S1ADR the slave address register.

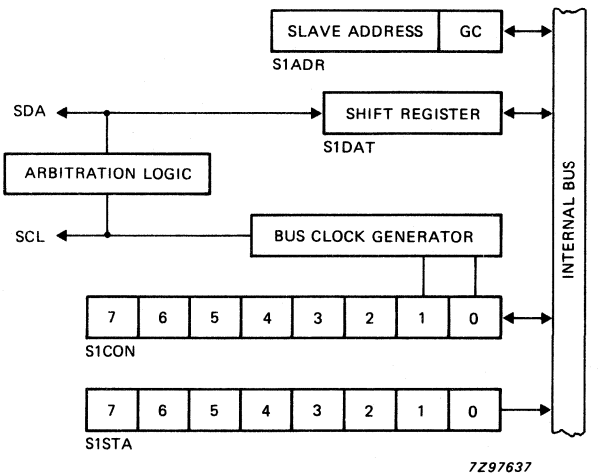


Fig. 11 Block diagram of I^2C serial I/O.

The I^2C serial I/O has complete autonomy in byte handling and operates in 4 modes:

1. Master transmitter
2. Master receiver
3. Slave transmitter
4. Slave receiver

Slave address recognition is performed by hardware.

The I^2C bus consists of two lines; a data line (SDA) and a clock line (SCL). These lines also function as the I/O port lines on P1.7 and P1.6. The system is unique because data transport, clock generation, address recognition and bus control arbitration are all controlled by hardware.

Serial control register S1CON

	X	ENSI	STA	STO	SI	AA	CR1	CR0
S1CON	7	6	5	4	3	2	1	0

Bits CR1 and CR0 determine the clock frequency that is generated in the master mode of operation. Table 1 displays the clock rate when using a 12 MHz crystal.

Table 1 Clock state when using a 12 MHz crystal

CR1/CR0	bit frequency	f_{osc} divided by
0 0	12,5 kHz	960
0 1	100 kHz	120
1 0	200 kHz	60 ($f_{osc} < 6$ MHz meeting I^2C)
1 1	22,5 - 0,5 kHz	96 x (256-reload value Timer 1) (reload value range: 0-254 in mode 2)

AA

Assert acknowledge bit. When this bit is set, an acknowledge is returned after any one of the following conditions:

- Own slave address is received
- General call address is received (S1ADR.0 = logic 1)
- A data byte is received, while the device is programmed to be a master receiver
- A data byte is received, while the device is a selected slave receiver

When this bit is reset, no acknowledge is returned. Consequently, no interrupt is requested when the own address or general call address is received.

DEVELOPMENT DATA

SI

SIO1 interrupt flag. This flag is set, and an interrupt request is generated, after any of the following events occur:

- A START condition is generated in MST mode
- The own slave address has been received during AA = logic 1
- The general call address has been received while S1ADR.0 and AA = logic 1
- A data byte has been received or transmitted in MST mode (even if arbitration is lost)
- A data byte has been received or transmitted as selected slave
- A STOP or START condition is received as selected slave receiver or transmitter

STO

STOP flag. When in master mode, and this bit is set a STOP condition is generated. A STOP condition detected on the I^2C bus clears this bit. This bit may also be set in slave mode in order to recover from an error condition. Then no STOP condition is generated to the I^2C bus, but the hardware releases the SDA and SCL lines and switches to the not selected slave receiver mode. The STOP flag is cleared by the hardware.

STA

START flag. When this bit is set in slave mode, the hardware checks the I^2C bus and generates a START condition if the bus is free or after the bus becomes free. If the device operates in master mode it will generate a repeated START condition.

FUNCTIONAL DESCRIPTION (continued)

ENSI

0 = Serial I/O Disabled and reset. P1.6 and P1.7 I/O port function with open drain

1 = Serial I/O Enabled. Output ports P1.6 and P1.7 must be set to logic 1

Serial status register S1STA (S1STA is a read-only register)

	SC4	SC3	SC2	SC1	SC0	0	0	0
S1STA	7	6	5	4	3	2	1	0

S1STA.3 - S1STA.7 hold a status code. S1STA.0 - S1STA.2 are held LOW. The contents of the status register may be used as a vector to a service routine. This optimizes the response time of the software and consequently that of the I^2C bus.

The following is a list of the status codes in decimal representation. The decimal value corresponds to the value of the upper five bits of the status register.

SLA : 7-bit slave address
R : Read bit
W : Write bit
ACK : Acknowledgement (acknowledge bit = logic 0)
$\overline{ACK}$: Not acknowledgement (acknowledge bit = logic 1)
DATA : 8-bit data byte to or from I^2C bus
MST : Master
SLV : Slave
TRX : Transmitter
REC : Receiver

MST/TRX mode

S1STA value

1 – A START condition has been transmitted
2 – A repeated START condition has been transmitted
3 – SLA and W have been transmitted, ACK has been received
4 – SLA and W have been transmitted, $\overline{ACK}$ received
5 – DATA of S1DAT has been transmitted, ACK received
6 – DATA of S1DAT has been transmitted, $\overline{ACK}$ received
7 – Arbitration lost in SLA and R/W or DATA

MST/REC mode

S1STA value

7 – Arbitration lost while returning $\overline{ACK}$
8 – SLA and R have been transmitted, ACK received
9 – SLA and R have been transmitted, $\overline{ACK}$ received
10 – DATA has been received, ACK returned
11 – DATA has been received, $\overline{ACK}$ returned

SLV/REC mode

S1STA value

12 – Own SLA and W have been received, ACK returned
13 – Arbitration lost in SLA and R/W as MST. Own SLA and W have been received, ACK returned
14 – General CALL has been received, ACK returned
15 – Arbitration lost in SLA and R/W as MST. General call has been received
16 – Previously addressed with own SLA. DATA byte received, ACK returned
17 – Previously addressed with own SLA. DATA byte received $\overline{ACK}$ returned
18 – Previously addressed with general call. DATA byte has been received ACK has been returned
19 – Previously addressed with general call. DATA byte has been received, $\overline{ACK}$ has been returned
20 – A STOP condition or repeated START condition has been received while still addressed as SLV/REC or SLV/TRX.

SLV/TRX mode

S1STA value

21 – Own SLA and R have been received, ACK returned
22 – Arbitration lost in SLA and R/W as MST. Own SLA and R have been received, ACK returned
23 – DATA byte has been transmitted, ACK received
24 – DATA byte has been transmitted, $\overline{ACK}$ received
24 – Last DATA byte has been transmitted (AA = logic 0) ACK received

Miscellaneous

S1STA value

00 – Bus error during MST mode or selected SLV mode, due to an erroneous START or STOP condition

The data shift register S1DAT

	7	6	5	4	3	2	1	0
S1DAT								

This register contains the serial data to be transmitted or data which has just been received. Bit 7 is transmitted or received first; i.e. data is shifted from right to left.

Address register S1ADR

	slave address							GC
S1ADR	7	6	5	4	3	2	1	0

S1ADR.0, GC : 0 = general call address is not recognized
1 = general call address is recognized

S1ADR,7-1 : own slave address

This 8-bit register may be loaded with the 7-bit slave address to which the controller will respond when programmed as a slave receiver/transmitter. The LSB bit (GC) is used to determine whether the general call address is recognized.

FUNCTIONAL DESCRIPTION (continued)

Idle and Power-down operation (see Fig. 12)

Idle mode operation permits the interrupt, serial ports and timer blocks to continue to function while the clock to the CPU is halted. The following functions are also switched off when the processor enters the IDLE mode.

Timer T2	(stopped and reset)
PWM0, PWM1	(reset, output = HIGH)
ADC	(aborted if in progress)

The following functions remain active during IDLE mode. These functions may generate an interrupt or reset and thus end the IDLE mode.

Timer 0, Timer 1
Timer T3
SIO0, SIO1
External interrupt

The Power-down operation freezes the oscillator. The Power-down mode can only be activated by setting the PD bit in the PCON register. The PD bit can only be set if the EWN input is HIGH.

Idle mode

The instruction that sets PCON.0 is the last instruction executed in the normal operating mode before Idle mode is activated. Once in the Idle mode, the CPU status is preserved in its entirety: the Stack Pointer, Program Counter, Program Status Word, Accumulator, RAM and all other registers maintain their data during Idle mode. The status of the external pins during Idle mode is shown in Table 2.

There are two ways to terminate the Idle mode:

Activation of any enabled interrupt will cause PCON.0 to be cleared by hardware terminating Idle mode. The interrupt is serviced, and following return from interrupt instruction RETI, the next instruction to be executed will be the one which follows the instruction that wrote a logic 1 to PCON.0.

The flag bits GF0 and GF1 may be used to determine whether the interrupt was received during normal execution or during the Idle mode. For example, the instruction that writes to PCON.0 can also set or clear one or both flag bits. When Idle mode is terminated by an interrupt, the service routine can examine the status of the flag bits.

The second way of terminating the Idle mode is with an external hardware reset, or an internal reset caused by an overflow of Timer T3. Since the oscillator is still running, the hardware reset is required to be active for two machine cycles (24 oscillator periods) to complete the reset operation.

Power-down mode

The instruction that sets PCON.1 is the last executed prior to going into the Power-down mode. Once in Power-down mode, the oscillator is stopped. Only the contents of the on-chip RAM are preserved. The Special Function Registers are not saved. A hardware reset is the only way of exiting the Power-down mode.

In the Power-down mode, V_{DD} may be reduced to minimize circuit power consumption. The voltage must not be reduced until the Power-down mode is entered, but must be restored before the hardware reset is applied which will free the oscillator. Reset should not be released until the oscillator has restarted and stabilized.

The status of the external pins during Power-down mode is shown in Table 2. If the Power-down mode is activated while in external program memory, the port data that is held in the Special Function Register P2 is restored to Port 2. If the data is a logic 1, the port pin is held HIGH during the Power-down mode by the strong pull-up transistor P1 (see Fig. 6).

Table 2 Status of external pins during Idle and Power-down modes

mode	memory	ALE	$\overline{PSEN}$	Port 0	Port 1	Port 2	Port 3	Port 4	PWM0N/PWM1N
Idle (1)	internal	1	1	port data	port data	port data	port data	port data	HIGH
Idle (1)	external	1	1	floating	port data	address	port data	port data	HIGH
Power-down	internal	0	0	port data	port data	port data	port data	port data	HIGH
Power-down	external	0	0	floating	port data	port data	port data	port data	HIGH

Power control register (PCON)

These special modes are activated by software via the Special Function Register PCON. Its hardware address is 87H. PCON is not bit addressable.

PCON

SMOD	X	X	WLE	GF1	GF0	PD	IDL
7	6	5	4	3	2	1	0

Bit		Definition
SMOD	PCON.7	Double Baud rate bit. When set to logic 1 the baud rate is doubled when the serial port SIO0 is being used in modes 1, 2 or 3
	PCON.6	(reserved)
	PCON.5	(reserved)
WLE	PCON.4	Watchdog load enable; This flag must be set by software prior to loading T3 (watchdog timer). It is cleared by loading T3
GF1	PCON.3	general-purpose flag bit
GF0	PCON.2	general-purpose flag bit
PD	PCON.1	Power-down bit; setting this bit activates power-down mode, can only be set if input EWN is logic 1
IDL	PCON.0	Idle mode bit; setting this bit activates the idle mode operation

If logic 1s are written to PD and IDL at the same time, PD takes precedence. The reset value of PCON is (0XX00000).

DEVELOPMENT DATA

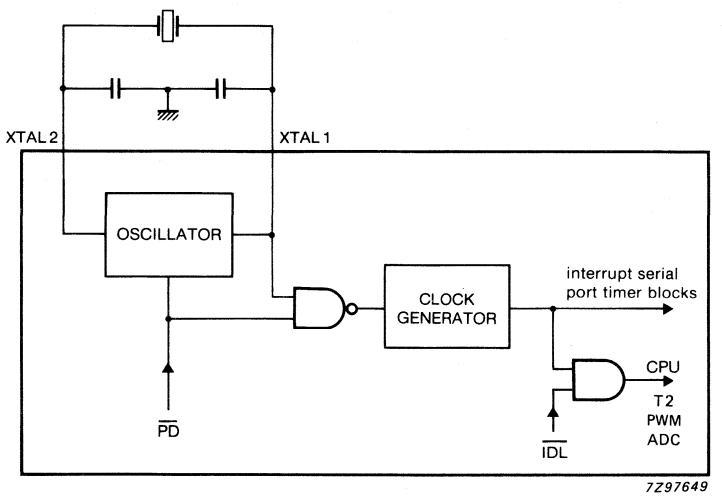


Fig. 12 Internal Idle and Power-down clock configuration.

FUNCTIONAL DESCRIPTION (continued)

Interrupt system (see Fig. 13)

External events and the real-time-driven on-chip peripherals require service by the CPU asynchronous to the execution of any particular section of code. To tie the asynchronous activities of these functions to normal program execution a multiple-source, two-priority-level, nested interrupt system is provided. Interrupt response latency is from 3 μs to 8 μs when using a 12 MHz crystal. The PCB83C552 acknowledges interrupt requests from fifteen sources as follows:

- $\overline{\text{INT0}}$ and $\overline{\text{INT1}}$; externally via pins 26 and 27 respectively
- Timer 0 and Timer 1; from the two internal counters
- Timer T2, 8 separate interrupts; 4 capture interrupts, 3 compare interrupts and an overflow interrupt
- ADC end-of-conversion interrupt
- I^2C serial I/O port interrupt
- UART serial I/O port interrupt

Each interrupt vectors to a separate location in program memory for its service program. Each source can be individually enabled or disabled by a corresponding bit in the IE0 or IE1 register, moreover each interrupt may be programmed to a high or low priority level using a corresponding bit in the IP0 or IP1 register. Also all enabled sources can be globally disabled or enabled. Both external interrupts can be programmed to be level-activated or transition-activated; an active LOW level allows "wire-ORing" of several interrupt sources to one input pin.

DEVELOPMENT DATA

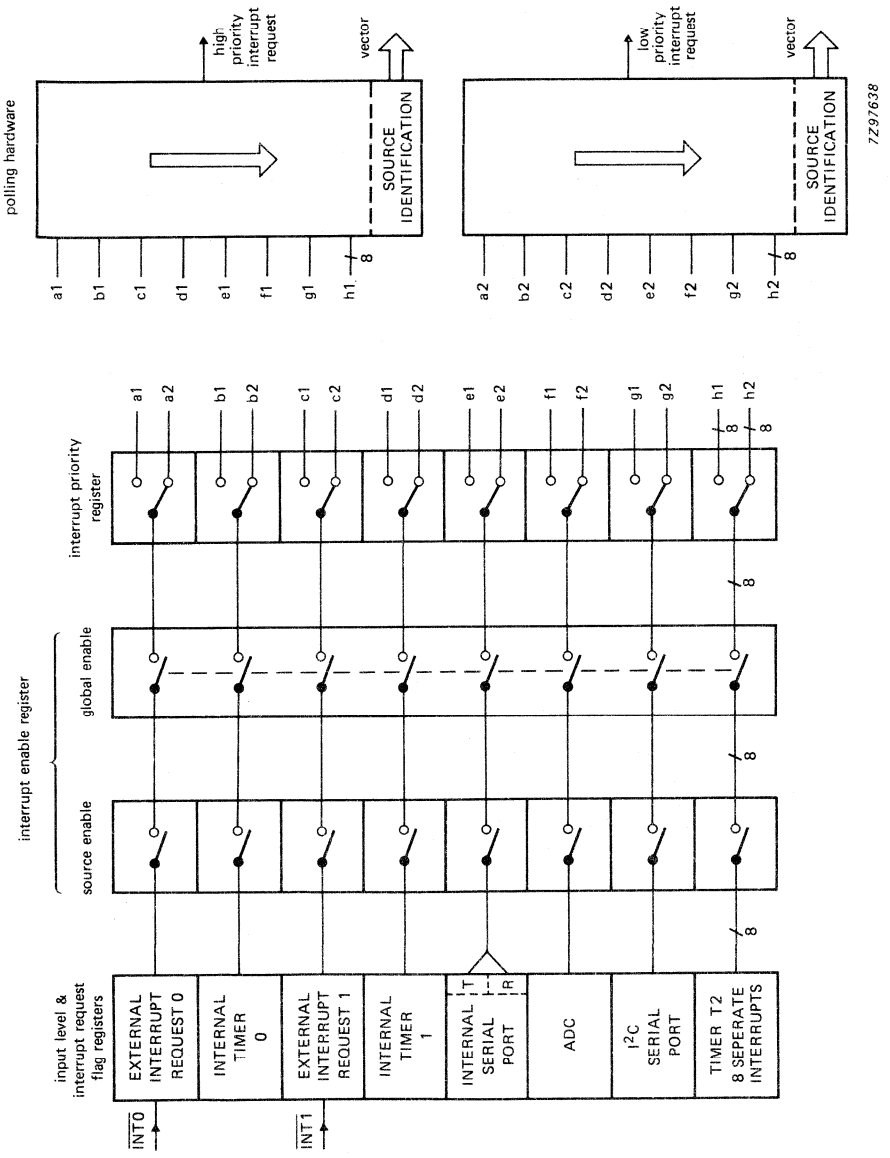


Fig. 13 Interrupt system.

FUNCTIONAL DESCRIPTION (continued)

Interrupt enable registers

IEN0

EA	EAD	ES1	ES0	ET1	EX1	ET0	EX0
7	6	5	4	3	2	1	0

Bit		Function
IEN0.7	EA	General enable/disable control 0 = No interrupt is enabled 1 = Any individually enabled interrupt will be accepted
IEN0.6	EAD	Enable ADC interrupt
IEN0.5	ES1	Enable SIO1 I^2C interrupt
IEN0.4	ES	Enable SIO0 (UART) interrupt
IEN0.3	ET1	Enable Timer 1 interrupt
IEN0.2	EX1	Enable External 1 interrupt
IEN0.1	ET0	Enable Timer 0 interrupt
IEN0.0	EX0	Enable External 0 interrupt

IEN1

ET2	EMC2	EMC1	ECM0	ECT3	ECT2	ECT1	ECT0
7	6	5	4	3	2	1	0

Bit		Function
IEN1.7	ET2	Enable T2 overflow interrupt(s)
IEN1.6	ECM2	Enable T2 comparator 2 interrupt
IEN1.5	ECM1	Enable T2 comparator 1 interrupt
IEN1.4	ECM0	Enable T2 comparator 0 interrupt
IEN1.3	ECT3	Enable T2 capture register 3 interrupt
IEN1.2	ECT2	Enable T2 capture register 2 interrupt
IEN1.1	ECT1	Enable T2 capture register 1 interrupt
IEN1.0	ECT0	Enable T2 capture register 0 interrupt

Interrupt priority registers

IP0

X	PAD	PS1	PS0	PT1	PX1	PT0	PX0
7	6	5	4	3	2	1	0

Bit		Function
IP0.7		Unused
IP0.6	PAD	ADC interrupt priority level
IP0.5	PS1	SIO1 (I^2C) interrupt priority level
IP0.4	PS0	SIO0 (UART) interrupt priority level
IP0.3	PT1	Timer 1 interrupt priority level
IP0.2	PX1	External interrupt 1 priority level
IP0.1	PT0	Timer 0 interrupt priority level
IP0.0	PX0	External interrupt 0 priority level

IP1

PT2	PCM2	PCM1	PCM0	PCT3	PCT2	PCT1	PCT0
7	6	5	4	3	2	1	0

Bit		Function
IP1.7	PT2	T2 overflow interrupt(s) priority level
IP1.6	PCM2	T2 comparator 2 priority interrupt
IP1.5	PCM1	T2 comparator 1 priority interrupt
IP1.4	PCM0	T2 comparator 0 priority interrupt
IP1.3	PCT3	T2 capture register 3 priority interrupt
IP1.2	PCT2	T2 capture register 2 priority interrupt
IP1.1	PCT1	T2 capture register 1 priority interrupt
IP1.0	PCT0	T2 capture register 0 priority interrupt

FUNCTIONAL DESCRIPTION (continued)

Table 3 shows the interrupt vectors. The vector indicates the ROM location where the appropriate interrupt service routine starts.

Table 3 Interrupt vectors

Source		Vector
External 0	X0	0003H
Timer 0 overflow	T0	000BH
External 1	X1	0013H
Timer 1 overflow	T1	001BH
Serial I/O 0 (UART)	S0	0023H
Serial I/O 1 (I^2C)	S1	002BH
T2 capture 0	CT0	0033H
T2 capture 1	CT1	003BH
T2 capture 2	CT2	0043H
T2 capture 3	CT3	004BH
ADC completion	ADC	0053H
T2 compare 0	CM0	005BH
T2 compare 1	CM1	0063H
T2 compare 2	CM2	006BH
T2 overflow	T2	0073H

Interrupt priority

Each interrupt source can be provided by software with two priority levels; logic 1 and logic 0. If both levels are requested simultaneously, the processor will branch to the logic 1 vector. If there are simultaneous requests from sources of the same priority, then interrupts will be serviced in the following order:

X0, S1, ADC, T0, CT0, CM0, X1, CT1, CM1, T1, CT2, CM2, S0, CT3, T2

A logic 0 interrupt routine can be interrupted by a logic 1 interrupt.

Oscillator circuitry

The oscillator circuitry of the PCB83C552 is a single-stage inverting amplifier in a Pierce oscillator configuration. The circuitry between XTAL 1 and XTAL 2 is basically an inverter biased to the transfer point. Either a crystal or ceramic resonator can be used as the feedback element to complete the oscillator circuitry. Both are operated in parallel resonance. XTAL 1 (pin 36) is the high gain amplifier input, and XTAL 2 (pin 35) is the output (see Fig. 14). To drive the PCB83C552 externally, XTAL 1 is driven from an external source and XTAL 2 left open-circuit (see Fig. 15).

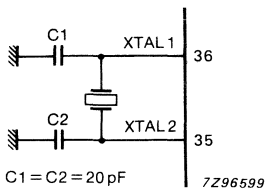


Fig. 14 PCB83C552 oscillator circuit.

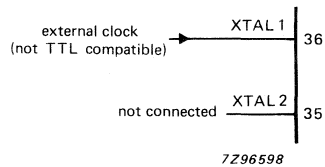


Fig. 15 Driving the PCB83C552 from an external source.

DEVELOPMENT DATA

Reset circuitry (see Fig. 16)

The reset circuitry for the PCB83C552 is connected to the reset pin RST. A Schmitt trigger is used at the input for noise rejection. The output of the Schmitt trigger is sampled by the reset circuitry every machine cycle.

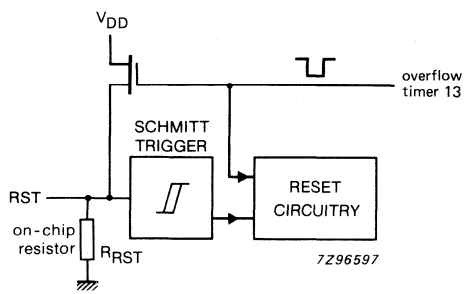


Fig. 16 Reset configuration at RST.

A reset is accomplished by holding the RST pin HIGH for at least two machine cycles (24 oscillator periods), while the oscillator is running. The CPU responds by executing an internal reset. During reset ALE and PSEN output a HIGH level. In order to perform a correct reset, this level must not be affected by external elements.

Also with the PCB83C552, the RST line can be pulled HIGH internally by a pull-up transistor activated by the watchdog timer T3. The length of the output pulse from T3 is 3 machine cycles. A pulse of such short duration is necessary in order to recover from a processor or system fault as fast as possible.

FUNCTIONAL DESCRIPTION (continued)

N.B. It can be seen, that the short reset pulse from Timer T3 cannot discharge the power-on reset capacitor (see Fig. 17). Consequently, when the watchdog timer is also used to reset external devices this capacitor arrangement should not be connected to the RST pin, and an extra circuit should be used to perform the power-on reset operation. It should be remembered that a Timer T3 overflow, if enabled, will force a reset condition to the 83C552 by an internal connection, whether the output RTS is tied LOW or not.

The internal reset is executed during the second cycle in which RST is HIGH and is repeated every cycle until RST goes LOW. It leaves the internal registers as follows:

Register	Content
ACC	0000 0000
ADCON	XX00 0000
ADCH	XXXX XXXX
B	0000 0000
CML0 - CML2	0000 0000
CMH0 - CMH1	0000 0000
CTCON	0000 0000
CTL0 - CTL3	XXXX XXXX
CTH0 - CTH3	XXXX XXXX
DPL	0000 0000
DPH	0000 0000
IEN0	0000 0000
IEN1	0000 0000
IP0	X000 0000
IP1	0000 0000
PCH	0000 0000
PCL	0000 0000
PCON	0XX0 0000
PSW	0000 0000
PWM0	0000 0000
PWM1	0000 0000
PWMP	0000 0000
P0 - P4	1111 1111
P5	XXXX XXXX
RTE	0000 0000
SOBUF	XXXX XXXX
SOCON	0000 0000
S1ADR	0000 0000
S1CON	X000 0000
S1DAT	0000 0000
S1STA	1111 1000
SP	0000 0111
STE	XX00 0000
TCON	0000 0000
TH0, TH1	0000 0000
TMH2	0000 0000
TL0, TL1	0000 0000
TML2	0000 0000
TMOD	0000 0000
TM2CON	0000 0000
TM2IR	0111 0000
T3	0000 0000

The internal RAM is not affected by reset. When V_{DD} is turned on, the RAM content is indeterminate.

Power-on reset (see Fig. 17)

When V_{DD} is turned on, and provided its rise-time does not exceed 10 ms, an automatic reset can be obtained by connecting the RST pin to V_{DD} via a 10 μF capacitor. When the power is switched on, the current drawn by RST is the difference between V_{DD} and the capacitor voltage, and decreases from V_{DD} as the capacitor charges through the internal resistor (R_{RST}) to ground. The larger the capacitor, the more slowly V_{RST} decreases. V_{RST} must remain above the lower threshold of the Schmitt trigger long enough to effect a complete reset. The time required is the oscillator start-up time, plus 2 machine cycles.

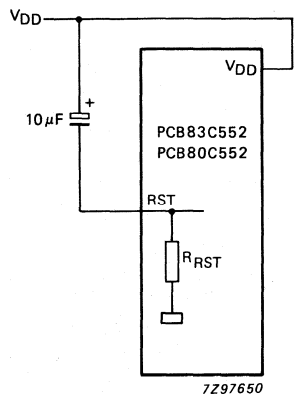


Fig. 17 Power-on reset.

INSTRUCTION SET

Table 4 Instruction set description

mnemonic		description	bytes/ cycles		opcode (hex.)
Arithmetic operations					
ADD	A,Rr	Add register to A	1	1	2*
ADD	A,direct	Add direct byte to A	2	1	25
ADD	A,@Ri	Add indirect RAM to A	1	1	26, 27
ADD	A,#data	Add immediate data to A	2	1	24
ADDC	A,Rr	Add register to A with carry flag	1	1	3*
ADDC	A,direct	Add direct byte to A with carry flag	2	1	35
ADDC	A,@Ri	Add indirect RAM to A with carry flag	1	1	36, 37
ADDC	A,#data	Add immediate data to A with carry flag	2	1	34
SUBB	A,Rr	Subtract register from A with borrow	1	1	9*
SUBB	A,direct	Subtract direct byte from A with borrow	2	1	95
SUBB	A,@Ri	Subtract indirect RAM from A with borrow	1	1	96, 97
SUBB	A,#data	Subtract immediate data from A with borrow	2	1	94
INC	A	Increment A	1	1	04
INC	Rr	Increment register	1	1	0*
INC	direct	Increment direct byte	2	1	05
INC	@Ri	Increment indirect RAM	1	1	06, 07
DEC	A	Decrement A	1	1	14
DEC	Rr	Decrement register	1	1	1*
DEC	direct	Decrement direct byte	2	1	15
DEC	@Ri	Decrement indirect RAM	1	1	16, 17
INC	DPTR	Increment data pointer	1	2	A3
MUL	AB	Multuply A & B	1	4	A4
DIV	AB	Divide A by B	1	4	84
DA	A	Decimal adjust A	1	1	D4

mnemonic		description	bytes/ cycles		opcode (hex.)
Logic operations					
ANL	A,Rr	AND register to A	1	1	5*
ANL	A,direct	AND direct byte to A	2	1	55
ANL	A,@Ri	AND indirect RAM to A	1	1	56, 57
ANL	A,#data	AND immediate data to A	2	1	54
ANL	direct,A	AND A to direct byte	2	1	52
ANL	direct,#data	AND immediate data to direct byte	3	2	53
ORL	A,Rr	OR register to A	1	1	4*
ORL	A,direct	OR direct byte to A	2	1	45
ORL	A,@Ri	OR indirect RAM to A	1	1	46, 47
ORL	A,#data	OR immediate data to A	2	1	44
ORL	direct, A	OR A to direct byte	2	1	42
ORL	direct,#data	OR immediate data to direct byte	3	2	43
XRL	A,Rr	Exclusive-OR register to A	1	1	6*
XRL	A,direct	Exclusive-OR direct byte to A	2	1	65
XRL	A,@Ri	Exclusive-OR indirect RAM to A	1	1	66, 67
XRL	A,#data	Exclusive-OR immediate data to A	2	1	64
XRL	direct,A	Exclusive-OR A to direct byte	2	1	62
XRL	direct,#data	Exclusive-OR immediate data to direct byte	3	2	63
CLR	A	Clear A	1	1	E4
CPL	A	Complement A	1	1	F4
RL	A	Rotate A left	1	1	23
RLC	A	Rotate A left through the carry flag	1	1	33
RR	A	Rotate A right	1	1	03
RRC	A	Rotate A right through the carry flag	1	1	13
SWAP	A	Swap nibbles within A	1	1	C4

INSTRUCTION SET (continued)

mnemonic		description	bytes/ cycles		opcode (hex.)
Data transfer					
MOV	A,Rr	Move register to A	1	1	E*
MOV	A,direct**	Move direct byte to A	2	1	E5
MOV	A,@Ri	Move indirect RAM to A	1	1	E6, E7
MOV	A,#data	Move immediate data to A	2	1	74
MOV	Rr,A	Move A to register	1	1	F*
MOV	Rr,direct	Move direct byte to register	2	2	A*
MOV	Rr,#data	Move immediate data to register	2	1	7*
MOV	direct,A	Move A to direct byte	2	1	F5
MOV	direct,Rr	Move register to direct byte	2	2	8*
MOV	direct,direct	Move direct byte to direct	3	2	85
MOV	direct,@Ri	Move indirect RAM to direct byte	2	2	86, 87
MOV	direct,#data	Move immediate data to direct byte	3	2	75
MOV	@Ri,A	Move A to indirect RAM	1	1	F6, F7
MOV	@Ri,direct	Move direct byte to indirect RAM	2	2	A6, A7
MOV	@Ri,#data	Move immediate data to indirect RAM	2	1	76, 77
MOV	DPTR,#data16	Load data pointer with a 16-bit constant	3	2	90
MOVC	A,@A+DPTR	Move code byte relative to DPTR to A	1	2	93
MOVC	A,@A+PC	Move code byte relative to PC to A	1	2	83
MOVX	A,@Ri	Move external RAM (8-bit address) to A	1	2	E2, E3
MOVX	A,@DPTR	Move external RAM (16-bit address) to A	1	2	E0
MOVX	@Ri,A	Move A to external RAM (8-bit address)	1	2	F2, F3
MOVX	@DPTR,A	Move A to external RAM (16-bit address)	1	2	F0
PUSH	direct	Push direct byte onto stack	2	2	C0
POP	direct	Pop direct byte from stack	2	2	D0
XCH	A,Rr	Exchange register with A	1	1	C*
XCH	A,direct	Exchange direct byte with A	2	1	C5
XCH	A,@Ri	Exchange indirect RAM with A	1	1	C6, C7
XCHD	A,@Ri	Exchange LOW-order digit indirect RAM with A	1	1	D6, D7

** MOV A,ACC is not permitted.

mnemonic		description	bytes/ cycles		opcode (hex.)
Boolean variable manipulation					
CLR	C	Clear carry flag	1	1	C3
CLR	bit	Clear direct bit	2	1	C2
SETB	C	Set carry flag	1	1	D3
SETB	bit	Set direct bit	2	1	D2
CPL	C	Complement carry flag	1	1	B3
CPL	bit	Complement direct bit	2	1	B2
ANL	C,bit	AND direct bit to carry flag	2	2	82
ANL	C,/bit	AND complement of direct bit to carry flag	2	2	B0
ORL	C,bit	OR direct bit ti carry flag	2	2	72
ORL	C,/bit	OR complement of direct bit to carry flag	2	2	A0
MOV	C,bit	Move direct bit to carry flag	2	1	A2
MOV	bit,C	Move carry flag to direct bit	2	2	92
Program and machine control					
ACALL	addr11	Absolute subroutine call	2	2	●1addr
LCALL	addr16	Long subroutine call	3	2	12
RET		Return from subroutine	1	2	22
RET1		Return from interrupt	1	2	32
AJMP	addr11	Absolute jump	2	2	▲1addr
LJMP	addr16	Long jump	3	2	02
SJMP	rel	Short jump (relative address)	2	2	80
JMP	@A+DPTR	Jump indirect relative to the DPTR	1	2	73
JZ	rel	Jump if A is zero	2	2	60
JNZ	rel	Jump if A is not zero	2	2	70
JC	rel	Jump if carry flag is set	2	2	40
JNC	rel	Jump is no carry flag	2	2	50
JB	bit,rel	Jump if direct bit is set	3	2	20
JNB	bit,rel	Jump if direct bit is not set	3	2	30
JBC	bit,rel	Jump if direct bit is set and clear bit	3	2	10
CJNE	A,direct,rel	Compare direct to A and jump if not equal	3	2	B5
CJNE	A,#data,rel	Compare immediate to A and jump if not equal	3	2	B4
CJNE	Rr,#data,rel	Compare immediate to reg. and jump if not equal	3	2	B*
CJNE	@Ri,#data,rel	Compare immediate to ind. and jump if not equal	3	2	B6, B7
DJNZ	Rr,rel	Decrement register and jump if not zero	2	2	D*
DJNZ	direct,rel	Decrement direct and jump if not zero	3	2	D5
NOP		No operation	1	1	00

DEVELOPMENT DATA

Notes to Table 4

Data addressing modes

Rr	Working register R0-R7.
direct	128 internal RAM locations and any special function register (SFR).
@Ri	Indirect internal RAM location addressed by register R0 or R1 of the actual register bank.
#data	8-bit constant included in instruction.
#data16	16-bit constant included as bytes 2 and 3 of instruction.
bit	direct addressed bit in internal RAM or SFR.
addr16	16-bit destination address. Used by LCALL and LJMP. The branch will be anywhere within the 64 K-byte program memory address space.
addr11	11-bit destination address. Used by ACALL and AJMP. The branch will be within the same 2 K-byte page of program memory as the first byte of the following instruction.
rel	Signed (two's complement) 8-bit offset byte. Used by SJMP and all conditional jumps. Range is −128 to + 127 bytes relative to first byte of the following instruction.

Hexadecimal opcode cross-reference to Table 5

* : 8, 9, A, B, C, D, E, F.
● : 11, 31, 51, 71, 91, B1, D1, F1.
▲ : 01, 21, 41, 61, 81, A1, C1, E1.

DEVELOPMENT DATA

Table 5 Instruction map

first hexadecimal character of opcode ↓ — second hexadecimal character of opcode →

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NOP	AJMP page 0	LJMP addr16	RR A	INC A	INC dir	INC @Ri 0	1	INC Rr 0	1	2	3	4	5	6	7
1	JBC bit,addr8	ACALL page 0	LCALL addr16	RRC A	DEC A	DEC dir	DEC @Ri 0	1	DEC Rr 0	1	2	3	4	5	6	7
2	JB bit,addr8	AJMP page 1	RET	RL A	ADD A,#data	ADD A,dir	ADD A,@Ri 0	1	ADD A,Rr 0	1	2	3	4	5	6	7
3	JNB bit,addr8	ACALL page 1	RET1	RLC A	ADDC A,#data	ADDC A,dir	ADDC A,@Ri 0	1	ADDC A,Rr 0	1	2	3	4	5	6	7
4	JC addr8	AJMP page 2	ORL dir,A	ORL dir,#data	ORL A,#data	ORL A,dir	ORL A,@Ri 0	1	ORL A,Rr 0	1	2	3	4	5	6	7
5	JNC addr8	ACALL page 2	ANL dir,A	ANL dir,#data	ANL A,#data	ANL A,dir	ANL A,@Ri 0	1	ANL A,Rr 0	1	2	3	4	5	6	7
6	JZ addr8	AJMP page 3	XRL dir,A	XRL dir,#data	XRL A,#data	XRL A,dir	XRL A,@Ri 0	1	XRL A,Rr 0	1	2	3	4	5	6	7
7	JNZ addr8	ACALL page 3	ORL C,bit	JMP @A+DPTR	MOV A,#data	MOV dir,#data	MOV @Ri,#data 0	1	MOV Rr,#data 0	1	2	3	4	5	6	7
8	SJMP addr8	AJMP page 4	ANL C,bit	MOVC A,@A+PC	DIV AB	MOV dir,dir	MOV dir,@Ri 0	1	MOV dir,Rr 0	1	2	3	4	5	6	7
9	MOV DPTR, #data 16	ACALL page 4	MOV bit,C	MOVC A,@A+DPTR	SUBB A,#data	SUBB A,dir	SUBB A,@Ri 0	1	SUBB A,Rr 0	1	2	3	4	5	6	7
A	ORL C,/bit	AJMP page 5	MOV C,bit	INC DPTR	MUL AB		MOV @Ri,dir 0	1	MOV Rr,dir 0	1	2	3	4	5	6	7
B	ANL C,/bit	ACALL page 5	CPL bit	CPL C	CJNE A, #data, addr8	CJNE A,dir, addr8	CJNE @Ri,#data, addr8 0	1	CJNE Rr,#data, addr8 0	1	2	3	4	5	6	7
C	PUSH dir	AJMP page 6	CLR bit	CLR C	SWAP A	XCH A,dir	XCH A,@Ri 0	1	XCH A,Rr 0	1	2	3	4	5	6	7
D	POP dir	ACALL page 6	SETB bit	SETB C	DA A	DJNZ dir,addr8	XCHD A,@Ri 0	1	DJNZ Rr, addr8 0	1	2	3	4	5	6	7
E	MOVX A,@DPTR	AJMP page 7	MOVX A,@Ri 0	1	CLR A	MOV A,dir	MOV A,@Ri 0	1	MOV A,Rr 0	1	2	3	4	5	6	7
F	MOVX @DPTR,A	ACALL page 7	MOVX @Ri,A 0	1	CPL A	MOV dir,A	MOV @Ri,A 0	1	MOV Rr,A 0	1	2	3	4	5	6	7

RATINGS

Limiting values in accordance with the Absolute Maximum System (IEC 134)

Input voltage on any pin with respect to ground (V_{SS})	V_I	−0,5 to + 7	V
Input, output current	$\pm I_I$, I_O	max. 5	mA
Total power dissipation	P_{tot}	max. 1	W
Storage temperature range	T_{stg}	−65 to + 150	°C
Operating ambient temperature range	T_{amb}	0 to + 70	°C

D.C. CHARACTERISTICS

V_{DD} = 5 V (± 10%); V_{SS} = 0 V; T_{amb} = 0 to + 70 °C; all voltages with respect to V_{SS} unless otherwise specified.

parameter	symbol	min.	max.	unit	conditions
Supply voltage	V_{DD}	4,5	5,5	V	
Supply current					
operating (note 1)	I_{DD}	–	tbf	mA	f_{CLK} = 12 MHz
idle mode (note 2)	I_{DD}	–	tbf	mA	f_{CLK} = 12 MHz
Power-down current	I_{PD}	–	tbf	µA	V_{DD} = 2 V (note 3)
Inputs (see note 6)					
LOW level input voltage (except $\overline{EA}$, P1.6/SCL, P1.7/SDA)	V_{IL}	−0,5	$0,2V_{DD}-0,1$	V	
LOW level input voltage ($\overline{EA}$)	V_{IL1}	−0,5	$0,2V_{DD}-0,3$	V	
LOW level input voltage (P1.6/SCL, P1.7/SDA)	V_{IL2}	−0,5	1,5	V	
HIGH level input voltage (except XTAL 1, RST, P1.6/SCL, P1.7/SDA)	V_{IH}	$0,2V_{DD}+0,9$	$V_{DD}+0,5$	V	
HIGH level input voltage (XTAL 1, RST)	V_{IH1}	$0,7V_{CC}$	$V_{DD}+0,5$	V	
HIGH level input voltage (P1.6/SCL, P1.7/SDA)	V_{IH2}	3,0	$V_{DD}+0,5$	V	
Input current logic 0 (Ports 1, 2, 3 and 4; except P1.6/SCL, P1.7/SDA)	$-I_{IL}$	–	50	µA	V_I = 0,45 V
Input current logic 1 to 0 transition (Ports 1, 2, 3 and 4; except P.1/SCL, P1.7/SDA)	$-I_{TL}$	–	500	µA	V_I = 2 V
Input leakage current (Port 0, $\overline{EA}$, STADC, EWN, P1.6/SCL, P1.7/SDA)	$\pm I_{LI}$	–	10	µA	$0,45\,V < V_I < V_{DD}$

parameter	symbol	min.	max.	unit	conditions
Outputs					
LOW level output voltage (note 4) (Ports 1, 2, 3 and 4; except P1.6/SCL, P1.7/SDA)	V_{OL}	–	0,45	V	I_{OL} = 1,6 mA
LOW level output voltage (note 4) (Port 0, ALE, $\overline{PSEN}$)	V_{OL1}	–	0,45	V	I_{OL} = 3,2 mA
LOW level output voltage (P1.6/SCL, P1.7/SCL)	V_{OL2}	–	0,4	V	I_{OL} = 3,0 mA
HIGH level output voltage (Ports 1, 2, 3 and 4; except P1.6/SCL, P1.7/SDA)	V_{OH}	2,4	–	V	$-I_{OH}$ = 80 μA; V_{DD} = 5 V ± 10%
		$0{,}75V_{DD}$	–	V	$-I_{OH}$ = 30 μA
		$0{,}9V_{DD}$	–	V	$-I_{OH}$ = 10 μA
HIGH level output voltage (note 5) (Port 0 in external Bus mode, ALE, $\overline{PSEN}$)	V_{OH1}	2,4	–	V	$-I_{OH}$ = 400 μA; V_{DD} = 5 V ± 10%
		$0{,}75V_{DD}$	–	V	$-I_{OH}$ = 150 μA
		$0{,}9V_{DD}$	–	V	$-I_{OH}$ = 40 μA
RST pull-down resistor	R_{RST}	40	125	kΩ	
I/O pin capacitance	$C_{I/O}$	–	10	pF	test freq. = 1 MHz; T_{amb} = 25 °C

DEVELOPMENT DATA

D.C. CHARACTERISTICS (continued)

V_{DD} = 5,0 V ± 10%; AV_{SS} = 0 V; V_{REF} = 5,0 V; T_{amb} = −0 °C to + 70 °C unless otherwise specified

parameter	symbol	min.	max.	unit
Analogue inputs				
Analogue input voltage	AV_{IN}	AV_{SS}−0,2	AV_{DD}+0,2	V
Reference voltage	AV_{REF-}	AV_{SS}−0,2	−	V
	AV_{REF+}	−	AV_{DD}+0,2	V
Analogue input capacitance (note 7)	C_{IA}	−	−	pF
Sampling time	t_{ADS}	−	$32t_{CY}$	μs
Conversion time (including sample time)	t_{ADC}	−	$88t_{CY}$	μs
Differential non-linearity $AV_{REF+} = AV_{DD}$ $AV_{REF-} = AV_{SS}$	DL_e	−	± 1,0	LSB
Integral non-linearity $AV_{REF+} = AV_{DD}$ $AV_{REF-} = AV_{SS}$	IL_e	−	tbf	LSB
Offset error	OS_e	−	tbf	mV
Gain error	G_e	−	tbf	%
AV_{REF} supply current (note 8)	AI_s	−	tbf	μA
Internal resistance of analogue source	$R_{ia\,max}$	−	tbf	0
Internal resistance of analogue reference source	$R_{ir\,max}$	−	tbf	0

Notes to the d.c. characteristics

1. The operating supply current is measured with all output pins disconnected; XTAL 1 driven with $t_r = t_f$ = 10 ns, $V_{IL} = V_{SS}$ + 0,5 V, $V_{IH} = V_{DD}$ −0,5 V; XTAL 2 not connected; $\overline{EA}$ = RST = Port 0 = V_{DD}.
2. The idle mode supply current is measured with all output pins disconnected; XTAL 1 driven with $t_r = t_f$ = 10 ns, $V_{IL} = V_{SS}$ + 0,5 V, $V_{IH} = V_{DD}$ −0,5 V; XTAL 2 not connected; $\overline{EA}$ = Port 0 = V_{DD}; RST = V_{SS}.
3. The power down current is measured with all output pins disconnected; XTAL 2 not connected; $\overline{EA}$ = Port 0 = V_{DD}; RST = V_{SS}.
4. Capacitive loading on Port 0 and Port 2 may cause spurious noise pulses to be superimposed on the LOW level output voltage of ALE, Port 1 and Port 3. The noise is due to external Bus capacitance discharging into the Port 0 and Port 2 pins when these pins make a 1-to-0 transition during bus operations. In the most adverse condition (capacitive loading > 100 pF) the noise pulse on ALE line may exceed 0,8 V. In this event it may be required to qualify ALE with a Schmitt trigger, or use an address latch with a Schmitt trigger STROBE input.
5. Capacitive loading on Port 0 and Port 2 may cause the HIGH level output voltage on ALE and $\overline{PSEN}$ to momentarily fall below the 0,9 V_{DD} specification when the address bits are stabilizing.
6. The input threshold voltage of P1.6 and P1.7 (SIO1) meets the I^2C specification, so an input voltage below 1,5 V will be recognized as a logic 0 while an input voltage above 3,0 V will be recognized as a logic 1.
7. The internal resistance of the analogue source must be less than the $R_{ia\,max}$ to ensure full loading of the sample capacitance during sample time.
8. The internal resistance of the analogue reference source must be less than $R_{ir\,max}$.

DEVELOPMENT DATA

A.C. CHARACTERISTICS

V_{DD} = 5 V ± 10%; V_{SS} = 0 V; T_{amb} = 0 to + 70 °C; C_L = 100 pF (Port 0, ALE and $\overline{PSEN}$); C_L = 80 pF (all other outputs); unless otherwise specified (see waveforms Figs 19, 20 and 21)

parameter	symbol	10 MHz		12 MHz		variable clock		unit
		min.	max.	min.	max.	min.	max.	
Program memory								
ALE pulse duration	t_{LL}	160	–	127	–	$2t_{CK}-40$	–	ns
Address set-up time to ALE	t_{AL}	60	–	43	–	$t_{CK}-40$	–	ns
Address hold time after ALE	t_{LA}	65	–	48	–	$t_{CK}-35$	–	ns
Time from ALE to valid instruction input	t_{LIV}	–	300	–	233	–	$4t_{CK}-100$	ns
Time from ALE to control pulse $\overline{PSEN}$	t_{LC}	75	–	58	–	$t_{CK}-25$	–	ns
Control pulse duration $\overline{PSEN}$	t_{CC}	265	–	215	–	$3t_{CK}-35$	–	ns
Time from $\overline{PSEN}$ to valid instruction input	t_{CIV}	–	175	–	125	–	$3t_{CK}-125$	ns
Input instruction hold time after $\overline{PSEN}$	t_{CI}	0	–	0	–	0	–	ns
Input instruction float delay after $\overline{PSEN}$*	t_{CIF}	–	80	–	63	–	$t_{CK}-20$	ns
Address valid after $\overline{PSEN}$*	t_{AC}	92	–	75	–	$t_{CK}-8$	–	ns
Address to valid instruction input	t_{AIV}	–	385	–	302	–	$5t_{CK}-115$	ns
Address float time to $\overline{PSEN}$	t_{AFC}	−12	–	−12	–	0	–	ns

* Interfacing the PCB83C552 to devices with float times up to 75 ns is permitted. This limited bus contention will not cause damage to Port 0 drivers.

A.C. CHARACTERISTICS (continued)

parameter	symbol	10 MHz		12 MHz		variable clock		unit
		min.	max.	min.	max.	min.	max.	
External data memory								
$\overline{RD}$ pulse duration	t_{RR}	500	–	400	–	$6t_{CK}-100$	–	ns
$\overline{WR}$ pulse duration	t_{WW}	500	–	400	–	$6t_{CK}-100$	–	ns
Address set up time to ALE	t_{AL}	–	–	28	–			ns
Address hold time after ALE	t_{LA}	65	–	48	–	$t_{CK}-35$	–	ns
$\overline{RD}$ to valid data input	t_{RD}	–	335	–	250	–	$5t_{CK}-165$	ns
Data hold time after $\overline{RD}$	t_{DR}	0	–	0	–	0	–	ns
Data float delay after $\overline{RD}$	t_{DFR}	–	130	–	97	–	$2t_{CK}-70$	ns
Time from ALE to valid data input	t_{LD}	–	650	–	517	–	$8t_{CK}-150$	ns
Address to valid data input	t_{AD}	–	735	–	585	–	$9t_{CK}-165$	ns
Time from ALE to $\overline{RD}$ or $\overline{WR}$	t_{LW}	250	350	200	300	$3t_{CK}-50$	$3t_{CK}+50$	ns
Time from address to $\overline{RD}$ or $\overline{WR}$	t_{AW}	270	–	203	–	$4t_{CK}-130$	–	ns
Time from $\overline{RD}$ or $\overline{WR}$ HIGH to ALE HIGH	t_{WHLH}	60	140	43	123	$t_{CK}-40$	$t_{CK}+40$	ns
Data valid to $\overline{WR}$ transition	t_{DWX}	40	–	23	–	$t_{CK}-60$	–	ns
Data set-up time before $\overline{WR}$	t_{DW}	550	–	433	–	$7t_{CK}-150$	–	ns
Data hold time after $\overline{WR}$	t_{WD}	50	–	33	–	$t_{CK}-50$	–	ns
Address float delay after $\overline{RD}$	t_{AFR}	–	12	–	12	–	12	ns

Where:

$1/t_{CK}$ = 3,5 to 12 MHz (see Fig. 19 and Table 6).

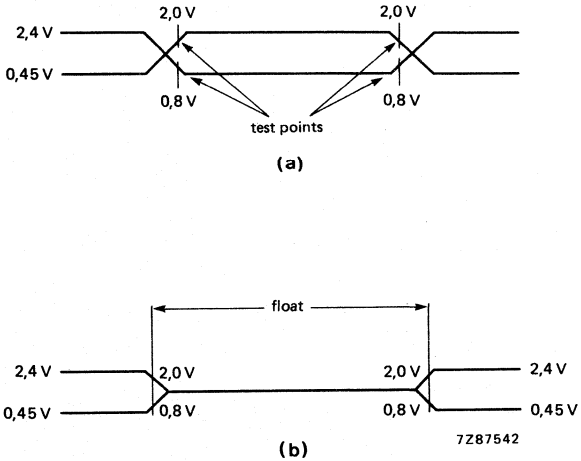


Fig. 18 A.C. testing input, output waveform (a) and float waveform (b).

A.C. testing inputs are driven at 2,4 V for a logic 1 and 0,45 V for a logic 0. Timing measurements are taken at 2,0 V for a logic 1 and 0,8 V for logic 0. The float state is defined as the point at which a Port 0 pin sinks 3,2 mA or sources 400 μA at the voltage test levels.

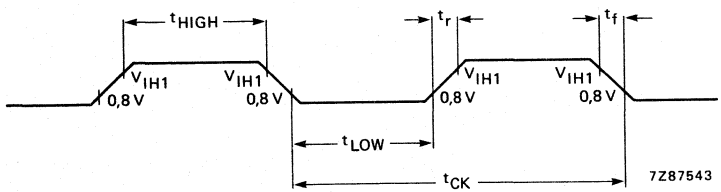


Fig. 19 External clock drive XTAL 1 (see Table 6).

Table 6 External clock drive XTAL 1 (see Fig. 19)

parameter	symbol	variable clock (f = 3,5 to 12 MHz)		unit
		min.	max.	
oscillator clock period	t_{CK}	83,3	286	ns
HIGH time	t_{HIGH}	20	t_{CK}-t_{LOW}	ns
LOW time	t_{LOW}	20	t_{CK}-t_{HIGH}	ns
rise time	t_r	–	20	ns
fall time	t_f	–	20	ns

A.C. CHARACTERISTICS (continued)

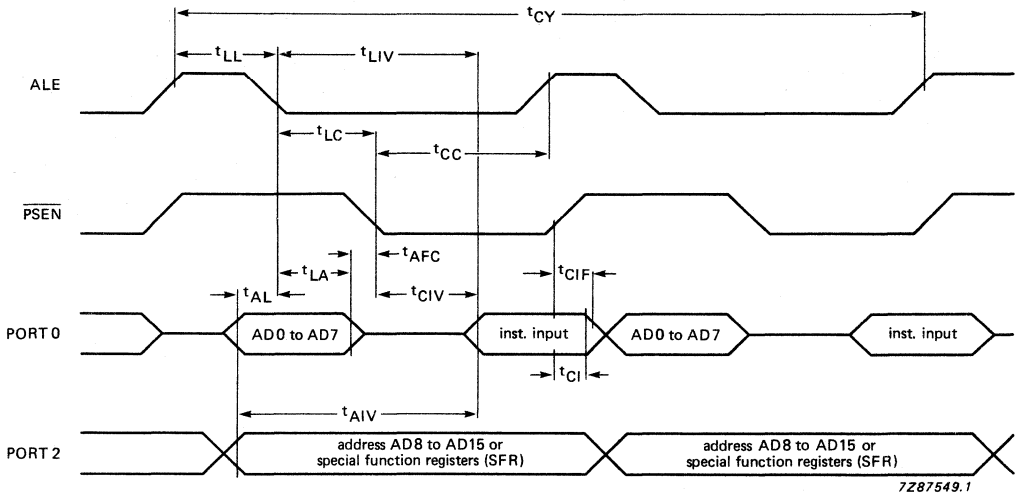


Fig. 20 Read from program memory.

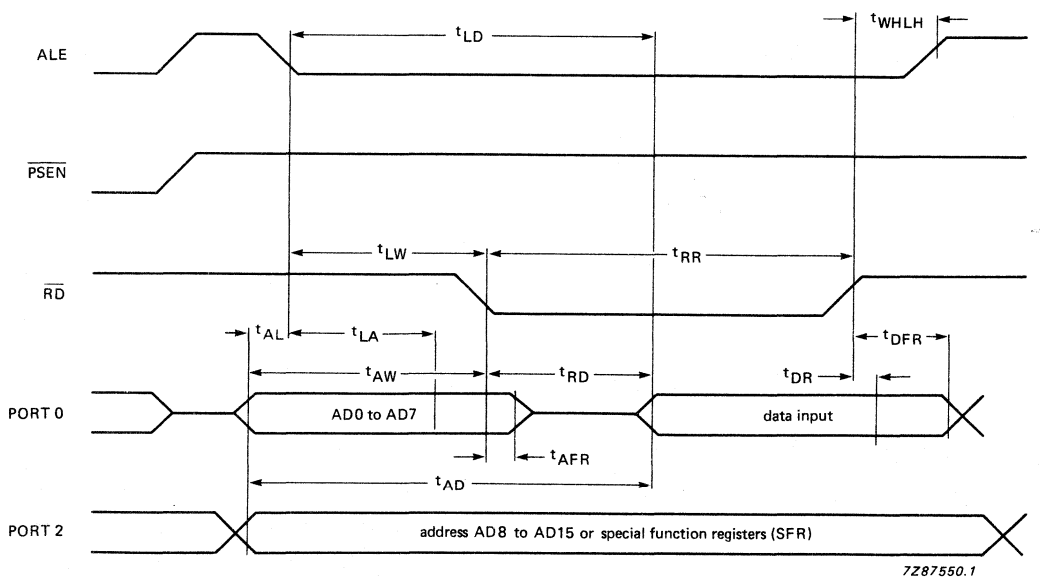


Fig. 21 Read from data memory.

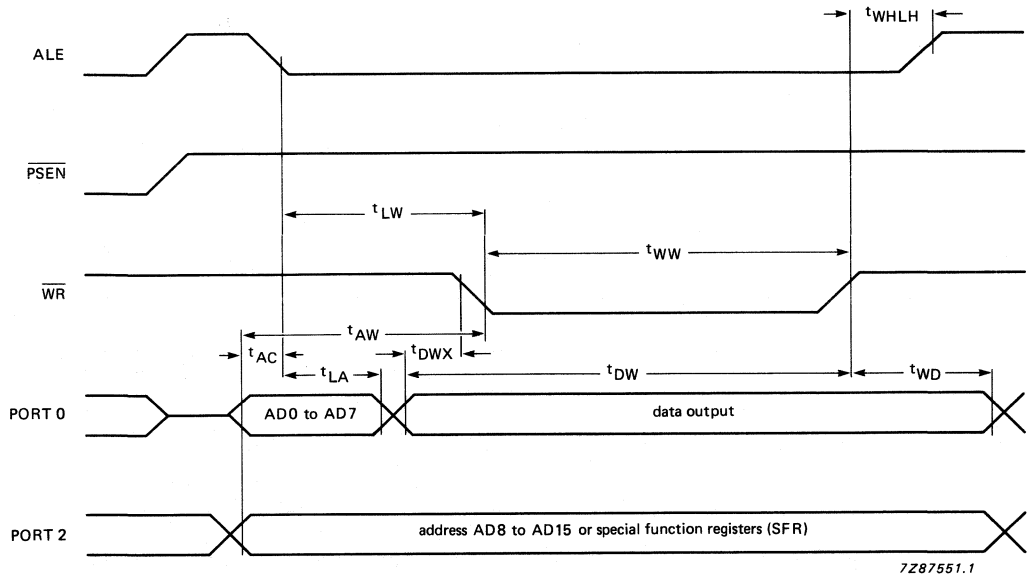


Fig. 22 Write to data memory.

DEVELOPMENT DATA

Purchase of Philip' I²C components conveys a license under the Philips' I²C patent to use the components in the I²C-system provided the system conforms to the I²C specifications defined by Philips.

A.C. CHARACTERISTICS (continued)

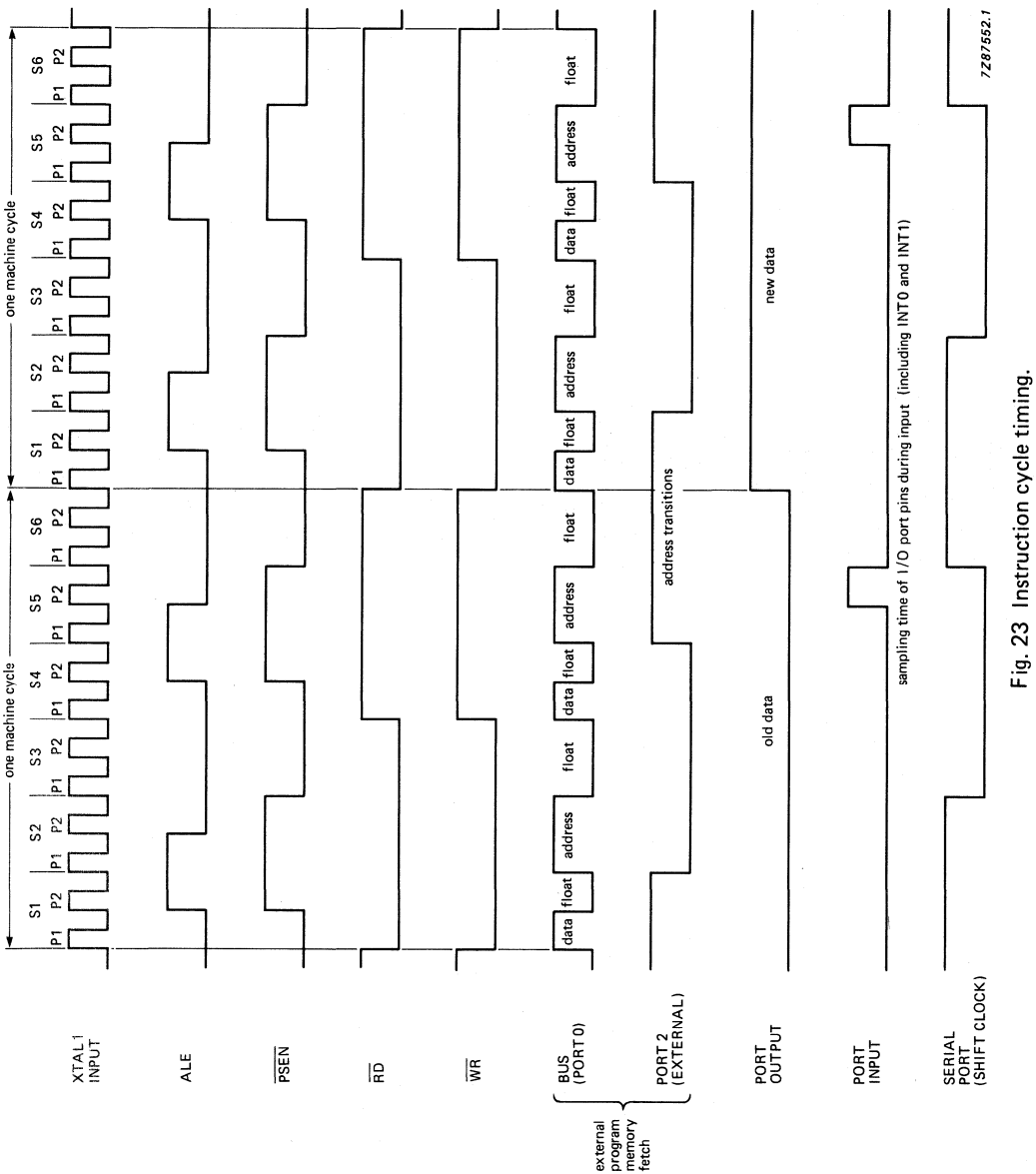


Fig. 23 Instruction cycle timing.

SINGLE-CHIP 8-BIT MICROCONTROLLER

GENERAL DESCRIPTION

The PCB83C652 single-chip 8-bit microcontroller is manufactured in an advanced CMOS process and is a derivative of the PCB80C51 microcontroller family. PCB83C652 has the same instruction set as the PCB80C51. Two versions of the derivative exist although the generic term "PCB83C652" is used to refer to both family members:

- PCB80C652: ROM-less version of the PCB83C652
- PCB83C652: 8 K bytes mask-programmable ROM, 256 bytes RAM

This device provides architectural enhancements that make it applicable in a variety of applications, in general control systems.

The PCB83C652 contains a non-volatile 8 K x 8 read-only program memory (not ROM-less version), a volatile 256 x 8 read/write data memory; four 8-bit I/O ports; two 16-bit timer/event counters (identical to the timers of the 80C51), a multi-source, two-priority-level, nested interrupt structure; an I^2C interface, UART and on-chip oscillator and timing circuits. For systems that require extra capability, the PCB83C652 can be expanded using standard TTL compatible memories and logic.

The device also functions as an arithmetic processor having facilities for both binary and BCD arithmetic plus bit-handling capabilities. The instruction set consists of over 100 instructions; 44% one-byte, 41% two-byte and 15% three-byte. With a 12 MHz crystal, 58% of the instructions are executed in 1 μs and 40% in 2 μs. Multiply and divide instructions require 4 μs.

Features

- 80C51 central processing unit
- 8 K x 8 ROM, expandable externally to 64 K bytes
- 256 x 8 RAM, expandable externally to 64 K bytes
- Two standard 16-bit timer/counters
- Four 8-bit I/O ports
- I^2C bus serial I/O port with byte orientated master and slave functions
- Full-duplex UART facilities

PACKAGE OUTLINE

PCB83C652: 44-lead PLCC; plastic, leaded-chip-carrier (SOT-187A)

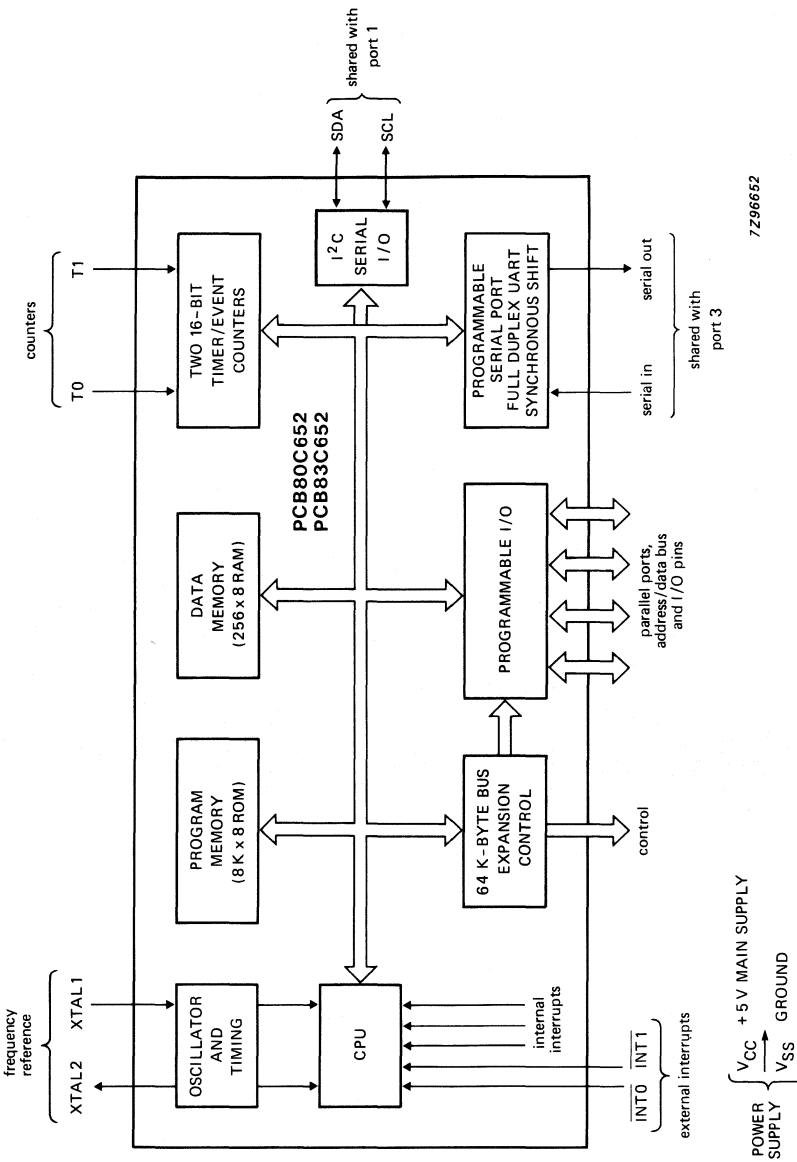


Fig. 1 Block diagram.

DEVELOPMENT DATA

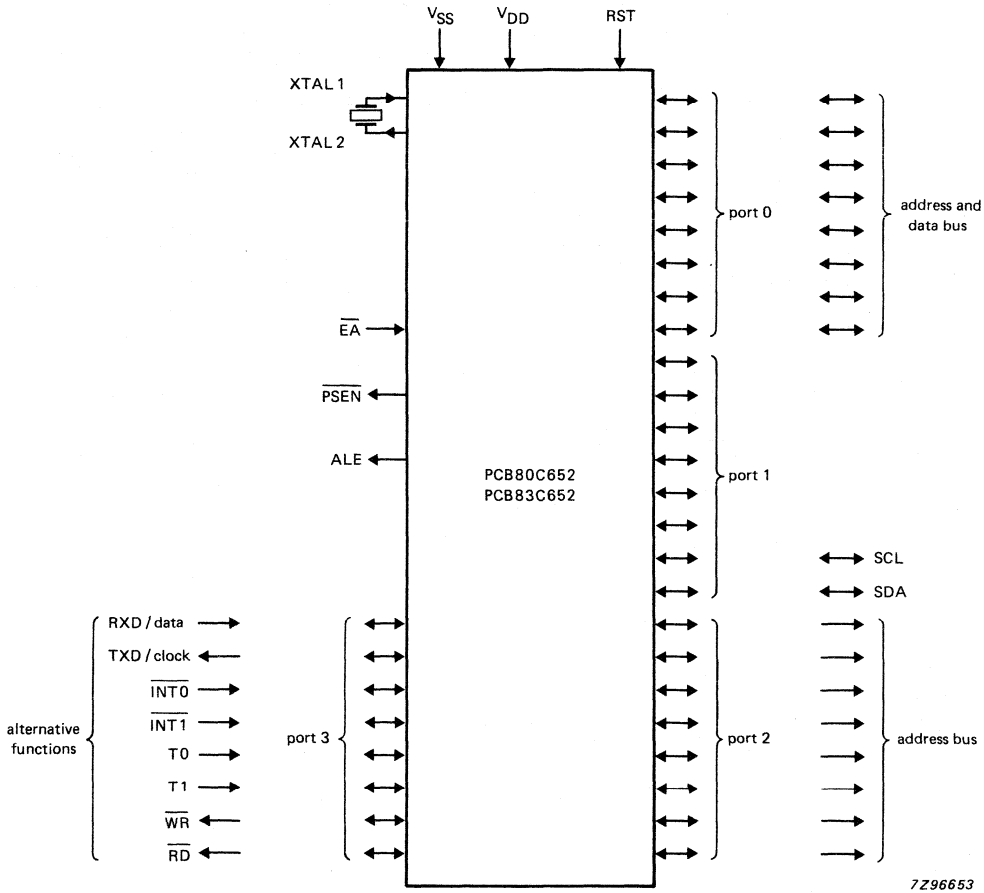


Fig. 2 Functional diagram.

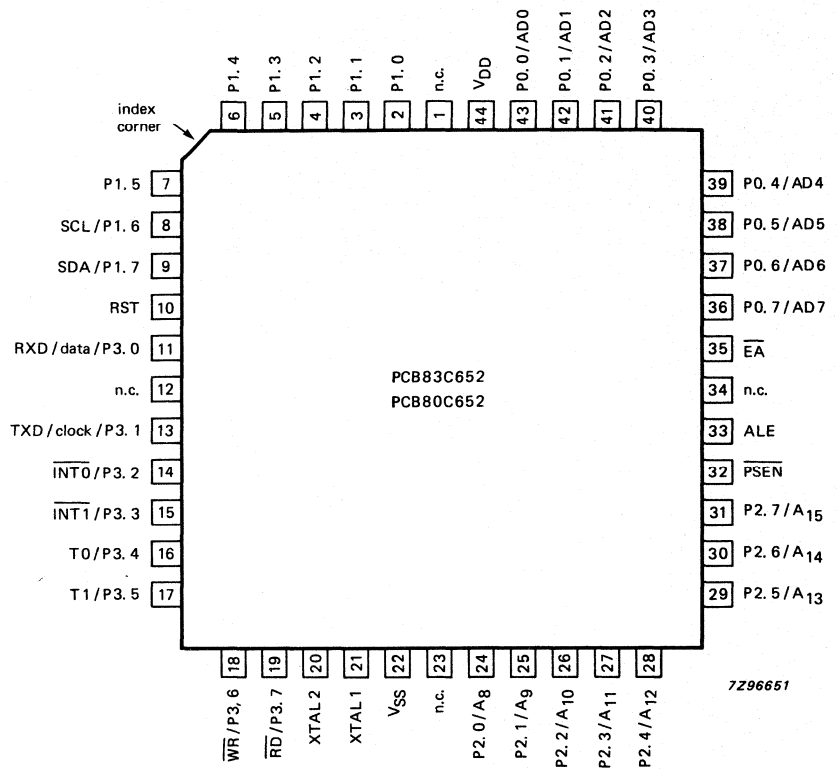


Fig. 3 Pinning diagram for PCB83C652.

PINNING

2-9	P1.0-P1.7	**Port 1:** 8-bit quasi-bidirectional I/O port. Port 1 can sink/source one TTL (= 4 LS TTL) input. It can drive CMOS inputs without external pull-ups, except P1.6 and P1.7 which have open drain outputs.

Port pin	Alternative function
P1.6	SCL: I^2C-bus serial port clock line
P1.7	SDA: I^2C-bus serial port data line

10 **RST**: a high level on this pin for two machine cycles while the oscillator is running resets the device. An internal pull-down permits Power-On reset using only a capacitor connected to V_{DD}.

11, 13-19	P3.0-P3.7	**Port 3:** 8-bit quasi-bidirectional I/O port with internal pull-ups. It also serves the following alternative functions:

Port pin	Alternative function
P3.0	RXD/data: serial port receiver data input (asynchronous) or data input/output (synchronous)
P3.1	TXD/clock: serial port transmitter data output (asynchronous) or clock output (synchronous)
P3.2	$\overline{\text{INT0}}$: external interrupt 0 or gate control input for timer/event counter 0
P3.3	$\overline{\text{INT1}}$: external interrupt 1 or gate control input for timer/event counter 1
P3.4	T0: external input for timer/event counter 0
P3.5	T1: external input for timer/event counter 1
P3.6	$\overline{\text{WR}}$: external data memory write strobe
P3.7	$\overline{\text{RD}}$: external data memory read strobe

Operation of an alternative function is determined by the relevant output latch programmed to logic 1. Port 3 can sink/source one TTL (= 4 LSTTL) input. It can drive CMOS inputs without external pull-ups.

20	XTAL 2	**Crystal input 2:** output of the inverting amplifier that forms the oscillator. Left open-circuit when an external oscillator clock is used (see Figs 10 and 11).
21	XTAL 1	**Crystal input 1:** input to the inverting amplifier that forms the oscillator, and input to the internal clock generator. Receives the external oscillator clock signal when an external oscillator is used (see Figs 10 and 11).
22	V_{SS}	**Ground:** circuit ground potential.
24-31	P2.0-P2.7	**Port 2:** 8-bit quasi-bidirectional I/O port with internal pull-ups. During access to external memories (RAM/ROM) that use 16-bit addresses (MOVX @DPTR) Port 2 emits the high order address byte. When external RAM is accessed with an 8-bit address (MOVX @Ri) Port 2 emits the contents of the P2 special function register. Port 2 can sink/source one TTL (= 4 LSTTL) input. It can drive CMOS inputs without external pull-ups.
32	$\overline{\text{PSEN}}$	**Program Store Enable output:** read strobe to the external Program Memory via port 0 and 2. It is activated twice each machine cycle during fetches from external Program Memory. When executing out of external Program Memory two activations of $\overline{\text{PSEN}}$ are skipped during each access to external Data Memory. $\overline{\text{PSEN}}$ is not activated (remains HIGH) during no fetches from external Program Memory. $\overline{\text{PSEN}}$ can sink/source 8 LS TTL inputs. It can drive CMOS inputs without an external pull-up.
33	ALE	**Address Latch Enable output:** latches the low byte of the address during accesses to external memory in normal operation. It is activated every six oscillator periods except during an external data memory access. ALE can sink/source 8 LS TTL inputs. It can drive CMOS inputs without an external pull-up.

DEVELOPMENT DATA

PINNING (continued)

35	$\overline{EA}$	**External Access input:** When $\overline{EA}$ is held at a TTL high level the CPU executes out of the internal Program Memory (ROM) provided the program counter is less than 8192. When $\overline{EA}$ is held at a TTL low level, the CPU executes out of external Program Memory via port 0 and port 2. $\overline{EA}$ is not allowed to float.
36-43	P0.7-P0.0	**Port 0:** 8-bit open drain bidirectional I/O port. It is also the multiplexed low-order address and data bus during accesses to external memory (during these accesses it activates internal pull-ups). Port 0 can sink/source eight LSTTL inputs.
44	V_{DD}	**Power supply:** + 5 V power supply pin during normal operation, Idle mode and Power Down mode.

At power-on, the voltage on any pin at any time must not be higher or lower than V_{DD} + 0,5 V or V_{SS} – 0,5 V respectively.

FUNCTIONAL DESCRIPTION

General

The PCB83C652 is a stand-alone high-performance microcontroller designed for use in real-time applications such as instrumentation and industrial control.

The device provides, in addition to the 80C51 standard functions, a serial I^2C bus interface. As well as the parallel bus, functions may also be expanded using the I^2C bus utilizing the complete line of the I^2C clips family.

The PCB83C652 is a control-oriented CPU with on-chip program and data memory. It can be extended with external program and data memory up to 64 K bytes. For systems requiring extra capability, the PCB83C652 can be expanded using standard memories and peripherals.

The PCB83C652 has two software selectable modes of reduced activity for further power reduction — Idle and Power Down. The Idle mode freezes the CPU while allowing the RAM, timers, serial ports and interrupt system to continue functioning. The Power Down mode saves the RAM contents but freezes the oscillator causing all other chip functions to be inoperative.

Memory organization

The central processing unit (CPU) manipulates operands in three memory spaces; these are the 64 K-byte external data memory, 256-byte internal data memory and the 64 K-byte internal and external program memory. The internal data memory address space is sub-divided into the 256-byte internal data RAM and 128-byte Special Function Register (SFR) address spaces, as shown in Fig. 4. Figure 5 shows the Special Function Register memory map. Internal RAM locations 0-127 are directly and indirectly addressable. Internal RAM locations 128-255 are only indirectly addressable as internal data RAM. The special function register locations 128-255 are only directly addressable.

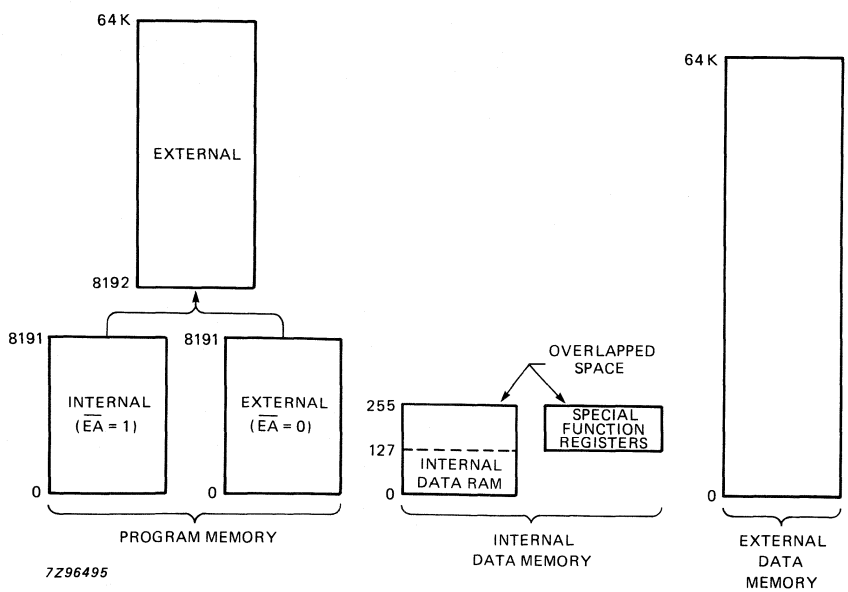
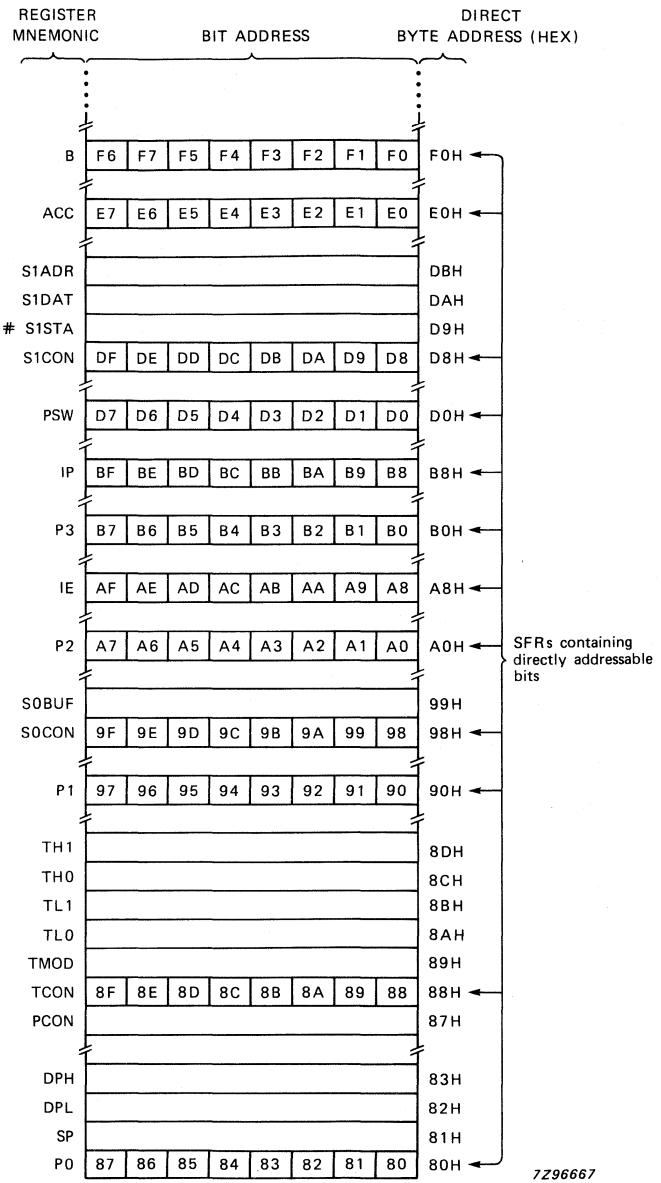


Fig. 4 Memory map.



S1STA is a read-only register.

Fig. 5 Special function registers memory map.

FUNCTIONAL DESCRIPTION (continued)

The internal data RAM contains four register banks (each with eight registers), 128 addressable bits, and the stack. The stack depth is limited by the available internal data RAM and its location is determined by the 8-bit stack pointer. All registers except the program counter and the four 8-register banks reside in the special function register address space. These memory mapped registers include arithmetic registers, pointers, I/O ports, interrupt system registers, timers and serial port registers. There are 128 addressable bit locations in the SFR address space.

The PCB83C652 contains 256 bytes of internal data RAM and 24 special function registers. It provides a non-paged program memory address space to accommodate relocatable code. Conditional branches are performed relative to the program counter. The register-indirect jump permits branching relative to a 16-bit base register with an offset provided by an 8-bit index register. 16-bit jumps and calls permit branching to any location in the contiguous 64 K program memory address space.

Addressing

The PCB83C652 has five methods for addressing source operands:

- Register.
- Direct.
- Register-Indirect.
- Immediate.
- Base-Register plus Index-Register-Indirect.

The first three methods can be used for addressing destination operands. Most instructions have a "destination/source" field that specifies the data type, addressing methods and operands involved. For operations other than MOVs, the destination operand is also a source operand.

Access to memory addressing is as follows:

- Registers in one of the four 8-register banks through Register, Direct, or Register-Indirect.
- 256 bytes of internal data RAM through Direct or Register-Indirect. Bytes 0-127 may be addressed directly/indirectly. Bytes 128-255 share their address locations with the SFR registers and so may only be addressed indirectly as data RAM.
- Special function registers through Direct at address locations 128-255.
- External data memory through Register-Indirect.
- Program memory look-up tables through Base-Register plus Index-Register-Indirect.

The PCB83C652 is classified as an 8-bit device since the internal ROM, RAM, Special Function Registers (SFR), Arithmetic Logic Unit (ALU), and external data bus are each 8-bits wide. It performs operations on bit, nibble, byte and double-byte data types.

Facilities are available for byte transfer, logic, and integer arithmetic operations. Data transfer, logic, and conditional branch operations can be performed directly on Boolean variables to provide excellent bit handling.

Instruction set

The PCB83C652 uses a powerful instruction set to allow expansion of on-chip CPU peripherals and to optimize byte efficiency and execution speed. Assigned opcodes add new high-power operations and permit new addressing modes. The instruction set consists of 49 single-byte, 45 two-byte and 17 three-byte instructions. When using a 12 MHz oscillator, 64 instructions execute in 1 μs and 45 instructions execute in 2 μs. Multiply and divide instructions execute in 4 μs.

FUNCTIONAL DESCRIPTION (continued)

I/O facilities

The PCB83C652 has four 8-bit ports. Ports 0-3 are the same as in the 80C51, with the exception of the additional functions of port 1. Port lines P1.7 and P1.6 may be selected as the SDA and SCL lines of serial port SIO1 (I^2C). Because the I^2C bus may be active while the device is disconnected from V_{DD}, these pins are provided with open drain drivers.

N.B. Therefore these pins do not have pull-up devices when used as ports.

Ports 0, 1, 2 and 3 perform the following alternative functions:

- Port 0; provides the multiplexed low-order address and data bus used for expanding the PCB83C652 with standard memories and peripherals.
- Port 1; Port 1 is partly used for the I^2C bus functions;
  - SCL and SDA for the I^2C interface, P1.6 and P1.7 respectively.

Bits whose alternate function is not used may be used as normal bidirectional I/O pins.

- Port 2; provides the high-order address bus when expanding the PCB83C652 with external program memory and/or external data memory.
- Port 3; pins can be configured individually to provide:
  - external interrupt request inputs
  - counter inputs
  - serial port receiver input and transmitter output
  - control signals to READ and WRITE external data memory

The generation or use of a Port 3 pin as an alternative function is carried out automatically by the PCB83C652 provided the pin is loaded with a HIGH content.

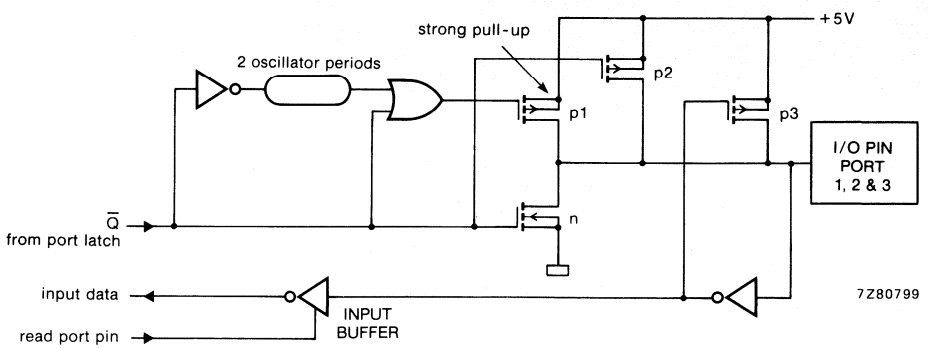


Fig. 6 I/O buffers in the PCB83C652 (Ports 1, 2 and 3).

Timer/event counters

The PCB83C652 contains two 16-bit timer/event counters; Timer 0 and Timer 1. Timer 0 and Timer 1 may be programmed to carry out the following functions:

- Measure time intervals and pulse durations
- Count events
- Generate interrupt requests.

Timer 0 and Timer 1 can be programmed independently to operate in three modes:

- Mode 0; 8-bit timer or 8-bit counter each with divide by 32 prescaler
- Mode 1; 16-bit time-interval or event counter
- Mode 2; 8-bit time-interval or event counter with automatic reload upon overflow.

Timer 0 can be programmed to operate in an additional mode as follows:

- Mode 3; one 8-bit time-interval or event counter and one 8-bit time-interval counter.

When Timer 0 is in Mode 3, Timer 1 can be programmed to operate in Modes 0, 1 or 2 but cannot set an interrupt request flag or generate an interrupt. However the overflow from Timer 1 can be used to pulse the serial Port transmission-rate generator.

The frequency handling range of these counters with a 12 MHz crystal is as follows:

- In the timer function, the timer is incremented at a frequency of 1 MHz, that is, a division by 12 of the oscillator frequency
- 0 Hz to an upper limit of 0,5 MHz when programmed for external inputs.

Both internal and external inputs can be gated to the counter by a second external source for directly measuring pulse durations.

The counters are started and stopped under software control. Each one sets its interrupt request flag when it overflows from all logic 1s to all logic 0s (or automatic reload value), with the exception of mode 3 as previously described.

Serial I/O (see Fig. 7)

The PCB83C652 is equipped with two independent serial ports. SIO0 is the full duplex UART port and is identical to the serial port of the PCB80C51.

Serial port SIO1 supports the I^2C bus, the function of which is controlled by the S1CON register. S1STA is the status register whose contents may also be used as a vector to various service routines. S1DAT is the data shift register and S1ADR the slave address register.

FUNCTIONAL DESCRIPTION (continued)

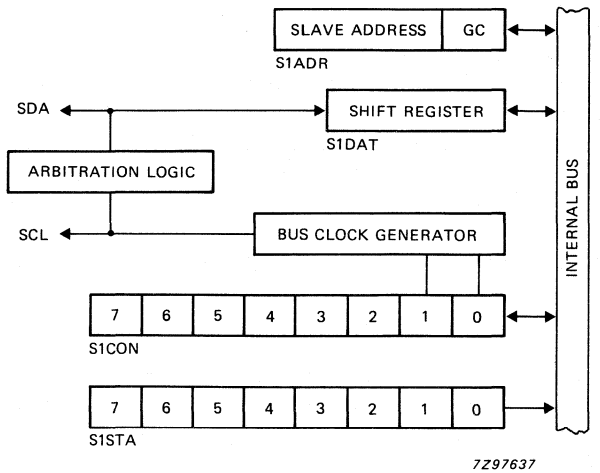


Fig. 7 Block diagram of I^2C serial I/O.

The I^2C serial I/O has complete autonomy in byte handling and operates in 4 modes:

1. Master transmitter
2. Master receiver
3. Slave transmitter
4. Slave receiver

Slave address recognition is performed by hardware.

The I^2C bus consists of two lines; a data line (SDA) and a clock line (SCL). These lines also function as the I/O port lines on P1.7 and P1.6. The system is unique because data transport, clock generation, address recognition and bus control arbitration are all controlled by hardware.

Serial control register S1CON

	X	ENSI	STA	STO	SI	AA	CR1	CR0
S1CON	7	6	5	4	3	2	1	0

Bits CR1 and CR0 determine the clock frequency that is generated in the master mode of operation. Table 1 displays the clock rate when using a 12 MHz crystal.

Table 1 Clock rate when using a 12 MHz crystal

CR1 / CR0	bit frequency	f_{osc} divided by
0 0	12,5 kHz	960
0 1	100 kHz	120
1 0	200 kHz	60 ($f_{osc} < 6$ MHz meeting I^2C)
1 1	62,5–0,5 kHz	96 x (256-reload value Timer 1) (reload value range: 0-254 in mode 2)

AA

Assert acknowledge bit. When this bit is set, an acknowledge is returned after any one of the following conditions:

- Own slave address is received
- General call address is received (S1ADR.0 = logic 1)
- A data byte is received, while the device is programmed to be a master receiver
- A data byte is received, while the device is a selected slave receiver.

When this bit is reset, no acknowledge is returned. Consequently, no interrupt is requested when the own address or general call address is received.

SI

SIO1 interrupt flag. This flag is set, and an interrupt request is generated, after any of the following events occur:

- A START condition is generated in MST mode
- The own slave address has been received during AA = logic 1
- The general call address has been received while S1ADR.0 and AA = logic 1
- A data byte has been received or transmitted in MST mode (even if arbitration is lost)
- A data byte has been received or transmitted as selected slave
- A STOP or START condition is received as selected slave Rx or Tx.

STO

STOP flag. When in master mode,and this bit is set a STOP condition is generated. A STOP condition detected on the I^2C bus clears this bit. This bit may also be set in slave mode in order to recover from an error condition. Then no STOP condition is generated to the I^2C bus, but the hardware releases the SDA and SCL lines and switches to the not selected slave receiver mode. The STOP flag is cleared by the hardware.

STA

START flag. When this bit is set, the hardware checks the I^2C bus and generates a START condition if the bus is free. If the device is already programmed as either master/transmitter or master/receiver, it will generate a repeated START condition.

ENSI

0 = Serial I/O Disabled and reset. P1.6 and P1.7 I/O port function with open drain

1 = Serial I/O Enabled. Output ports P1.6 and P1.7 must be set to logic 1.

FUNCTIONAL DESCRIPTION (continued)

Serial status register S1STA

	SC4	SC3	SC2	SC1	SC0	0	0	0
S1STA	7	6	5	4	3	2	1	0

S1STA.3 - S1STA.7 hold a status code. S1STA.0 - S1STA.2 are held LOW. The contents of the status register may be used as a vector to a service routine. This optimizes the response time of the software and consequently that of the I^2C bus.

The following is a list of the status codes in decimal representation. The decimal value corresponds to the value of the upper five bits of the status register.

SLA : 7-bit slave address
R : Read bit
W : Write bit
ACK : Acknowledgement (acknowledge bit = logic 0)
$\overline{\text{ACK}}$: Not acknowledgement (acknowledge bit = logic 1)
DATA : 8-bit data byte to or from I^2C bus
MST : Master
SLV : Slave
TRX : Transmitter
REC : Receiver

MST/TRX mode

S1STA value

1 - A START condition has been transmitted
2 - A repeated START condition has been transmitted
3 - SLA and W have been transmitted, ACK has been received
4 - SLA and W have been transmitted, $\overline{\text{ACK}}$ received
5 - DATA of S1DAT has been transmitted, ACK received
6 - DATA of S1DAT has been transmitted, $\overline{\text{ACK}}$ received
7 - Arbitration lost in SLA and R/W or DATA

MST/REC mode

S1STA value

7 - Arbitration lost while returning $\overline{\text{ACK}}$
8 - SLA and R have been transmitted, ACK received
9 - SLA and R have been transmitted, $\overline{\text{ACK}}$ received
10 – DATA has been received, ACK returned
11 - DATA has been received, $\overline{\text{ACK}}$ returned

SLV/REC mode

S1STA value

- 12 - Own SLA and W have been received, ACK returned
- 13 - Arbitration lost in SLA and R/W as MST. Own SLA and W have been received, ACK returned
- 14 - General CALL has been received, ACK returned
- 15 - Arbitration lost in SLA and R/W as MST. General call has been received
- 16 - Previously addressed with own SLA. DATA byte received, ACK returned
- 17 - Previously addressed with own SLA. DATA byte received, $\overline{\text{ACK}}$ returned
- 18 - Previously addressed with general call. DATA byte has been received, ACK has been returned
- 19 - Previously addressed with general call. DATA byte has been received, $\overline{\text{ACK}}$ has been returned
- 20 - A STOP condition or repeated START condition has been received while still addressed as SLV/REC or SLV/TRX.

SLV/TRX mode

S1STA value

- 21 - Own SLA and R have been received, ACK returned
- 22 - Arbitration lost in SLA and R/W as MST. Own SLA and R have been received, ACK returned
- 23 - DATA byte has been transmitted, ACK received
- 24 - DATA byte has been transmitted, $\overline{\text{ACK}}$ received
- 25 - Last DATA byte has been transmitted (AA = logic 0), ACK received

Miscellaneous

S1STA value

- 00 - Bus error during MST mode or selected SLV mode, due to an erroneous START or STOP condition.

The data shift register S1DAT

	7	6	5	4	3	2	1	0
S1DAT								

This register contains the serial data to be transmitted or data which has just been received. Bit 7 is transmitted or received first; i.e. data is shifted from right to left.

Address register S1ADR

	slave address							GC
S1ADR	7	6	5	4	3	2	1	0

S1ADR.0, GC : 0 = general call address is not recognized
1 = general call address is recognized

S1ADR,7-1 : own slave address

This 8-bit register may be loaded with the 7-bit slave address to which the controller will respond when programmed as a slave receiver/transmitter. The LSB bit (GC) is used to determine whether the general call address is recognized.

FUNCTIONAL DESCRIPTION (continued)

Idle and Power-down operation (see Fig. 8)

Idle mode operation permits the interrupt, serial ports and timer blocks to continue to function while the clock to the CPU is halted.

The following functions remain active during IDLE mode. These functions may generate an interrupt or reset and thus end the IDLE mode:

Timer 0, Timer 1
SIO0, SIO1
External interrupt

The Power-down operation freezes the oscillator. The Power-down mode can only be activated by setting the PD bit in the PCON register.

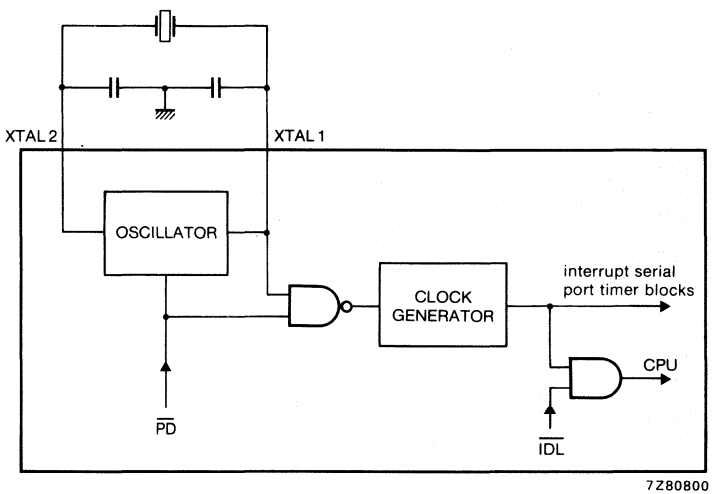


Fig. 8 Internal Idle and Power-down clock configuration.

Power control register (PCON)

These special modes are activated by software via the Special Function Register PCON. Its hardware address is 87H. PCON is not bit addressable.

PCON

SMOD	X	X	X	GF1	GF0	PD	IDL
7	6	5	4	3	2	1	0

Bit		Definition
SMOD	PCON.7	Double Baud rate bit. When set to logic 1 the baud rate is doubled if the serial port SIO0 is being used in modes 1, 2 or 3.
	PCON.6	(reserved)
	PCON.5	(reserved)
	PCON.4	(reserved)
GF1	PCON.3	general-purpose flag bit
GF0	PCON.2	general-purpose flag bit
PD	PCON.1	Power-down bit; setting this bit activates power-down mode
IDL	PCON.0	Idle mode bit; setting this bit activates the idle mode operation.

If logic 1s are written to PD and IDL at the same time, PD takes precedence. The reset value of PCON is (0XX00000).

Idle mode

The instruction that sets PCON.0 is the last instruction executed in the normal operating mode before Idle mode is activated. Once in the Idle mode, the CPU status is preserved in its entirety: the Stack Pointer, Program Counter, Program Status Word, Accumulator, RAM and all other registers maintain their data during Idle mode. The status of the external pins during Idle mode is shown in Table 2.

There are two ways to terminate the Idle mode:

Activation of any enabled interrupt will cause PCON.0 to be cleared by hardware terminating Idle mode. The interrupt is serviced, and following return from interrupt instruction RETI, the next instruction to be executed will be the one which follows the instruction that wrote a logic 1 to PCON.0.

The flag bits GF0 and GF1 may be used to determine whether the interrupt was received during normal execution or during the Idle mode. For example, the instruction that writes to PCON.0 can also set or clear one or both flag bits. When Idle mode is terminated by an interrupt, the service routine can examine the status of the flags bits.

The second way of terminating the Idle mode is with an external hardware reset. Since the oscillator is still running, the hardware reset is required to be active for two machine cycles (24 oscillator periods) to complete the reset operation.

Power-down mode

The instruction that sets PCON.1 is the last executed prior to going into the Power-down mode. Once in Power-down mode, the oscillator is stopped. Only the contents of the on-chip RAM are preserved. The Special Function Registers are not saved. A hardware reset is the only way of exiting the Power--down mode.

In the Power-down mode, V_{DD} may be reduced to minimize circuit power consumption. The voltage must not be reduced until the Power-down mode is entered, but must be restored before the hardware reset is applied which will free the oscillator. Reset should not be released until the oscillator has restarted and stabilized.

The status of the external pins during Power-down mode is shown in Table 2. If the Power-down mode is activated while in external program memory, the port data that is held in the Special Function Register P2 is restored to Port 2. If the data is a logic 1, the port pin is held HIGH during the Power--down mode by the strong pull-up transistor p1 (see Fig. 6).

FUNCTIONAL DESCRIPTION (continued)

Table 2 Status of the external pins during Idle and Power-down modes.

mode	memory	ALE	$\overline{PSEN}$	Port 0	Port 1	Port 2	Port 3
Idle	internal	1	1	port data	port data	port data	port data
Idle	external	1	1	floating	port data	address	port data
Power-down	internal	0	0	port data	port data	port data	port data
Power-down	external	0	0	floating	port data	port data	port data

Note: Port 1.7 and 1.6 if selected, function as SDA and SCL respectively in the IDLE mode.

Interrupt system (see Fig. 9)

External events and the real-time-driven on-chip peripherals require service by the CPU asynchronous to the execution of any particular section of code. To tie the asynchronous activities of these functions to normal program execution a multiple-source, two-priority-level, nested interrupt system is provided. Interrupt response latency is from 3 μs to 8 μs when using a 12 MHz crystal. The PCB83C652 acknowledges interrupt requests from six sources as follows:

- $\overline{INT0}$ and $\overline{INT1}$; externally via pins 14 and 15 respectively
- Timer 0 and Timer 1; from the two internal counters
- I^2C serial I/O interrupt
- UART serial I/O port interrupt

Each interrupt vectors to a separate location in program memory for its service program. Each source can be individually enabled or disabled by a corresponding bit in the IE register, moreover each interrupt may be programmed to a high or low priority level using a corresponding bit in the IP register. Also all enabled sources can be globally disabled or enabled. Both external interrupts can be programmed to be level-activated or transition-activated, and an active LOW level allows "wire-ORing" of several interrupt sources to the input pin.

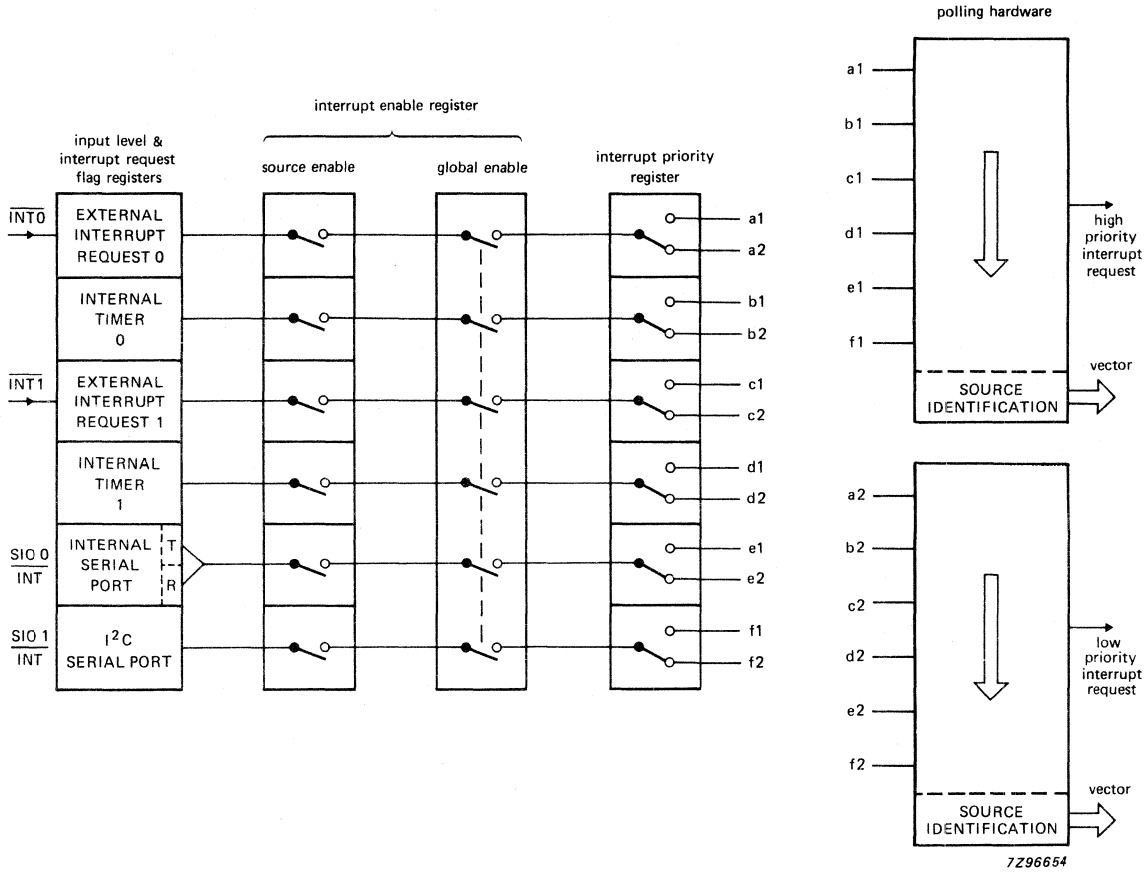


Fig. 9 Interrupt system.

FUNCTIONAL DESCRIPTION (continued)

Interrupt enable register

IE

EA	–	ES1	ES0	ET1	EX1	ET0	EX0
7	6	5	4	3	2	1	0

Bit		Function
IE.7	EA	General enable/disable control 0 = No interrupt is enabled 1 = Any individually enabled interrupt will be accepted
IE.6	–	Unused
IE.5	ES1	Enable SIO1 I^2C interrupt
IE.4	ES0	Enable SIO0 (UART) interrupt
IE.3	ET1	Enable Timer 1 interrupt
IE.2	EX1	Enable External 1 interrupt
IE.1	ET0	Enable Timer 0 interrupt
IE.0	EX0	Enable External 0 interrupt

Interrupt priority register

IP

X	X	PS1	PS0	PT1	PX1	PT0	PX0
7	6	5	4	3	2	1	0

Bit		Function
IP.7	–	Unused
IP.6	–	Unused
IP.5	PS1	SIO1 (I^2C) interrupt priority level
IP.4	PS0	SIO0 (UART) interrupt priority level
IP.3	PT1	Timer 1 interrupt priority level
IP.2	PX1	External interrupt 1 priority level
IP.1	PT0	Timer 0 interrupt priority level
IP.0	PX0	External interrupt 0 priority level

Table 3 shows the interrupt vectors. The vector indicates the ROM location where the appropriate interrupt reduce routine starts.

Table 3 Interrupt vectors

Source		Vector
External 0	X0	0003H
Timer 0 overflow	T0	000BH
External 1	X1	0013H
Timer 1 overflow	T1	001BH
Serial I/O 0 (UART)	S0	0023H
Serial I/O 1 (I^2C)	S1	002BH

Interrupt priority

Each interrupt source can be provided by software with two priority levels; logic 1 and logic 0. If both levels are requested simultaneously, the processor will branch to the level logic 1 vector. If there are simultaneous requests from sources of the same priority, then interrupts will be serviced in the following order:

X0, S1, T0, X1, T1, S0

A logic 0 interrupt routine can be interrupted by a logic 1 interrupt.

Oscillator circuitry

The oscillator circuitry of the PCB83C652 is a single-stage inverting amplifier in a Pierce oscillator configuration. The circuitry between XTAL 1 and XTAL 2 is basically an inverter biased to the transfer point. Either a crystal or ceramic resonator can be used as the feedback element to complete the oscillator circuitry. Both are operated in parallel resonance. XTAL 1 (pin 21) is the high gain amplifier input, and XTAL 2 (pin 20) is the output (see Fig. 10). To drive the PCB83C652 externally, XTAL 1 is driven from an external source and XTAL 2 left open-circuit (see Fig. 11).

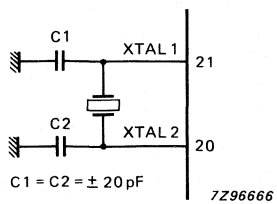


Fig. 10 PCB83C652 oscillator circuit.

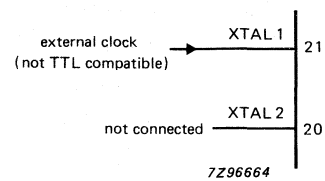


Fig. 11 Driving the PCB83C652 from an external source.

FUNCTIONAL DESCRIPTION (continued)

Reset circuitry (see Fig. 12)

The reset circuitry for the PCB83C652 is connected to the reset pin RST. A Schmitt trigger is used at the input for noise rejection. The output of the Schmitt trigger is sampled by the reset circuitry every machine cycle.

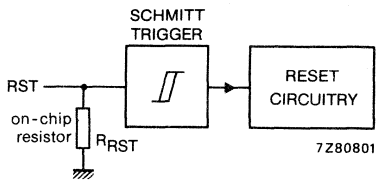


Fig. 12 Reset configuration at RST.

A reset is accomplished by holding the RST pin HIGH for at least two machine cycles (24 oscillator periods), while the oscillator is running. The CPU responds by executing an internal reset. During reset ALE and PSEN output a HIGH level. In order to perform a correct reset, this level must not be affected by external elements.

The internal reset is executed during the second cycle in which RST is HIGH and is repeated every cycle until RST goes LOW. It leaves the internal registers as follows:

Register	Content
ACC	0000 0000
B	0000 0000
DPL	0000 0000
DPH	0000 0000
IE	0X00 0000
IP	XX00 0000
PCH	0000 0000
PCL	0000 0000
PCON	0XXX 0000
PSW	0000 0000
P0 - P3	1111 1111
S0BUF	XXXX XXXX
S0CON	0000 0000
S1ADR	0000 0000
S1CON	X000 0000
S1DAT	0000 0000
S1STA	1111 1000
SP	0000 0111
TCON	0000 0000
TH0, TH1	0000 0000
TL0, TL1	0000 0000
TMOD	0000 0000

The internal RAM is not affected by reset. When V_{DD} is turned on, the RAM content is indeterminate.

Power-on reset (see Fig. 13)

When V_{DD} is turned on an automatic reset can be obtained by connecting the RST pin to V_{DD} via a 10 μF capacitor. When the power is switched on, the current drawn by RST is the difference between V_{DD} and the capacitor voltage, and decreases from V_{DD} as the capacitor charges through the internal resistor (R_{RST}) to ground. The larger the capacitor, the more slowly V_{RST} decreases. V_{RST} must remain above the lower threshold of the Schmitt trigger long enough to effect a complete reset. The time required is the oscillator start-up time, plus 2 machine cycles.

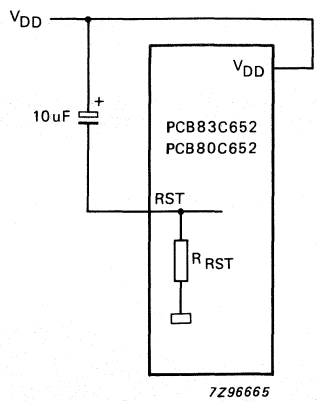


Fig. 13 Power-on reset.

INSTRUCTION SET

Table 4 Instruction set description

mnemonic		description	bytes/ cycles		opcode (hex.)
Arithmetic operations					
ADD	A,Rr	Add register to A	1	1	2*
ADD	A,direct	Add direct byte to A	2	1	25
ADD	A,@Ri	Add indirect RAM to A	1	1	26, 27
ADD	A,#data	Add immediate data to A	2	1	24
ADDC	A,Rr	Add register to A with carry flag	1	1	3*
ADDC	A,direct	Add direct byte to A with carry flag	2	1	35
ADDC	A,@Ri	Add indirect RAM to A with carry flag	1	1	36, 37
ADDC	A,#data	Add immediate data to A with carry flag	2	1	34
SUBB	A,Rr	Subtract register from A with borrow	1	1	9*
SUBB	A,direct	Subtract direct byte from A with borrow	2	1	95
SUBB	A,@Ri	Subtract indirect RAM from A with borrow	1	1	96, 97
SUBB	A,#data	Subtract immediate data from A with borrow	2	1	94
INC	A	Increment A	1	1	04
INC	Rr	Increment register	1	1	0*
INC	direct	Increment direct byte	2	1	05
INC	@Ri	Increment indirect RAM	1	1	06, 07
DEC	A	Decrement A	1	1	14
DEC	Rr	Decrement register	1	1	1*
DEC	direct	Decrement direct byte	2	1	15
DEC	@Ri	Decrement indirect RAM	1	1	16, 17
INC	DPTR	Increment data pointer	1	2	A3
MUL	AB	Multiply A & B	1	4	A4
DIV	AB	Divide A by B	1	4	84
DA	A	Decimal adjust A	1	1	D4

mnemonic		description	bytes/ cycles		opcode (hex.)
Logic operations					
ANL	A,Rr	AND register to A	1	1	5*
ANL	A,direct	AND direct byte to A	2	1	55
ANL	A,@Ri	AND indirect RAM to A	1	1	56, 57
ANL	A,#data	AND immediate data to A	2	1	54
ANL	direct,A	AND A to direct byte	2	1	52
ANL	direct,#data	AND immediate data to direct byte	3	2	53
ORL	A,Rr	OR register to A	1	1	4*
ORL	A,direct	OR direct byte to A	2	1	45
ORL	A,@Ri	OR indirect RAM to A	1	1	46. 47
ORL	A,#data	OR immediate data to A	2	1	44
ORL	direct,A	OR A to direct byte	2	1	42
ORL	direct,#data	OR immediate data to direct byte	3	2	43
XRL	A,Rr	Exclusive-OR register to A	1	1	6*
XRL	A,direct	Exclusive-OR direct byte to A	2	1	65
XRL	A,@Ri	Exclusive-OR indirect RAM to A	1	1	66, 67
XRL	A,#data	Exclusive-OR immediate data to A	2	1	64
XRL	direct,A	Exclusive-OR A to direct byte	2	1	62
XRL	direct,#data	Exclusive-OR immediate data to direct byte	3	2	63
CLR	A	Clear A	1	1	E4
CPL	A	Complement A	1	1	F4
RL	A	Rotate A left	1	1	23
RLC	A	Rotate A left through the carry flag	1	1	33
RR	A	Rotate A right	1	1	03
RRC	A	Rotate A right through the carry flag	1	1	13
SWAP	A	Swap nibbles within A	1	1	C4

INSTRUCTION SET (continued)

mnemonic		description	bytes/ cycles		opcode (hex.)
Data transfer					
MOV	A,Rr	Move register to A	1	1	E*
MOV	A,direct (**)	Move direct byte to A	2	1	E5
MOV	A,@Ri	Move indirect RAM to A	1	1	E6, E7
MOV	A,#data	Move immediate data to A	2	1	74
MOV	Rr,A	Move A to register	1	1	F*
MOV	Rr,direct	Move direct byte to register	2	2	A*
MOV	Rr,#data	Move immediate data to register	2	1	7*
MOV	direct,A	Move A to direct byte	2	1	F5
MOV	direct,Rr	Move register to direct byte	2	2	8*
MOV	direct,direct	Move direct byte to direct	3	2	85
MOV	direct,@Ri	Move indirect RAM to direct byte	2	2	86, 87
MOV	direct,#data	Move immediate data to direct byte	3	2	75
MOV	@Ri,A	Move A to indirect RAM	1	1	F6, F7
MOV	@Ri,direct	Move direct byte to indirect RAM	2	2	A6, A7
MOV	@Ri,#data	Move immediate data to indirect RAM	2	1	76, 77
MOV	DPTR,#data 16	Load data pointer with a 16-bit constant	3	2	90
MOVC	A,@A+DPTR	Move code byte relative to DPTR to A	1	2	93
MOVC	A,@A+PC	Move code byte relative to PC to A	1	2	83
MOVX	A,@Ri	Move external RAM (8-bit address) to A	1	2	E2, E3
MOVX	A,@DPTR	Move external RAM (16-bit address) to A	1	2	E0
MOVX	@Ri,A	Move A to external RAM (8-bit address)	1	2	F2, F3
MOVX	@DPTR,A	Move A to external RAM (16-bit address)	1	2	F0
PUSH	direct	Push direct byte onto stack	2	2	C0
POP	direct	Pop direct byte from stack	2	2	D0
XCH	A,Rr	Exchange register with A	1	1	C*
XCH	A,direct	Exchange direct byte with A	2	1	C5
XCH	A,@Ri	Exchange indirect RAM with A	1	1	C6, C7
XCHD	A,@Ri	Exchange LOW-order digit indirect RAM with A	1	1	D6, D7

** MOV A,ACC is not permitted.

mnemonic		description	bytes/	cycles	opcode (hex.)
Boolean variable manipulation					
CLR	C	Clear carry flag	1	1	C3
CLR	bit	Clear direct bit	2	1	C2
SETB	C	Set carry flag	1	1	D3
SETB	bit	Set direct bit	2	1	D2
CPL	C	Complement carry flag	1	1	B3
CPL	bit	Complement direct bit	2	1	B2
ANL	C,bit	AND direct bit to carry flag	2	2	82
ANL	C,/bit	AND complement of direct bit to carry flag	2	2	B0
ORL	C,bit	OR direct bit to carry flag	2	2	72
ORL	C,/bit	OR complement of direct bit to carry flag	2	2	A0
MOV	C,bit	Move direct bit to carry flag	2	1	A2
MOV	bit,C	Move carry flag to direct bit	2	2	92
Program and machine control					
ACALL	addr11	Absolute subroutine call	2	2	●1addr
LCALL	addr16	Long subroutine call	3	2	12
RET		Return from subroutine	1	2	22
RET1		Return from interrupt	1	2	32
AJMP	addr11	Absolute jump	2	2	▲1addr
LJMP	addr16	Long jump	3	2	02
SJMP	rel	Short jump (relative address)	2	2	80
JMP	@A+DPTR	Jump indirect relative to the DPTR	1	2	73
JZ	rel	Jump if A is zero	2	2	60
JNZ	rel	Jump if A is not zero	2	2	70
JC	rel	Jump if carry flag is set	2	2	40
JNC	rel	Jump if no carry flag	2	2	50
JB	bit,rel	Jump if direct bit is set	3	2	20
JNB	bit,rel	Jump if direct bit is not set	3	2	30
JBC	bit,rel	Jump if direct bit is set and clear bit	3	2	10
CJNE	A,direct,rel	Compare direct to A and jump if not equal	3	2	B5
CJNE	A,#data,rel	Compare immediate to A and jump if not equal	3	2	B4
CJNE	Rr,#data,rel	Compare immed. to reg. and jump if not equal	3	2	B*
CJNE	@Ri,#data,rel	Compare immed. to ind. and jump if not equal	3	2	B6, B7
DJNZ	Rr,rel	Decrement register and jump if not zero	2	2	D*
DJNZ	direct,rel	Decrement direct and jump if not zero	3	2	D5
NOP		No operation	1	1	00

Notes to Table 4

Data addressing modes

Rr	Working register R0-R7.
direct	128 internal RAM locations and any special function register (SFR).
@Ri	Indirect internal RAM location addressed by register R0 or R1 of the actual register bank.
#data	8-bit constant included in instruction.
#data16	16-bit constant included as bytes 2 and 3 of instruction.
bit	direct addressed bit in internal RAM or SFR.
addr16	16-bit destination address. Used by LCALL and LJMP. The branch will be anywhere within the 64 K-byte program memory address space.
addr11	11-bit destination address. Used by ACALL and AJMP. The branch will be within the same 2 K-byte page of program memory as the first byte of the following instruction.
rel	Signed (two's complement) 8-bit offset byte. Used by SJMP and all conditional jumps. Range is −128 to + 127 bytes relative to first byte of the following instruction.

Hexadecimal opcode cross-reference to Table 5

*: 8, 9, A, B, C, D, E, F.

●: 11, 31, 51, 71, 91, B1, D1, F1.

▲: 01, 21, 41, 61, 81, A1, C1, E1.

Table 5 Instruction map

first hexadecimal character of opcode ↓	second hexadecimal character of opcode →															
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NOP	AJMP page 0	LJMP addr 16	RR A	INC A	INC dir	INC @Ri 0	1	INC Rr 0	1	2	3	4	5	6	7
1	JBC bit,addr8	ACALL page 0	LCALL addr16	RRC A	DEC A	DEC dir	DEC @Ri 0	1	DEC Rr 0	1	2	3	4	5	6	7
2	JB bit,addr8	AJMP page 1	RET	RL A	ADD A,#data	ADD A,dir	ADD A,@Ri 0	1	ADD A,Rr 0	1	2	3	4	5	6	7
3	JNB bit,addr8	ACALL page 1	RET1	RLC A	ADDC A,#data	ADDC A,dir	ADDC A,@Ri 0	1	ADDC A,Rr 0	1	2	3	4	5	6	7
4	JC addr8	AJMP page 2	ORL dir,A	ORL dir,#data	ORL A,#data	ORL A,dir	ORL A,@Ri 0	1	ORL A,Rr 0	1	2	3	4	5	6	7
5	JNC addr8	ACALL page 2	ANL dir,A	ANL dir,#data	ANL A,#data	ANL A,dir	ANL A,@Ri 0	1	ANL A,Rr 0	1	2	3	4	5	6	7
6	JZ addr8	AJMP page 3	XRL dir,A	XRL dir,#data	XRL A,#data	XRL A,dir	XRL A,@Ri 0	1	XRL A,Rr 0	1	2	3	4	5	6	7
7	JNZ addr8	ACALL page 3	ORL C,bit	JMP @A+DPTR	MOV A, #data	MOV dir,#data	MOV @Ri,#data 0	1	MOV Rr,#data 0	1	2	3	4	5	6	7
8	SJMP addr8	AJMP page 4	ANL C,bit	MOVC A,@A+PC	DIV AB	MOV dir,dir	MOV dir,@Ri 0	1	MOV dir,Rr 0	1	2	3	4	5	6	7
9	MOV DPTR, #data 16	ACALL page 4	MOV bit,C	MOVC A,@A+DPTR	SUBB A,#data	SUBB A,dir	SUBB A,@Ri 0	1	SUBB A,Rr 0	1	2	3	4	5	6	7
A	ORL C,/bit	AJMP page 5	MOV C,bit	INC DPTR	MUL AB		MOV @Ri,dir 0	1	MOV Rr,dir 0	1	2	3	4	5	6	7
B	ANL C,/bit	ACALL page 5	CPL bit	CPL C	CJNE A, #data,addr8	CJNE A,dir,addr8	CJNE @Ri, #data,addr8 0	1	CJNE Rr,#data,addr8 0	1	2	3	4	5	6	7
C	PUSH dir	AJMP page 6	CLR bit	CLR C	SWAP A	XCH A,dir	XCH A,@Ri 0	1	XCH A,Rr 0	1	2	3	4	5	6	7
D	POP dir	ACALL page 6	SETB bit	SETB C	DA A	DJNZ dir,addr8	XCHD A,@Ri 0	1	DJNZ Rr,addr8 0	1	2	3	4	5	6	7
E	MOVX A,@DPTR	AJMP page 7	MOVX A,@Ri 0	1	CLR A	MOV A,dir	MOV A,@Ri 0	1	MOV A,Rr 0	1	2	3	4	5	6	7
F	MOVX @DPTR,A	ACALL page 7	MOVX @Ri,A 0	1	CPL A	MOV dir,A	MOV @Ri,A 0	1	MOV Rr,A 0	1	2	3	4	5	6	7

RATINGS

Limiting values in accordance with the Absolute Maximum System (IEC 134)

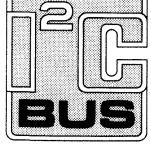
Input voltage on any pin with respect to ground (V_{SS})	V_I	−0,5 to +7	V
Input, output current	$\pm I_I, I_O$	max. 5	mA
Total power dissipation	P_{tot}	max. 1	W
Storage temperature range	T_{stg}	−65 to +150	°C
Operating ambient temperature range	T_{amb}	0 to +70	°C

DEVELOPMENT DATA

D.C. CHARACTERISTICS

V_{DD} = 5 V (±10%); V_{SS} = 0 V; T_{amb} = 0 to +70 °C; all voltages with respect to V_{SS} unless otherwise specified

parameter	symbol	min.	max.	unit	conditions
Supply voltage	V_{DD}	4,5	5,5	V	
Supply current					
operating (note 1)	I_{DD}	–	tbf	mA	f_{CLK} = 12 MHz
idle mode (note 2)	I_{DD}	–	tbf	mA	f_{CLK} = 12 MHz
Power-down current	I_{PD}	–	tbf	µA	V_{DD} = 2 V (note 3)
Inputs (see note 6)					
LOW level input voltage (except $\overline{EA}$, P1.6/SCL, P1.7/SDA	V_{IL}	−0,5	$0{,}2V_{DD}-0{,}1$	V	
LOW level input voltage ($\overline{EA}$)	V_{IL1}	−0,5	$0{,}2V_{DD}-0{,}3$	V	
LOW level input voltage (P1.6/SCL, P1.7/SDA)	V_{IL2}	−0,5	1,5	V	
HIGH level input voltage (except XTAL 1, RST, P1.6/SCL, P1.7/SDA)	V_{IH}	$0{,}2V_{DD}+0{,}9$	$V_{DD}+0{,}5$	V	
HIGH level input voltage (XTAL 1, RST)	V_{IH1}	$0{,}7V_{CC}$	$V_{DD}+0{,}5$	V	
HIGH level input voltage (P1.6/SCL, P1.7/SDA)	V_{IH2}	3,0	$V_{DD}+0{,}5$	V	
Input current logic 0 (Ports 1, 2, 3 and 4; except P1.6/SCL, P1.7/SDA)	$-I_{IL}$	–	50	µA	V_I = 0,45 V
Input current logic 1 to 0 transition (Ports 1, 2, 3 and 4; except P.1/SCL, P1.7/SDA)	$-I_{TL}$	–	500	µA	V_I = 2 V
Input leakage current (Port 0, $\overline{EA}$, STADC,EWN, P1.6/SCL, P1.7/SDA)	$\pm I_{LI}$	–	10	µA	$0{,}45\,V < V_I < V_{DD}$



Purchase of Philips' I²C components conveys a license under the Philips' I²C patent to use the components in the I²C system provided the system conforms to the I²C specification defined by Philips.

D.C. CHARACTERISTICS (continued)

parameter	symbol	min.	max.	unit	conditions
Outputs					
LOW level output voltage (note 4) (Ports 1, 2, 3 and 4; except P1.6/SCL, P1.7/SDA)	V_{OL}	–	0,45	V	I_{OL} = 1,6 mA
LOW level output voltage (note 4) (Port 0, ALE, $\overline{PSEN}$)	V_{OL1}	–	0,45	V	I_{OL} = 3,2 mA
LOW level output voltage (P1.6/SCL, P1.7/SCL)	V_{OL2}	–	0,4	V	I_{OL} = 3,0 mA
HIGH level output voltage (Ports 1, 2, 3 and 4; except P1.6/SCL, P1.7/SDA)	V_{OH}	2,4	–	V	$-I_{OH}$ = 80 μA; V_{DD} = 5 V ± 10%
		$0{,}75V_{DD}$	–	V	$-I_{OH}$ = 30 μA
		$0{,}9V_{DD}$	–	V	$-I_{OH}$ = 10 μA
HIGH level output voltage (note 5) (Port 0 in external Bus mode, ALE, $\overline{PSEN}$)	V_{OH1}	2,4	–	V	$-I_{OH}$ = 400 μA; V_{DD} = 5 V ± 10%
		$0{,}75V_{DD}$	–	V	$-I_{OH}$ = 150 μA
		$0{,}9V_{DD}$	–	V	$-I_{OH}$ = 40 μA
RST pull-down resistor	R_{RST}	40	125	kΩ	
I/O pin capacitance	$C_{I/O}$	–	10	pF	test freq. = 1 MHz; T_{amb} = 25 °C

Notes to the d.c. characteristics

1. The operating supply current is measured with all output pins disconnected; XTAL 1 driven with $t_r = t_f$ = 10 ns; $V_{IL} = V_{SS}$ + 0,5 V, $V_{IH} = V_{DD}$ –0,5 V; XTAL 2 not connected; $\overline{EA}$ = RST = Port 0 = V_{DD}.
2. The idle mode supply current is measured with all output pins disconnected; XTAL 1 driven with $t_r = t_f$ = 10 ns, $V_{IL} = V_{SS}$ + 0,5 V, $V_{IH} = V_{DD}$ –0,5 V; XTAL 2 not connected; $\overline{EA}$ = Port 0 = V_{DD}; RST = V_{SS}.
3. The power-down current is measured with all output pins disconnected; XTAL 2 not connected; $\overline{EA}$ = Port 0 = V_{DD}; RST = V_{SS}.
4. Capacitive loading on Port 0 and Port 2 may cause spurious noise pulses to be superimposed on the LOW level output voltage of ALE, Port 1 and Port 3. The noise is due to external Bus capacitance discharging into the Port 0 and Port 2 pins when these pins make a 1-to-0 transition during Bus operations. In the most adverse condition (capacitive loading > 100 pF) the noise pulse on ALE line may exceed 0,8 V. In this event it may be required to qualify ALE with a Schmitt trigger, or use an address latch with a Schmitt trigger STROBE input.
5. Capacitive loading on Port 0 and Port 2 may cause the HIGH level output voltage on ALE and $\overline{PSEN}$ to momentarily fall below the 0,9 V_{DD} specification when the address bits are stabilizing.
6. The input threshold voltage of P1.6 and P1.7 (SIO1) meets the I^2C specification, so an input voltage below 1,5 V will be recognized as a logic '0' while an input voltage above 3,0 V will be recognized as a logic 1.

A.C. CHARACTERISTICS

V_{DD} = 5 V ± 10%; V_{SS} = 0 V; T_{amb} = 0 to +70 °C; C_L = 100 pF (Port 0, ALE and $\overline{PSEN}$); C_L = 80 pF (all other outputs); unless otherwise specified (see waveforms Figs 16, 17 and 18)

parameter	symbol	10 MHz		12 MHz		variable clock		unit
		min.	max.	min.	max.	min.	max.	
Program memory								
ALE pulse duration	t_{LL}	160	–	127	–	$2t_{CK}-40$	–	ns
Address set-up time to ALE	t_{AL}	60	–	43	–	$t_{CK}-40$	–	ns
Address hold time after ALE	t_{LA}	65	–	48	–	$t_{CK}-35$	–	ns
Time from ALE to valid instruction input	t_{LIV}	–	300	–	233	–	$4t_{CK}-100$	ns
Time from ALE to control pulse $\overline{PSEN}$	t_{LC}	75	–	58	–	$t_{CK}-25$	–	ns
Control pulse duration $\overline{PSEN}$	t_{CC}	265	–	215	–	$3t_{CK}-35$	–	ns
Time from $\overline{PSEN}$ to valid instruction input	t_{CIV}	–	175	–	125	–	$3t_{CK}-125$	ns
Input instruction hold time after $\overline{PSEN}$	t_{CI}	0	–	0	–	0	–	ns
Input instruction float delay after $\overline{PSEN}$*	t_{CIF}	–	80	–	63	–	$t_{CK}-20$	ns
Address valid after $\overline{PSEN}$*	t_{AC}	92	–	75	–	$t_{CK}-8$	–	ns
Address to valid instruction input	t_{AIV}	–	385	–	302	–	$5t_{CK}-115$	ns
Address float time to $\overline{PSEN}$	t_{AFC}	−12	–	−12	–	0	–	ns

* Interfacing the PCB83C652 to devices with float times up to 75 ns is permitted. This limited bus contention will not cause damage to Port 0 drivers.

A.C. CHARACTERISTICS (continued)

parameter	symbol	10 MHz		12 MHz		variable clock		unit
		min.	max.	min.	max.	min.	max.	
External data memory								
$\overline{RD}$ pulse duration	t_{RR}	500	–	400	–	$6t_{CK}-100$	–	ns
$\overline{WR}$ pulse duration	t_{WW}	500	–	400	–	$6t_{CK}-100$	–	ns
Address hold time after ALE	t_{LA}	65	–	48	–	$t_{CK}-35$	–	ns
$\overline{RD}$ to valid data input	t_{RD}	–	335	–	250	–	$5t_{CK}-165$	ns
Data hold time after $\overline{RD}$	t_{DR}	0	–	0	–	0	–	ns
Data float delay after $\overline{RD}$	t_{DFR}	–	130	–	97	–	$2t_{CK}-70$	ns
Time from ALE to valid data input	t_{LD}	–	650	–	517	–	$8t_{CK}-150$	ns
Address to valid data input	t_{AD}	–	735	–	585	–	$9t_{CK}-165$	ns
Time from ALE to $\overline{RD}$ or $\overline{WR}$	t_{LW}	250	350	200	300	$3t_{CK}-50$	$3t_{CK}+50$	ns
Time from address to $\overline{RD}$ or $\overline{WR}$	t_{AW}	270	–	203	–	$4t_{CK}-130$	–	ns
Time from $\overline{RD}$ or $\overline{WR}$ HIGH to ALE HIGH	t_{WHLH}	60	140	43	123	$t_{CK}-40$	$t_{CK}+40$	ns
Data valid to $\overline{WR}$ transition	t_{DWX}	40	–	23	–	$t_{CK}-60$	–	ns
Data set-up time before $\overline{WR}$	t_{DW}	550	–	433	–	$7t_{CK}-150$	–	ns
Data hold time after $\overline{WR}$	t_{WD}	50	–	33	–	$t_{CK}-50$	–	ns
Address float delay after $\overline{RD}$	t_{AFR}	–	12	–	12	–	12	ns

Where:

$1/t_{CK}$ = 3,5 to 12 MHz (see Fig. 15 and Table 6).

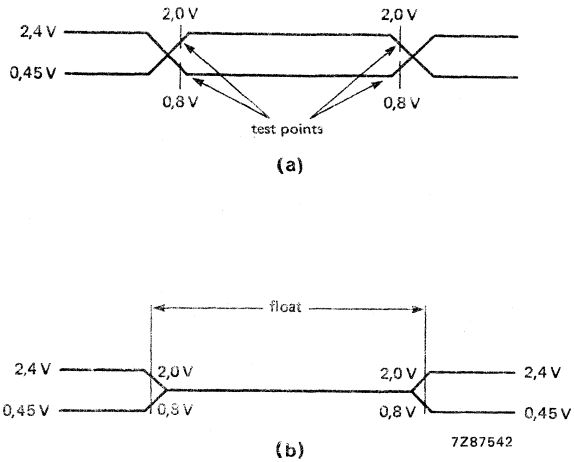


Fig. 14 A.C. testing input, output waveform (a) and float waveform (b).

A.C. testing inputs are driven at 2,4 V for a logic 1 and 0,45 V for a logic 0. Timing measurements are taken at 2,0 V for a logic 1 and 0,8 V for logic 0. The float state is defined as the point at which a Port 0 pin sinks 3,2 mA or sources 400 μA at the voltage test levels.

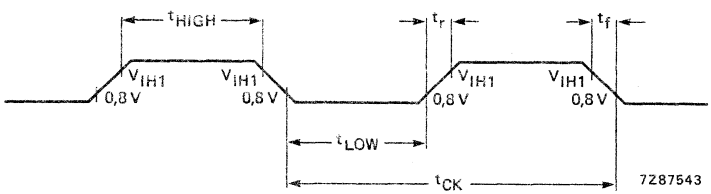


Fig. 15 External clock drive XTAL 1 (see Table 6).

Table 6 External clock drive XTAL 1 (see Fig. 15)

parameter	symbol	variable clock (f = 3,5 to 12 MHz)		unit
		min.	max.	
oscillator clock period	t_{CK}	83,3	286	ns
HIGH time	t_{HIGH}	20	$t_{CK}-t_{LOW}$	ns
LOW time	t_{LOW}	20	$t_{CK}-t_{HIGH}$	ns
rise time	t_r	–	20	ns
fall time	t_f	–	20	ns

A.C. CHARACTERISTICS (continued)

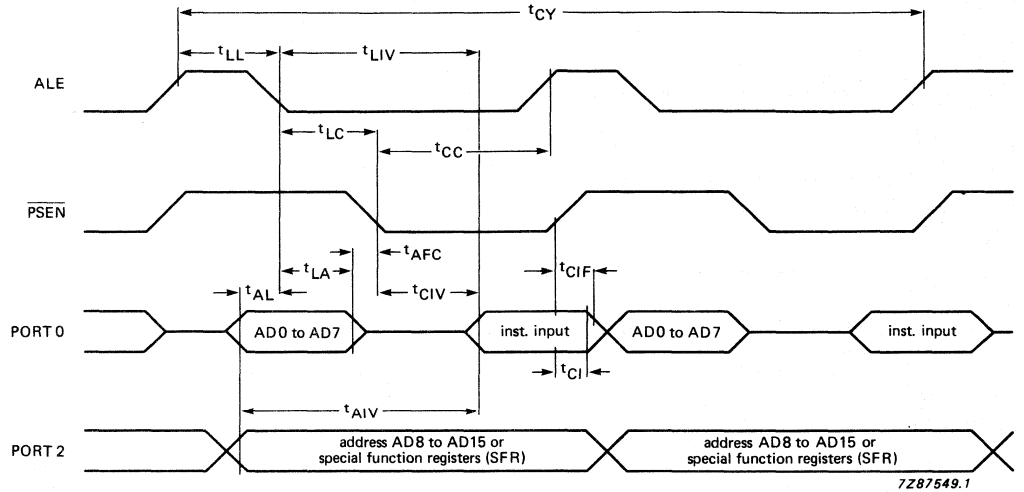


Fig. 16 Read from program memory.

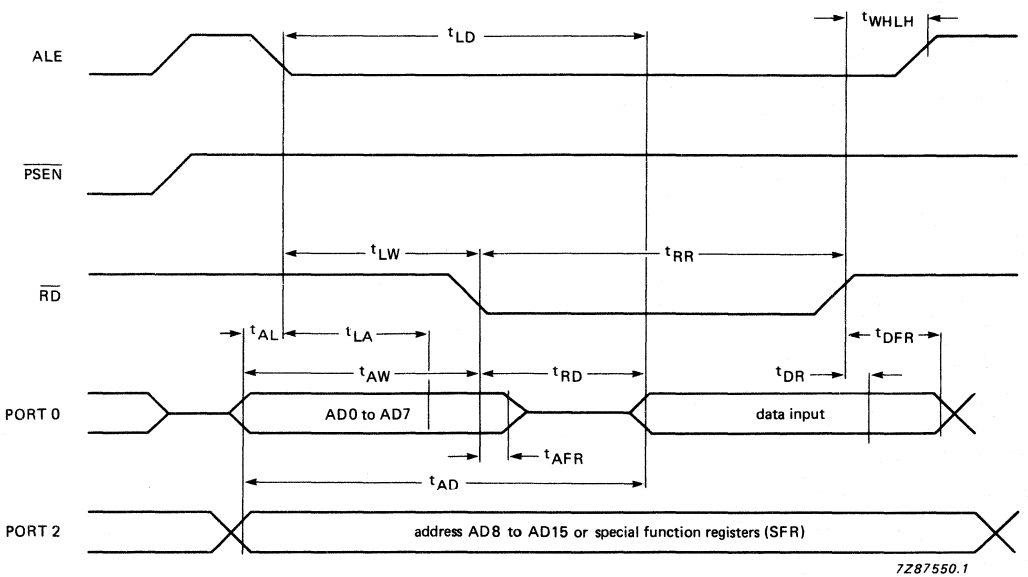


Fig. 17 Read from data memory.

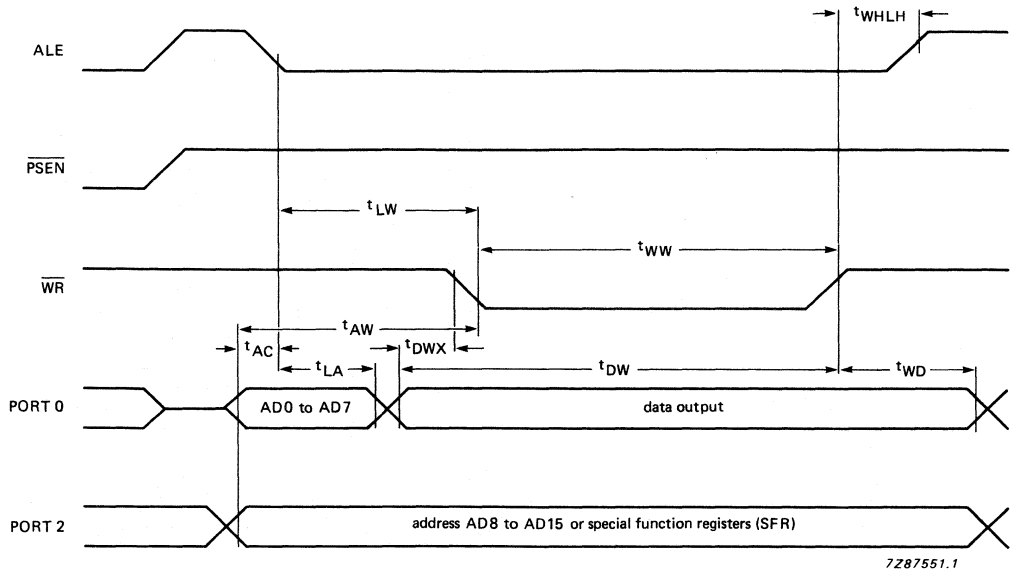


Fig. 18 Write to data memory.

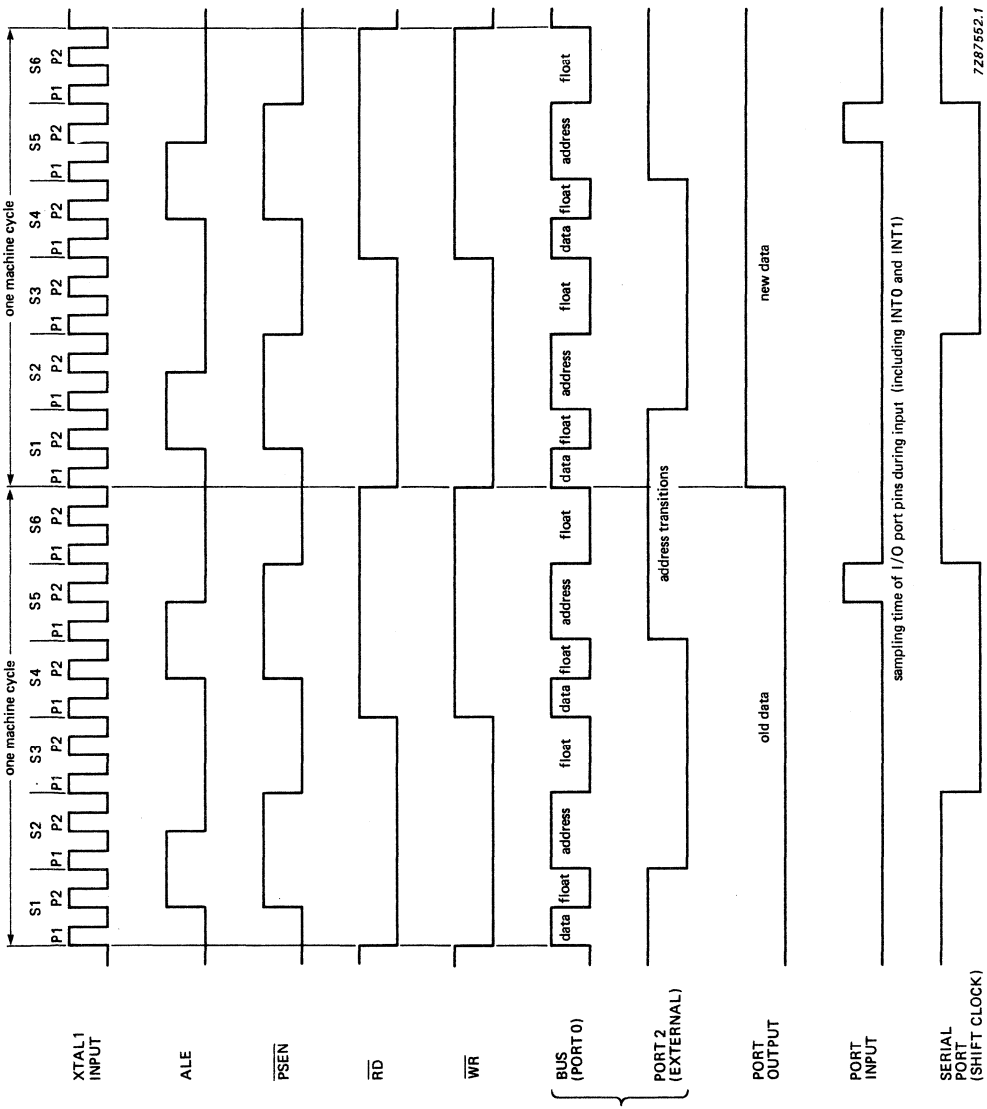


Fig. 19 Instruction cycle timing.

SINGLE-CHIP 8-BIT MICROCONTROLLER

DESCRIPTION

The PCB85C51 is the software development version of the PCB80C51 8-bit microcontroller family. The PCB85C51, manufactured in CMOS technology, is chiefly used for the software/hardware development of PCB80C51 applications. The 85C51 has the same electrical parameters as the 80C51 and has an identical instruction set. The 85C51 contains no internal ROM but can address externally as 'quasi-internal', up to 16 K of program memory via its 14 address and 8 data lines. Lines M1/M2 operate as memory control, the signal $\overline{\text{PSENE}}$ serves as memory chip enable. The PCB85C51 does contain a volatile data memory RAM of 128 x 8 bytes; 32 I/O lines; two 16-bit timer/event counters; a five-source, two-priority-level, nested interrupt structure; a serial I/O port for either multi-processor communications, I/O expansion, or full duplex UART; and on-chip oscillator and timing circuits. For systems that require extra capability, the PCB85C51 can be expanded using standard TTL compatible memories and logic.

The PCB85C51 has two software selectable modes of reduced activity for further power reduction — Idle and Power Down.
The Idle mode freezes the CPU while allowing the RAM, timers, serial port and interrupt system to continue functioning.
The Power Down mode saves the RAM contents but freezes the oscillator causing all other chip functions to be inoperative.

The device also functions as an arithmetic processor having facilities for both binary and BCD arithmetic plus bit-handling capabilities. The instruction set consists of 255 instructions; 44% one-byte, 41% two-byte and 15% three-byte. With a 12 MHz crystal, 58% of the instructions are executed in 1 μs and 40% in 2 μs. Multiply and divide instructions require 4 μs. Multiply, divide, subtract and compare are among the many instructions added to the standard PCB80C49 instruction set.

Features

- 128 x 8 RAM
- Four 8-bit ports, 32 I/O lines
- Two 16-bit timer/event counters
- Full-duplex serial port
- External memory expandable to 128 K, external ROM to 64 K and/or external RAM up to 64 K
- Boolean processing
- 218 bit-addressable locations
- On-chip oscillator
- Five-source interrupt structure with two priority levels
- 58% of instructions executed in 1 μs; multiply and divide in 4 μs; all other executed in 2 μs (at 12 MHz clock)
- Enhanced architecture with:
  - non-page-oriented instructions
  - direct addressing
  - four 8-byte + 1 byte register banks
  - stack depth up to 128-bytes
  - multiply, divide, subtract and compare instructions.

PACKAGE OUTLINES

PCB85C51B: 40-lead DIL piggy-back, metal ceramic (CERDIL) (SOT-215).

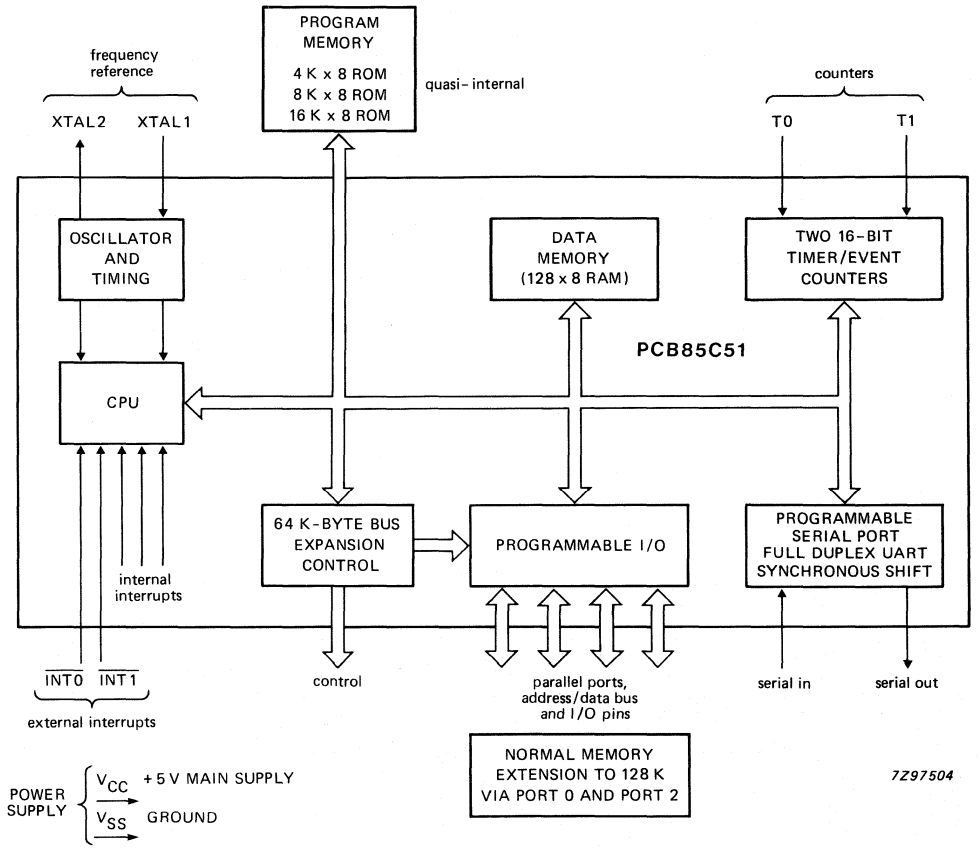


Fig. 1 PCB85C51 block diagram.

DEVELOPMENT DATA

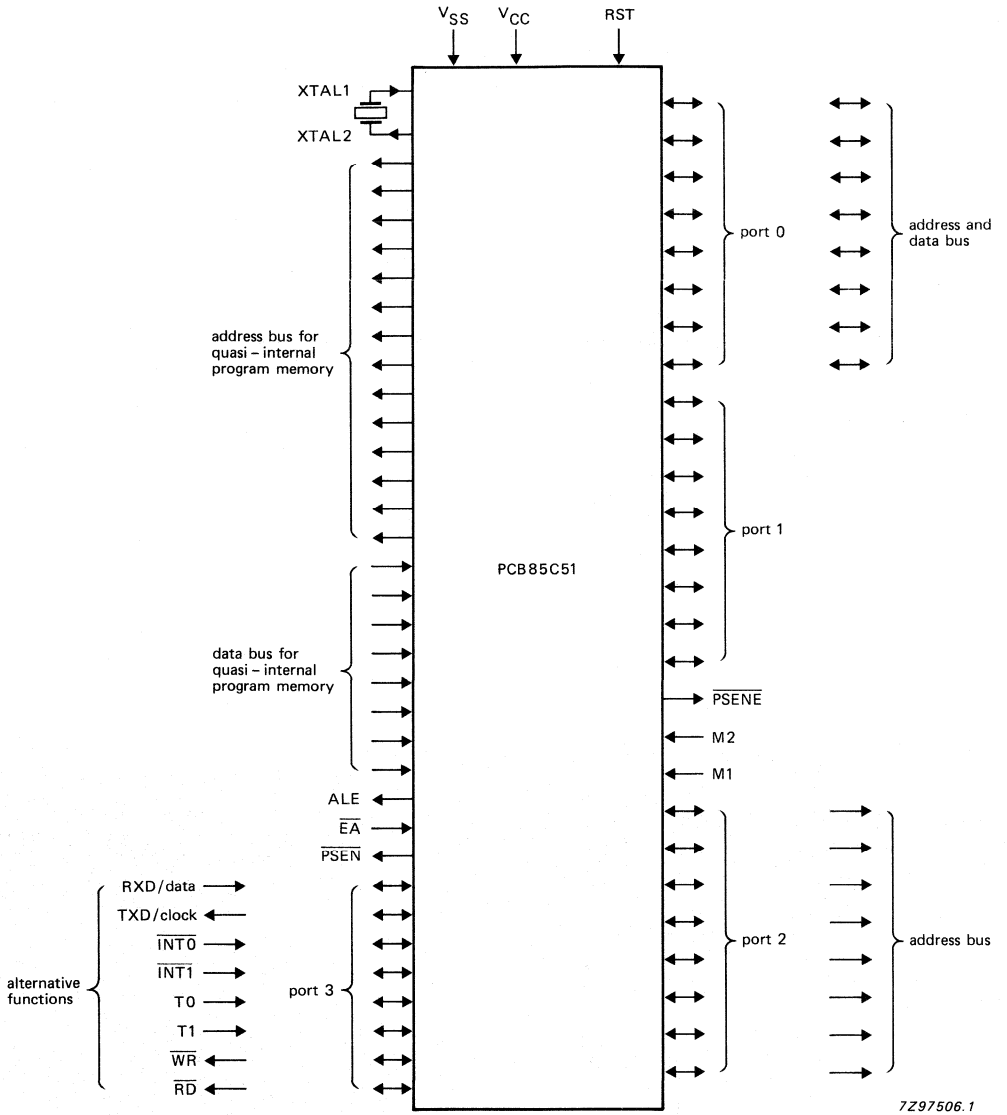


Fig. 2 Functional diagram of PCB85C51.

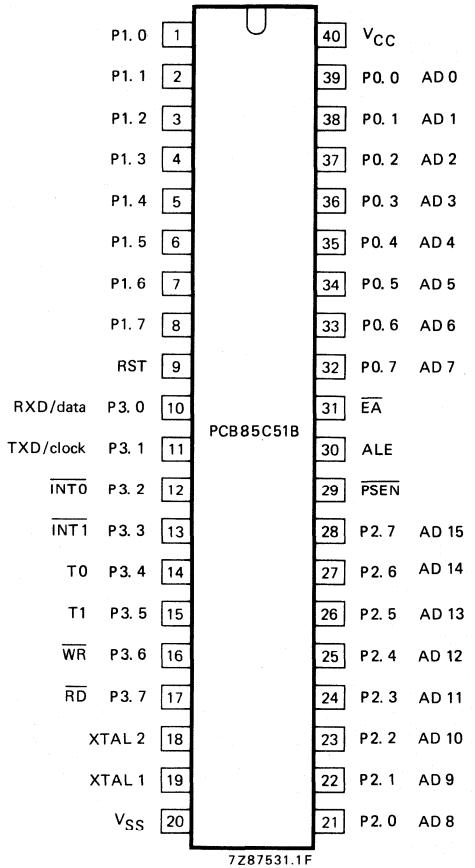


Fig. 3a Bottom pinning diagram for PCB85C51B.

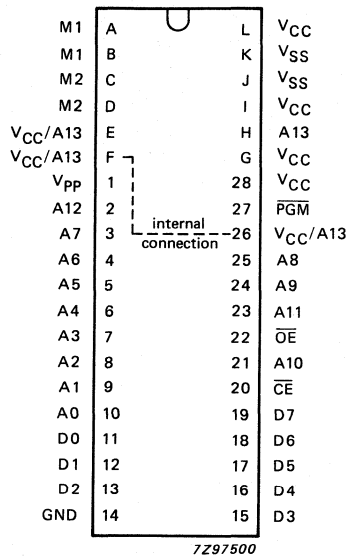


Fig. 3b Top pinning on 40-pin EPROM socket for PCB85C51B.

DEVELOPMENT DATA

PIN DESIGNATION (PCB85C51B)

1-8	P1.0-P1.7	**Port 1**: 8-bit quasi-bidirectional I/O port. Port 1 can sink/source one TTL (= 4 LS TTL) input. It can drive CMOS inputs without external pull-ups.
9		**RST**: a high level on this pin for two machine cycles while the oscillator is running resets the device. An internal pull-down permits Power-on reset using only a capacitor connected to V_{CC}.
10-17	P3.0-P3.7	**Port 3**: 8-bit quasi-bidirectional I/O port with internal pull-ups. It also serves the following alternative functions:

Port pin	Alternative function
P3.0	RXD/data: serial port receiver data input (asynchronous) or data input/output (synchronous)
P3.1	TXD/clock: serial port transmitter data output (asynchronous) or clock output (synchronous)
P3.2	INT0: external interrupt 0 or gate control input for timer/event counter 0
P3.3	INT1: external interrupt 1 or gate control input for timer/event counter 1
P3.4	T0: external input for timer/event counter 0
P3.5	T1: external input for timer/event counter 1
P3.6	WR: external data memory write strobe
P3.7	RD: external data memory read strobe

Operation of an alternative function is determined by the relevant output latch programmed to logic 1. Port 3 can sink/source one TTL input. It can drive CMOS inputs without external pull-ups.

18	XTAL 2	**Crystal input 2**: output of the inverting amplifier that forms the oscillator. Left open-circuit when an external oscillator is used (see Figs 8 and 9).
19	XTAL 1	**Crystal input 1**: input to the inverting amplifier that forms the oscillator, and input to the internal clock generator. Receives the external oscillator signal when an external oscillator is used (see Figs 8 and 9).
20	V_{SS}	**Ground**: circuit ground potential.
21-28	P2.0-P2.7	**Port 2**: 8-bit quasi-bidirectional I/O port with internal pull-ups. Port 2 can sink/source one TTL input. It can drive CMOS inputs without external pull-ups. During the access to external memories (RAM/ROM) that use 16-bit addresses (MOVX @DPTR) Port 2 emits the high order address byte. When used for an external RAM, an 8-bit address (MOVX @Ri) Port 2 emits the contents of P2 Special Function Register.
29	$\overline{\text{PSEN}}$	**Program Store Enable output**: read strobe to the external Program Memory. It is activated twice each machine cycle during fetches from external Program Memory. When executing out of external Program Memory two activations of $\overline{\text{PSEN}}$ are skipped during each access to external Data Memory. $\overline{\text{PSEN}}$ is not activated (remains HIGH) during no fetches from external Program Memory. $\overline{\text{PSEN}}$ can sink/source 8 LS TTL inputs. It can drive CMOS inputs without an external pull-up.
30	ALE	**Address Latch Enable output**: latches the low byte of the address during accesses to external memory in normal operation. It is activated every six oscillator periods except during an external data memory access. ALE can sink/source 8 LS TTL inputs. It can drive CMOS inputs without an external pull-up.

31	$\overline{EA}$	**External Access input**: When $\overline{EA}$ is held at a TTL high level the CPU executes out of the internal Program Memory (ROM), provided the Program Counter is less than the boundary selected by M1, M2. When $\overline{EA}$ is held at a TTL low level, the CPU executes out of external Program Memory. $\overline{EA}$ is not allowed to float.
32-39	P0.7-P0.0	**Port 0**: 8-bit open drain bidirectional I/O port, it is also the multiplexed low-order address and data bus during accesses to external memory (during these accesses it activates internal pull-ups). Port 0 can sink/source eight LSTTL inputs.
40	V_{CC}	**Power supply**: + 5 V power supply pin during normal operation, Idle mode and Power Down mode.

PIN DESIGNATION (Non-standard)

10-2 26-23, 21	A0-A7, A12 A13, A8, A9, A11 A10	**Address Bus**: for quasi-internal program memory
11-13, 15-19	D0-D2 D3-D7	**Data Bus**: for quasi-internal program memory
20 ($\overline{CE}$)	$\overline{PSENE}$	**Chip enable**: signal for external ROM connected via A0-A13 and D0-D7 (Active LOW). Available twice per machine cycle, timing same as user $\overline{PSEN}$.*
CD,AB	M2,M1	**Signals**: set to control the switch from 'quasi-internal' to external program memory. Signals should only be changed during reset.

M2	M1	
0	1	4 K Memory (80C51)
1	0	8 K Memory
1	1	16 K Memory
0 = V_{SS}		1 = V_{CC}

* An exception to this is when user $\overline{PSEN}$ remains HIGH during RESET, when the chip executes from internal program memory or is in the second cycle of a MOVX instruction.

DEVELOPMENT DATA

FUNCTIONAL DESCRIPTION

General

The PCB85C51 is designed for software development in real-time applications such as instrumentation, industrial control and intelligent computer peripherals.

The device provides the same hardware features, architectural enhancements and new instructions as the PCB80C51. The device can address up to 64 K bytes of program memory and/or up to 64 K bytes of data storage.

The PCB85C51 has a control-oriented CPU without on-chip program memory. Lines M1 and M2 operate as memory control. Below the boundary marked by M1 and M2 the opcodes are fetches out of the quasi-internal ROM. Above this boundary the ROM can be expanded up to 64 K as normal via the ports 0 and 2. The signal $\overline{\text{PSENE}}$ serves as memory chip enable. The PCB85C51 has the lower 4, 8 or 16 K bytes of program memory connected via A0-A13 and D0-D7. For systems requiring extra capability, the PCB85C51 can be expanded using standard memories and peripherals.

This bond-out version of the 80C51 reduces development problems to a minimum and provides maximum flexibility.

The PCB85C51 contains no on-chip program ROM, but does contain a volatile 128 x 8 read/write data memory; 32 I/O lines; two 16-bit timer/event counters; a five-source, two-priority-level nested interrupt structure; a serial I/O port for either multi-processor communications, I/O expansion or full duplex UART communications; and on-chip oscillator and timing circuits.

Central processing unit

The central processing unit (CPU) manipulates operands in four memory spaces. These are the 64 K-byte external data memory, 128-byte internal data memory, the 64 K-byte internal and external program memory. The internal data memory address space is sub-divided into the 128-byte internal data RAM and 128-byte special register (SFR) address spaces, as shown in Fig. 4.

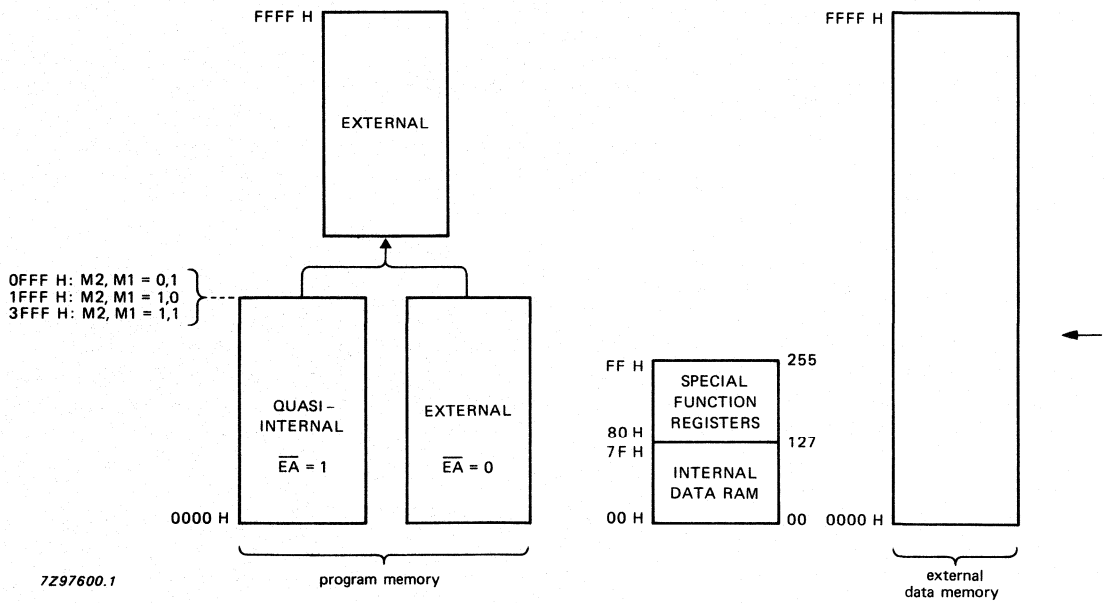


Fig. 4 Memory map.

FUNCTIONAL DESCRIPTION (continued)

The internal data RAM contains four register banks (each with eight registers), 128 addressable bits, and the stack. The stack depth is limited by the available internal data RAM and its location is determined by the 8-bit stack pointer. All registers except the program counter and the four 8-register banks reside in the special function register address space. These memory mapped registers include arithmetic registers, pointers, I/O ports, interrupt system registers, timers and serial port. There are 128 bit addressable locations in the SFR address space.

The PCB85C51 contains 128 bytes of internal data RAM and 20 special function registers. It provides a non-paged program memory address space to accommodate relocatable code. Conditional branches are performed relative to the program counter. The register-indirect jump permits branching relative to a 16-bit base register with an offset provided by an 8-bit index register. 16-bit jumps and calls permit branching to any location in the contiguous 64 K program memory address space.

The PCB85C51 has five methods for addressing source operands:

- Register.
- Direct.
- Register-Indirect.
- Immediate.
- Base-register-plus Index-Register-Indirect.

The first three methods can be used for addressing destination operands. Most instructions have a "destination/source" field that specifies the data type, addressing methods and operands involved. For operations other than moves, the destination operand is also a source operand.

Access addressing is as follows:

- Registers in the four 8-register banks through Register, Direct, or Register-Indirect.
- 128 bytes of internal data RAM through Direct or Register-Indirect.
- Special function registers through Direct at address locations 128-255.
- External data memory through Register-Indirect.
- Program memory look-up tables through Base-Register-plus Index-Register-Indirect.

The PCB85C51 is classified as an 8-bit device since the external ROM, RAM, Special Function Registers (SFR), Arithmetic Logic Unit (ALU), and external data bus are each 8-bits wide. It performs operations on bit, nibble, byte and double-byte data types.

Facilities are available for byte transfer, logic, and integer arithmetic operations. Data transfer, logic, and conditional branch operations can be performed directly on Boolean variables to provide excellent bit handling.

Instruction set

The PCB85C51 uses the same powerful instruction set as the 80C51 to allow expansion of on-chip CPU peripherals and to optimize byte efficiency and execution speed. Reassigned opcodes add new high-power operations and permit new addressing modes to make old operations more orthodox when compared to the 8048 family. The instruction set consists of 49 single-byte, 45 two-byte and 17 three-byte instructions. When using a 12 MHz oscillator, 64 instructions execute in 1 μs and 45 instructions execute in 2 μs. Multiply and divide instructions execute in 4 μs.

I/O facilities

The PCB85C51 has 32 I/O lines treated as 32 individual addressable bits and as four parallel 8-bit addressable ports. Ports 0, 1, 2 and 3 perform the following alternate functions:

- Port 0; provides the multiplexed low-order address and data bus used for expanding the PCB85C51 with standard memories and peripherals.
- Port 2; provides the high-order address bus when expanding the PCB85C51 with external program memory or more than 256 bytes of external data memory.

The address boundary for external access to program memory via port 0 and 2, is set by the signal on M1 and M2.

- Port 3; pins can be configured individually to provide:
  - external interrupt request inputs
  - counter inputs
  - serial port receiver input and transmitter output
  - control signals to READ and WRITE to external data memory.

The generation or use of a Port 3 pin as an alternate function is carried out automatically by the PCB85C51 provided the pin is loaded with a HIGH content.

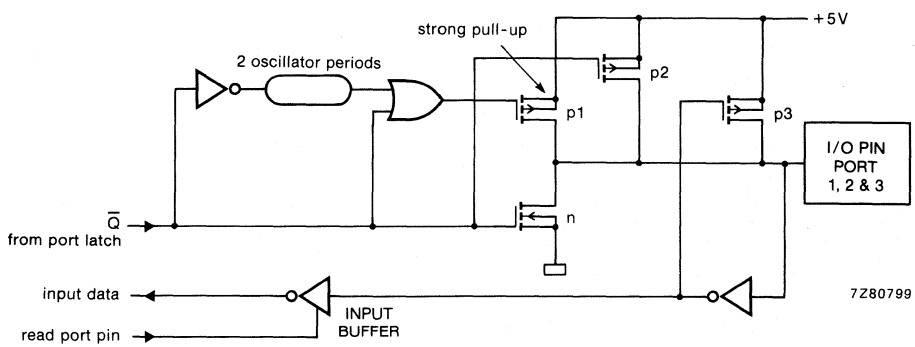


Fig. 5 I/O buffers in the PCB85C51 (Ports 1, 2 and 3).

Timer/event counters

The PCB85C51 contains two 16-bit registers, Timer 0 and Timer 1, that can be used as timers or event counters to carry out the following functions:

- Measure time intervals and pulse durations.
- Count events.
- Generate interrupt requests.

Each timer/event counter can be programmed independently to operate in three modes:

- Mode 0; 8-bit timer or 8-bit counter each with divide by 32 prescaler.
- Mode 1; 16-bit time-interval or event counter.
- Mode 2; 8-bit time-interval or event counter with automatic reload upon overflow.

Counter 0 can be programmed to operate in an additional mode as follows:

- Mode 3; one 8-bit time-interval or event counter and one 8-bit time-interval counter.

FUNCTIONAL DESCRIPTION (continued)

When counter 0 is in Mode 3, counter 1 can be programmed to operate in Modes 0, 1 or 2 but cannot set an interrupt request flag or generate an interrupt. However, the overflow from counter 1 can be used to pulse the serial Port transmission-rate generator.

The frequency handling range of these counters with a 12 MHz crystal is as follows:

- Up to 1 MHz when programmed for an input that is a division by 12 of the oscillator frequency.
- 0 Hz to an upper limit of 150 kHz to 0,5 MHz when programmed for external inputs.

Both internal and external inputs can be gated to the counter by a second external source for directly measuring pulse durations.

The counters are started and stopped under software control. Each one sets its interrupt request flag when it overflows from all 1's to all 0's (or automatic reload value).

On-chip peripheral functions

In addition to the CPU and memories, an interrupt system, extensive I/O facilities, and several peripheral functions are integrated on-chip to relieve the CPU of repetitious, complicated or time-critical tasks and to permit stringent real-time control of external system interfaces. The I/O facilities include the I/O pins, parallel ports, bidirectional address/data bus and the serial port for I/O expansion. The CPU peripheral functions integrated on-chip are the two 16-bit timer/event counters and the serial port.

Idle and Power Down operation (see Fig. 6)

The Power Down operation freezes the oscillator. The Idle mode operation allows the interrupt, serial port and timer blocks to continue to function while the clock to the CPU is halted.

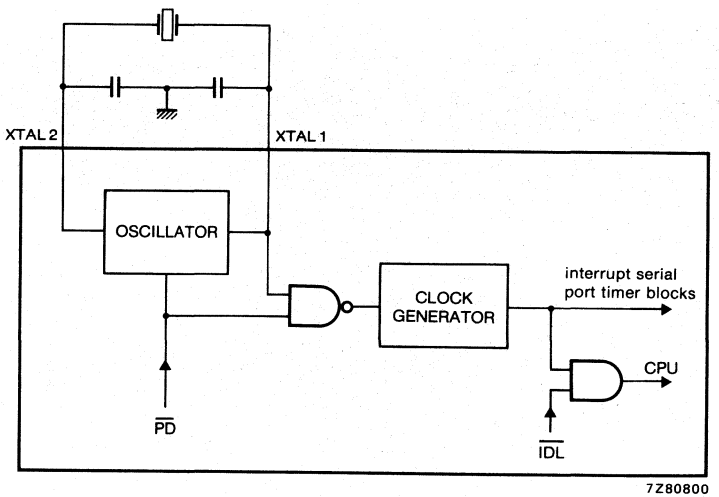


Fig. 6 Internal Idle and Power Down clock configuration.

Power control register (PCON)

These special modes are activated by software via the Special Function Register PCON. Its hardware address is 87H. PCON is not bit addressable.

MSB							LSB
7	6	5	4	3	2	1	0
SMOD	–	–	–	GF1	GF0	PD	IDL

symbol	position	name and function
SMOD	PCON.7	Double Baud rate bit when set to logic 1 the baud rate is doubled when the serial port is being used in either modes 1, 2 or 3
–	PCON.6	(reserved)
–	PCON.5	(reserved)
–	PCON.4	(reserved)
GF1	PCON.3	general-purpose flag bit
GF0	PCON.2	general-purpose flag bit
PD	PCON.1	Power-down bit setting this bit activates power-down operation
IDL	PCON.0	Idle mode bit setting this bit activates the idle mode operation

If a logic '1' is written to PD and IDL at the same time, PD takes precedence. The reset value of PCON is (0XXX000).

Idle mode

The instruction that sets PCON.0 is the last instruction executed in the normal operating mode before Idle mode is activated. Once in the Idle mode, the CPU status is preserved in its entirety: the Stack Pointer, Program Counter, Program Status Word, Accumulator, RAM and all other registers maintain their data during Idle mode. The status of the external pins during Idle mode is shown in Table 1.

There are two ways to terminate the Idle mode:

Activation of any enabled interrupt will cause PCON.0 to be cleared by hardware terminating Idle mode. The interrupt is serviced, and following return from interrupt instruction RETI, the next instruction to be executed will be the one which follows the instruction that wrote a logic 1 to PCON.0.

The flag bits GF0 and GF1 may be used to determine whether the interrupt was received during normal execution or during the Idle mode. For example, the second instruction that writes to PCON.0 can also set or clear one or both flag bits. When Idle mode is terminated by an interrupt, the service routine can examine the status of the flag bits.

The second way of terminating the Idle mode is with a hardware reset. Since the oscillator is still running, the hardware reset is required to be active for two machine cycles (24 oscillator periods) to complete the reset operation.

FUNCTIONAL DESCRIPTION (continued)

Power Down mode

The instruction that sets PCON.1 is the last executed prior to going into the Power Down mode. Once in Power Down mode, the oscillator is stopped. Only the contents of the on-chip RAM are preserved. The Special Function Registers are not saved. A hardware reset is the only way of exiting the Power Down mode.

In the Power Down mode, V_{CC} may be reduced to minimize circuit power consumption. The voltage must not be reduced until the Power Down mode is entered, but must be restored before the hardware reset is applied which will free the oscillator. Reset should not be released until the oscillator has restarted and stabilized.

The status of the external pins during Power Down mode is shown in Table 1. If the Power Down mode is activated while in external program memory, the port data that is held in the Special Function Register P2 is restored to Port 2. If the data is a logic 1, the port pin is held HIGH during the Power Down mode by the strong pull-up transistor p1 (see Fig. 5).

Table 1 Status of the external pins during Idle and Power Down modes

mode	memory	ALE	$\overline{PSEN}$	Port 0	Port 1	Port 2	Port 3
Idle	internal	1	1	port data	port data	port data	port data
Idle	external	1	1	floating	port data	address	port data
Power-down	internal	0	0	port data	port data	port data	port data
Power-down	external	0	0	floating	port data	port data	port data

Interrupt system (see Fig. 7)

External events and the real-time-driven on-chip peripherals require service by the CPU asynchronous to the execution of any particular section of code. To tie the asynchronous activities of these functions to normal program execution a multiple-source, two-priority-level, nested interrupt system is provided. Interrupt response latency is from 3 μs to 7 μs when using a 12 MHz crystal.

The PCB85C51 acknowledge interrupt requests from five sources as follows:

- INT0 and INT1; externally via pins 12 and 13 respectively.
- Timer 0 and Timer 1; from the two internal counters.
- Serial Port; from the internal serial I/O port.

Each interrupt vectors to a separate location in program memory for its service program.

Each source can be individually enabled or disabled and can be programmed to a high or low priority level. Also all enabled sources can be globally disabled or enabled. Both external interrupts can be programmed to be level-activated or transition-activated and are active LOW to allow "wire-ORing" of several interrupt sources to the input pin.

DEVELOPMENT DATA

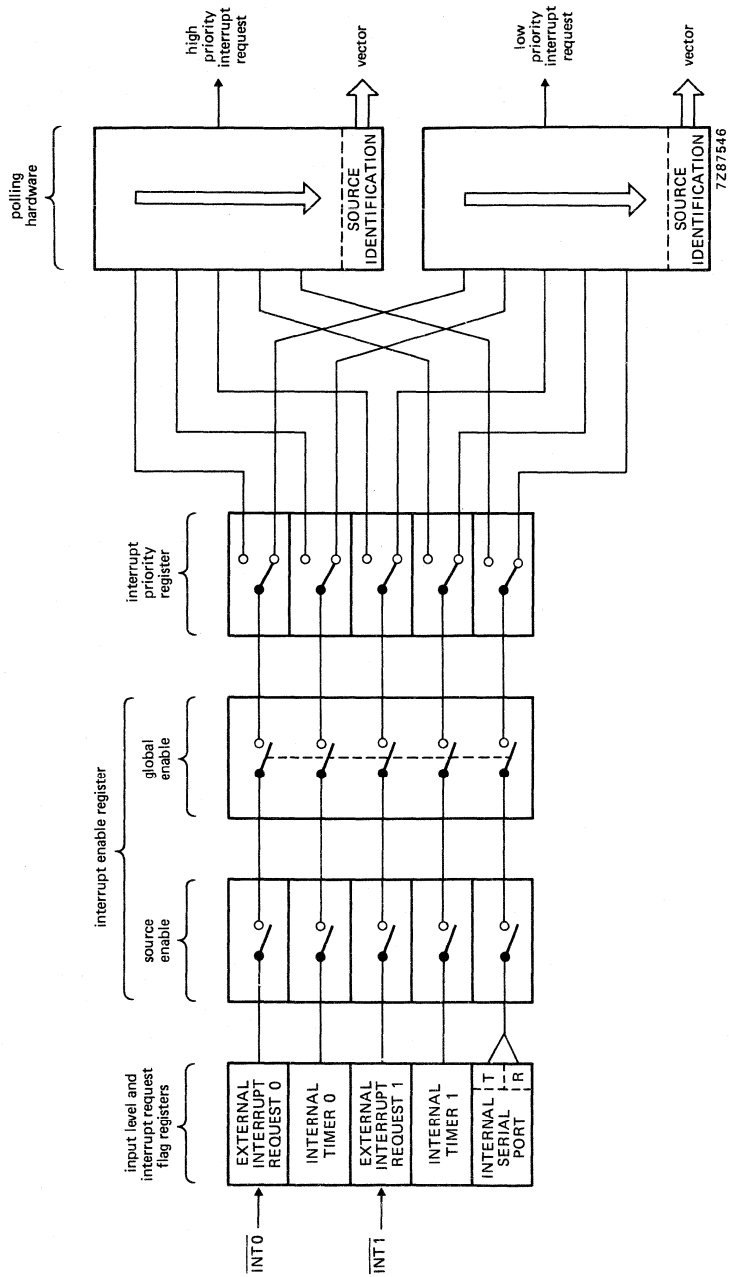


Fig. 7 Interrupt system.

FUNCTIONAL DESCRIPTION (continued)

Oscillator circuitry

The oscillator circuitry of the PCB85C51 is a single-stage inverting amplifier in a Pierce oscillator configuration. The circuitry between XTAL 1 and XTAL 2 is basically an inverter biased to the transfer point. Either a crystal or ceramic resonator can be used as the feedback element to complete the oscillator circuitry. Both are operated in parallel resonance. XTAL 1 (pin 19) is the high gain amplifier input, and XTAL 2 (pin 18) is the output (see Fig. 8). To drive the PCB85C51 externally, XTAL 1 is driven from an external source and XTAL 2 left open-circuit (see Fig. 9).

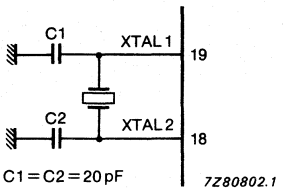


Fig. 8 PCB85C51 oscillator circuit.

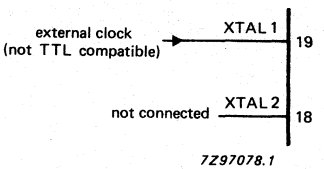


Fig. 9 PCB85C51 driven from an external source.

Reset circuitry

The reset circuitry for the PCB85C51 is connected to the reset pin, RST, as shown in Fig. 10. A Schmitt trigger is used at the input for noise rejection. The output of the Schmitt trigger is sampled by the reset circuitry every machine cycle.

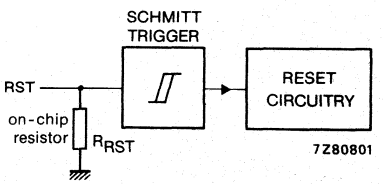


Fig. 10 Reset configuration at RST.

A reset is accomplished by holding the RST pin HIGH for at least two machine cycles (24 oscillator periods), while the oscillator is running. The CPU responds by executing an internal reset. It also configures the ALE and $\overline{\text{PSEN}}$ pins as inputs. (They are quasi-bidirectional.)

The internal reset is executed during the second cycle in which RST is HIGH and is repeated every cycle until RST goes LOW. It leaves the internal registers as follows:

Register	Content
PC	000H
ACC	00H
B	00H
PSW	00H
SP	07H
DPTR	0000H
P0 – P3	0FFH
IP	(XX000000)
IE	(0X000000)
TMOD	00H
TCON	00H
TH0	00H
TL0	00H
TH1	00H
TL1	00H
SCON	00H
SBUF	Intermediate
PCON	(0XXX0000)

The internal RAM is not affected by reset. When V_{CC} is turned on, the RAM content is indeterminate.

Power-on reset (see Fig. 11)

When V_{CC} is turned on, and provided its rise-time does not exceed 10 ms, an automatic reset can be obtained by connecting the RST pin to V_{CC} via a 10 μF capacitor. When the power is switched on, the voltage on the RST pin is the difference between V_{CC} and the capacitor voltage, this decreases from V_{CC} as the capacitor charges through the internal resistor (R_{RST}) to ground. The larger the capacitor, the more slowly V_{RST} decreases. V_{RST} must remain above the lower threshold of the Schmitt trigger long enough to effect a complete reset. The time required is the oscillator start-up time, plus 2 machine cycles. To avoid a 'latch-up' of the CMOS circuitry, the power supply on the RST pin at switch-on must not exceed V_{CC} + 0,5 V or V_{SS} −0,5 V. ←

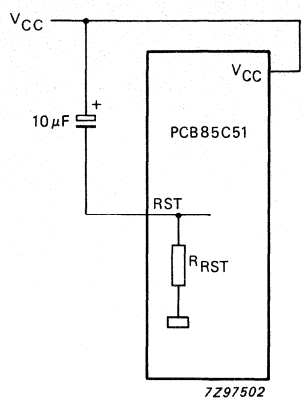


Fig. 11 Power-on reset.

INSTRUCTION SET

Table 2 Instruction set description

mnemonic		description	bytes/ cycles		opcode (hex.)
Arithmetic operations					
ADD	A,Rr	Add register to A	1	1	2*
ADD	A,direct	Add direct byte to A	2	1	25
ADD	A,@Ri	Add indirect RAM to A	1	1	26, 27
ADD	A,#data	Add immediate data to A	2	1	24
ADDC	A,Rr	Add register to A with carry flag	1	1	3*
ADDC	A,direct	Add direct byte to A with carry flag	2	1	35
ADDC	A,@Ri	Add indirect RAM to A with carry flag	1	1	36, 37
ADDC	A,#data	Add immediate data to A with carry flag	2	1	34
SUBB	A,Rr	Subtract register from A with borrow	1	1	9*
SUBB	A,direct	Subtract direct byte from A with borrow	2	1	95
SUBB	A,@Ri	Subtract indirect RAM from A with borrow	1	1	96, 97
SUBB	A,#data	Subtract immediate data from A with borrow	2	1	94
INC	A	Increment A	1	1	04
INC	Rr	Increment register	1	1	0*
INC	direct	Increment direct byte	2	1	05
INC	@Ri	Increment indirect RAM	1	1	06, 07
DEC	A	Decrement A	1	1	14
DEC	Rr	Decrement register	1	1	1*
DEC	direct	Decrement direct byte	2	1	15
DEC	@Ri	Decrement indirect RAM	1	1	16, 17
INC	DPTR	Increment data pointer	1	2	A3
MUL	AB	Multiply A & B	1	4	A4
DIV	AB	Divide A by B	1	4	84
DA	A	Decimal adjust A	1	1	D4

mnemonic		description	bytes/ cycles		opcode (hex.)
Logic operations					
ANL	A,Rr	AND register to A	1	1	5*
ANL	A,direct	AND direct byte to A	2	1	55
ANL	A,@Ri	AND indirect RAM to A	1	1	56, 57
ANL	A,#data	AND immediate data to A	2	1	54
ANL	direct,A	AND A to direct byte	2	1	52
ANL	direct,#data	AND immediate data to direct byte	3	2	53
ORL	A,Rr	OR register to A	1	1	4*
ORL	A,direct	OR direct byte to A	2	1	45
ORL	A,@Ri	OR indirect RAM to A	1	1	46, 47
ORL	A,#data	OR immediate data to A	2	1	44
ORL	direct,A	OR A to direct byte	2	1	42
ORL	direct,#data	OR immediate data to direct byte	3	2	43
XRL	A,Rr	Exclusive-OR register to A	1	1	6*
XRL	A,direct	Exclusive-OR direct byte to A	2	1	65
XRL	A,@Ri	Exclusive-OR indirect RAM to A	1	1	66, 67
XRL	A,#data	Exclusive-OR immediate data to A	2	1	64
XRL	direct,A	Exclusive-OR A to direct byte	2	1	62
XRL	direct,#data	Exclusive-OR immediate data to direct byte	3	2	63
CLR	A	Clear A	1	1	E4
CPL	A	Complement A	1	1	F4
RL	A	Rotate A left	1	1	23
RLC	A	Rotate A left through the carry flag	1	1	33
RR	A	Rotate A right	1	1	03
RRC	A	Rotate A right through the carry flag	1	1	13
SWAP	A	Swap nibbles within A	1	1	C4

INSTRUCTION SET (continued)

mnemonic		description	bytes/ cycles		opcode (hex.)
Data transfer					
MOV	A,Rr	Move register to A	1	1	E*
MOV	A,direct	Move direct byte to A	2	1	E5
MOV	A,@Ri	Move indirect RAM to A	1	1	E6, E7
MOV	A,#data	Move immediate data to A	2	1	74
MOV	Rr,A	Move A to register	1	1	F*
MOV	Rr,direct	Move direct byte to register	2	2	A*
MOV	Rr,#data	Move immediate data to register	2	1	7*
MOV	direct,A	Move A to direct byte	2	1	F5
MOV	direct,Rr	Move register to direct byte	2	2	8*
MOV	direct,direct	Move direct byte to direct	3	2	85
MOV	direct,@Ri	Move indirect RAM to direct byte	2	2	86, 87
MOV	direct,#data	Move immediate data to direct byte	3	2	75
MOV	@Ri,A	Move A to indirect RAM	1	1	F6, F7
MOV	@Ri,direct	Move direct byte to indirect RAM	2	2	A6, A7
MOV	@Ri,#data	Move immediate data to indirect RAM	2	1	76, 77
MOV	DPTR,#data16	Load data pointer with a 16-bit constant	3	2	90
MOVC	A,@A+DPTR	Move code byte relative to DPTR to A	1	2	93
MOVC	A,@A+PC	Move code byte relative to PC to A	1	2	83
MOVX	A,@Ri	Move external RAM (8-bit address) to A	1	2	E2, E3
MOVX	A,@DPTR	Move external RAM (16-bit address) to A	1	2	E0
MOVX	@Ri,A	Move A to external RAM (8-bit address)	1	2	F2, F3
MOVX	@DPTR,A	Move A to external RAM (16-bit address)	1	2	F0
PUSH	direct	Push direct byte onto stack	2	2	C0
POP	direct	Pop direct byte from stack	2	2	D0
XCH	A,Rr	Exchange register with A	1	1	C*
XCH	A,direct	Exchange direct byte with A	2	1	C5
XCH	A,@Ri	Exchange indirect RAM with A	1	1	C6, C7
XCHD	A,@Ri	Exchange LOW-order digit indirect RAM with A	1	1	D6, D7

mnemonic		description	bytes/ cycles		opcode (hex.)
Boolean variable manipulation					
CLR	C	Clear carry flag	1	1	C3
CLR	bit	Clear direct bit	2	1	C2
SETB	C	Set carry flag	1	1	D3
SETB	bit	Set direct bit	2	1	D2
CPL	C	Complement carry flag	1	1	B3
CPL	bit	Complement direct bit	2	1	B2
ANL	C,bit	AND direct bit to carry flag	2	2	82
ANL	C,/bit	AND complement of direct bit to carry flag	2	2	B0
ORL	C,bit	OR direct bit to carry flag	2	2	72
ORL	C,/bit	OR complement of direct bit to carry flag	2	2	A0
MOV	C,bit	Move direct bit to carry flag	2	1	A2
MOV	bit,C	Move carry flag to direct bit	2	2	92
Program and machine control					
ACALL	addr11	Absolute subroutine call	2	2	●1addr
LCALL	addr16	Long subroutine call	3	2	12
RET		Return from subroutine	1	2	22
RETI		Return from interrupt	1	2	32
AJMP	addr11	Absolute jump	2	2	▲1addr
LJMP	addr16	Long jump	3	2	02
SJMP	rel	Short jump (relative address)	2	2	80
JMP	@A+DPTR	Jump indirect relative to the DPTR	1	2	73
JZ	rel	Jump if A is zero	2	2	60
JNZ	rel	Jump if A is not zero	2	2	70
JC	rel	Jump if carry flag is set	2	2	40
JNC	rel	Jump if no carry flag	2	2	50
JB	bit,rel	Jump if direct bit is set	3	2	20
JNB	bit,rel	Jump if direct bit is not set	3	2	30
JBC	bit,rel	Jump if direct bit is set and clear bit	3	2	10
CJNE	A,direct,rel	Compare direct to A and jump if not equal	3	2	B5
CJNE	A,#data,rel	Compare immediate to A and jump if not equal	3	2	B4
CJNE	Rr,#data,rel	Compare immed. to reg. and jump if not equal	3	2	B*
CJNE	@Ri,#data,rel	Compare immed. to ind. and jump if not equal	3	2	B6, B7
DJNZ	Rr,rel	Decrement register and jump if not zero	2	2	D*
DJNZ	direct,rel	Decrement direct and jump if not zero	3	2	D5
NOP		No operation	1	1	00

DEVELOPMENT DATA

Notes to Table 2

Data addressing modes

Rr	Working register R0-R7.
direct	128 internal RAM locations and any special function register (SFR).
@Ri	Indirect internal RAM location addressed by register R0 or R1 of the actual register bank.
#data	8-bit constant included in instruction.
#data16	16-bit constant included as bytes 2 and 3 of instruction.
bit	direct addressed bit in internal RAM or SFR.
addr16	16-bit destination address. Used by LCALL and LJMP. The branch will be anywhere within the 64 K-byte program memory address space.
addr11	11-bit destination address. Used by ACALL and AJMP. The branch will be within the same 2 K-byte page of program memory as the first byte of the following instruction.
rel	Signed (two's complement) 8-bit offset byte. Used by SJMP and all conditional jumps. Range is −128 to + 127 bytes relative to first byte of the following instruction.

Hexadecimal opcode cross-reference to Table 2

*: 8, 9, A, B, C, D, E, F.

●: 11, 31, 51, 71, 91, B1, D1, F1.

▲: 01, 21, 41, 61, 81, A1, C1, E1.

RATINGS

Limiting values in accordance with the Absolute Maximum System (IEC 134)

Input voltage on any pin with respect to ground (V_{SS})	V_I	−0,5 to + 7	V
Input, output current	$\pm I_I, I_O$	max. 10	mA
Total power dissipation	P_{tot}	max. 1	W
Storage temperature range	T_{stg}	−65 to + 150	°C
Operating ambient temperature range	T_{amb}	0 to + 70	°C

D.C. CHARACTERISTICS

V_{CC} = 5 V (± 10%); V_{SS} = 0 V; T_{amb} = 0 to + 70 °C; all voltages with respect to V_{SS} unless otherwise specified.

parameter	symbol	min.	max.	unit	conditions
Supply voltage	V_{CC}	4,5	5,5	V	
Supply current					
operating (note 1)	I_{CC}	–	30*	mA	f_{CLK} = 12 MHz
Idle mode (note 2)	I_{CC}	–	10*	mA	f_{CLK} = 12 MHz
Power Down current	I_{PD}	–	100*	μA	V_{CC} = 2 V (note 3)
Inputs					
LOW level input voltage (except $\overline{EA}$)	V_{IL}	−0,5	$0{,}2V_{CC}-0{,}1$	V	
LOW level input voltage ($\overline{EA}$)	V_{IL1}	−0,5	$0{,}2V_{CC}-0{,}3$	V	
HIGH level input voltage (except XTAL 1, RST)	V_{IH}	$0{,}2V_{CC}+0{,}9$	$V_{CC}+0{,}5$	V	
HIGH level input voltage (XTAL 1, RST)	V_{IH1}	$0{,}7V_{CC}$	$V_{CC}+0{,}5$	V	
Input current logic 0 (Ports 1, 2 and 3)	$-I_{IL}$	–	50	μA	V_I = 0,45 V
Input current logic 1 to 0 transition (Ports 1, 2 and 3)	$-I_{TL}$	–	500	μA	V_I = 2 V
Input leakage current (Port 0, $\overline{EA}$)	$\pm I_{LI}$	–	10	μA	$0{,}45\ V < V_I < V_{CC}$

DEVELOPMENT DATA

* Preliminary value for 85C51 only. Current for piggy-back EPROM must be added.

D.C. CHARACTERISTICS (continued)

parameter	symbol	min.	max.	unit	conditions
Outputs					
LOW level output voltage (note 4) (Ports 1, 2 and 3)	V_{OL}	–	0,45	V	I_{OL} = 1,6 mA
LOW level output voltage (note 4) (Port 0, ALE, $\overline{PSEN}$)	V_{OL1}	–	0,45	V	I_{OL} = 3,2 mA
HIGH level output voltage (Ports 1, 2 and 3)	V_{OH}	2,4	–	V	$-I_{OH}$ = 80 μA; V_{CC} = 5 V ± 10%
		$0{,}75V_{CC}$	–	V	$-I_{OH}$ = 30 μA
		$0{,}9V_{CC}$	–	V	$-I_{OH}$ = 10 μA
HIGH level output voltage (note 5) (Port 0 in external Bus mode, ALE, $\overline{PSEN}$)	V_{OH1}	2,4	–	V	$-I_{OH}$ = 400 μA; V_{CC} = 5 V ± 10%
		$0{,}75V_{CC}$	–	V	$-I_{OH}$ = 150 μA
		$0{,}9V_{CC}$	–	V	$-I_{OH}$ = 40 μA
RST pull-down resistor	R_{RST}	40	125	kΩ	
I/O pin capacitance	$C_{I/O}$	–	10	pF	test freq. = 1 MHz; T_{amb} = 25 °C

Notes to the d.c. characteristics

1. The operating supply current is measured with all output pins disconnected; XTAL 1 driven with $t_r = t_f$ = 10 ns, $V_{IL} = V_{SS}$ + 0,5 V, $V_{IH} = V_{CC}$ –0,5 V; XTAL 2 not connected; $\overline{EA}$ = RST = Port 0 = V_{CC}.
2. The Idle mode supply current is measured with all output pins disconnected; XTAL 1 driven $t_r = t_f$ = 10 ns, $V_{IL} = V_{SS}$ + 0,5 V, $V_{IH} = V_{CC}$ –0,5 V; XTAL 2 not connected; $\overline{EA}$ = Port 0 = V_{CC}; RST = V_{SS}.
3. The Power Down current is measured with all output pins disconnected; XTAL 2 not connected; $\overline{EA}$ = Port 0 = V_{CC}; RST = V_{SS}.
4. Capacitive loading on Port 0 and Port 2 may cause spurious noise pulses to be superimposed on the LOW level output voltage of ALE, Port 1 and Port 3. The noise is due to external Bus capacitance discharging into the Port 0 and Port 2 pins when these pins make a 1-to-0 transition during Bus operations. In the most adverse condition (capacitive loading > 100 pF) the noise pulse on ALE line may exceed 0,8 V. In this event it may be required to qualify ALE with a Schmitt trigger, or use an address latch with a Schmitt trigger STROBE input.
5. Capacitive loading on Port 0 and Port 2 may cause the HIGH level output voltage on ALE and $\overline{PSEN}$ to momentarily fall below the 0,9 V_{CC} specification when the address bits are stabilizing.

A.C. CHARACTERISTICS (Non-standard pins. See Fig. 18)

Conditions: 12 MHz clock, C_L = 80 pF on A0-A13
C_L = 100 pF on $\overline{PSEN}$, ALE
V_{CC} = + 5 V ± 10%, T_{amb} = 0 to + 70 °C

parameter	symbol	min.	max.	unit
ALE pulse width	TLHLL	140	–	ns
Address set up to ALE	TAVLL	45	–	ns
Address hold up after ALE	TLLAX	50	–	ns
$\overline{PSENE}$ width	TPLPH	230	–	ns
$\overline{PSENE}$, ALE cycle time	TLHLH	500	–	ns
$\overline{PSENE}$ to valid emulation instruction in	TPLDV	150	–	ns
Input emulation instruction hold after $\overline{PSENE}$	TPHDX	0	–	ns
Address to valid emulation instruction in	TAVDV	–	320	ns
ALE to $\overline{PSENE}$	TLLPL	58	–	ns

DEVELOPMENT DATA

A.C. CHARACTERISTICS (Standard pins)

V_{CC} = 5 V ± 10%; V_{SS} = 0 V; T_{amb} = 0 to + 70 °C; C_L = 100 pF (Port 0, ALE and $\overline{PSENE}$); C_L = 80 pF (all other outputs); unless otherwise specified (see waveforms Figs 14, 15 and 16)

parameter	symbol	10 MHz		12 MHz		variable clock		unit
		min.	max.	min.	max.	min.	max.	
Program memory								
ALE pulse duration	t_{LL}	160	–	127	–	$2t_{CK}-40$	–	ns
Address set-up time to ALE	t_{AL}	60	–	43	–	$t_{CK}-40$	–	ns
Address hold time after ALE	t_{LA}	65	–	48	–	$t_{CK}-35$	–	ns
Time from ALE to valid instruction input	t_{LIV}	–	300	–	233	–	$4t_{CK}-100$	ns
Time from ALE to control pulse $\overline{PSEN}$	t_{LC}	75	–	58	–	$t_{CK}-25$	–	ns
Control pulse duration $\overline{PSEN}$	t_{CC}	265	–	215	–	$3t_{CK}-35$	–	ns
Time from $\overline{PSEN}$ to valid instruction input	t_{CIV}	–	175	–	125	–	$3t_{CK}-125$	ns
Input instruction hold time after $\overline{PSEN}$	t_{CI}	0	–	0	–	0	–	ns
Input instruction float delay after $\overline{PSEN}$*	t_{CIF}	–	80	–	63	–	$t_{CK}-20$	ns
Address valid after $\overline{PSEN}$*	t_{AC}	92	–	75	–	$t_{CK}-8$	–	ns
Address to valid instruction input	t_{AIV}	–	385	–	302	–	$5t_{CK}-115$	ns
Address float time to $\overline{PSEN}$	t_{AFC}	−12	–	−12	–	−12	–	ns

* Interfacing the PCB85C51 to devices with float times up to 75 ns is permitted. This limited bus contention will not cause damage to Port 0 drivers.

DEVELOPMENT DATA

parameter	symbol	10 MHz		12 MHz		variable clock		unit
		min.	max.	min.	max.	min.	max.	
External data memory								
$\overline{RD}$ pulse duration	t_{RR}	500	–	400	–	$6t_{CK}-100$	–	ns
$\overline{WR}$ pulse duration	t_{WW}	500	–	400	–	$6t_{CK}-100$	–	ns
Address hold time after ALE	t_{LA}	65	–	48	–	$t_{CK}-35$	–	ns
$\overline{RD}$ to valid data input	t_{RD}	–	335	–	250	–	$5t_{CK}-165$	ns
Data hold time after $\overline{RD}$	t_{DR}	0	–	0	–	0	–	ns
Data float delay after $\overline{RD}$	t_{DFR}	–	130	–	97	–	$2t_{CK}-70$	ns
Time from ALE to valid data input	t_{LD}	–	650	–	517	–	$8t_{CK}-150$	ns
Address to valid data input	t_{AD}	–	735	–	585	–	$9t_{CK}-165$	ns
Time from ALE to $\overline{RD}$ or $\overline{WR}$	t_{LW}	250	350	200	300	$3t_{CK}-50$	$3t_{CK}+50$	ns
Time from address to $\overline{RD}$ or $\overline{WR}$	t_{AW}	270	–	203	–	$4t_{CK}-130$	–	ns
Time from $\overline{RD}$ or $\overline{WR}$ HIGH to ALE HIGH	t_{WHLH}	60	140	43	123	$t_{CK}-40$	$t_{CK}+40$	ns
Data valid to $\overline{WR}$ transition	t_{DWX}	40	–	23	–	$t_{CK}-60$	–	ns
Data set-up time before $\overline{WR}$	t_{DW}	550	–	433	–	$7t_{CK}-150$	–	ns
Data hold time after $\overline{WR}$	t_{WD}	50	–	33	–	$t_{CK}-50$	–	ns
Address float delay after $\overline{RD}$	t_{AFR}	–	12	–	12	–	12	ns

Where:

$1/t_{CK}$ = 3,5 to 12 MHz and are preliminary values (see Fig. 13 and Table 3).

A.C. CHARACTERISTICS (continued)

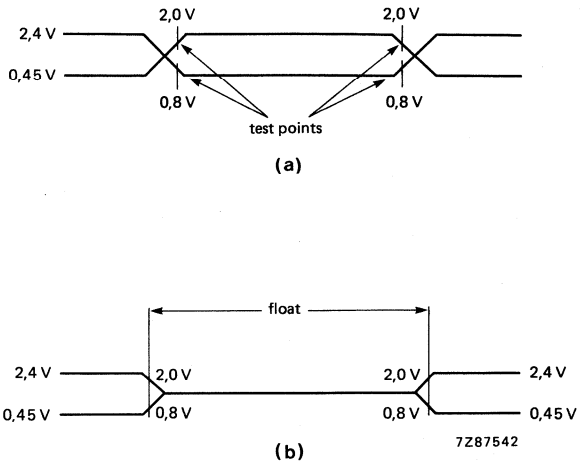


Fig. 12 A.C. testing input, output waveform (a) and float waveform (b).

A.C. testing inputs are driven at 2,4 V for a logic 1 and 0,45 V for a logic 0. Timing measurements are taken at 2,0 V for a logic 1 and 0,8 V for logic 0. The float state is defined as the point at which a Port 0 pin sinks 3,2 mA or sources 400 μA at the voltage test levels.

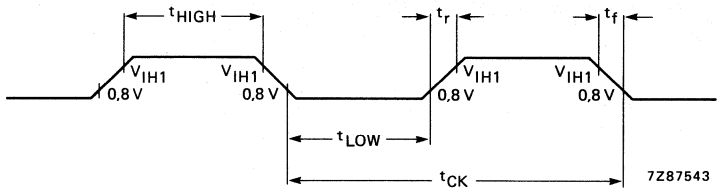


Fig. 13 External clock drive XTAL 1 (see Table 3).

Table 3 External clock drive XTAL 1 (see Fig. 13).

parameter	symbol	variable clock (f = 3,5 to 12 MHz) *		unit
		min.	max.	
Oscillator clock period	t_{CK}	83,3	286	ns
HIGH time	t_{HIGH}	20	$t_{CK}-t_{LOW}$	ns
LOW time	t_{LOW}	20	$t_{CK}-t_{HIGH}$	ns
rise time	t_r	–	20	ns
fall time	t_f	–	20	ns

* Preliminary values.

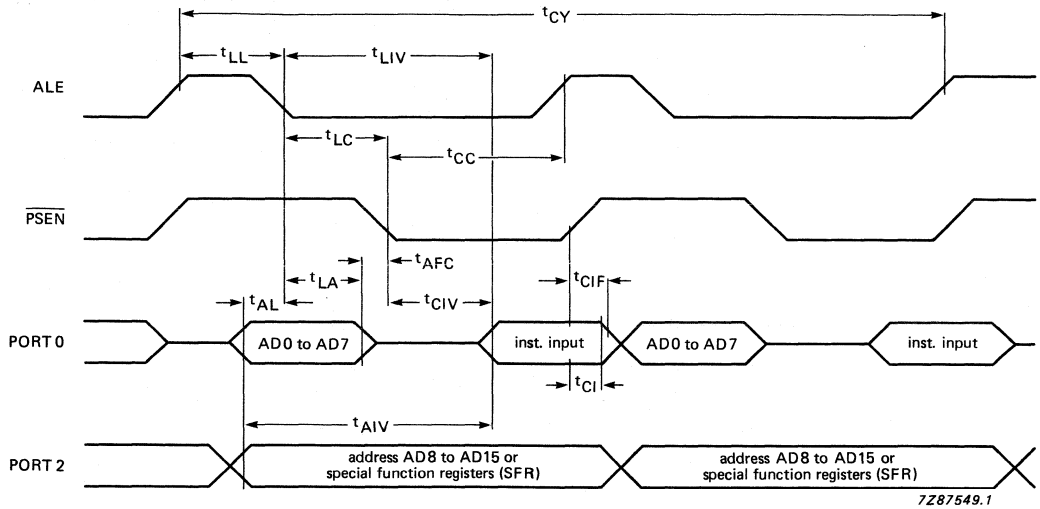


Fig. 14 Read from program memory via Port 0 and Port 2.

DEVELOPMENT DATA

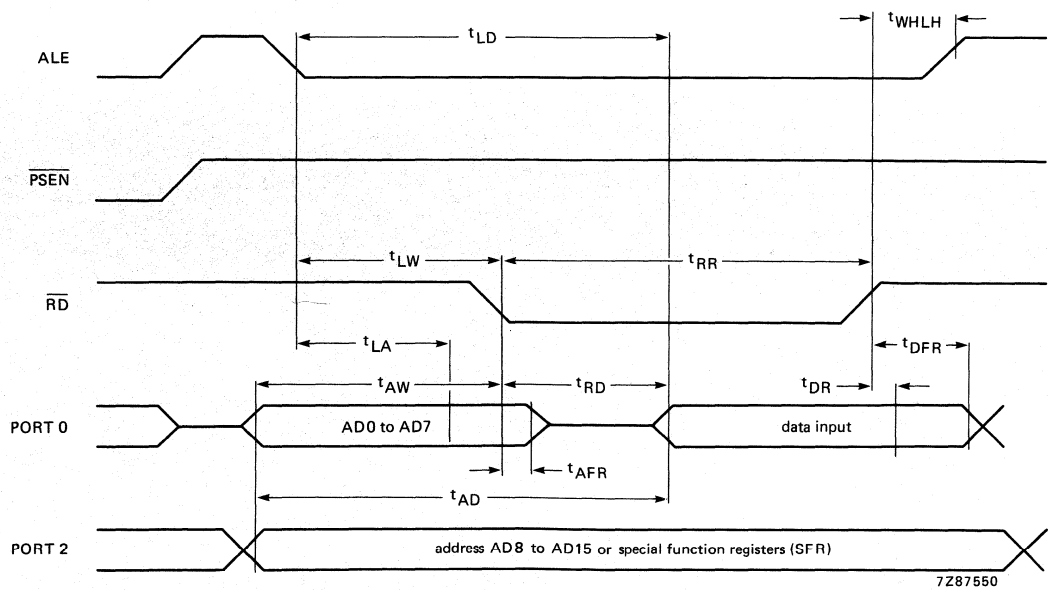


Fig. 15 Read from data memory via Port 0 and Port 2.

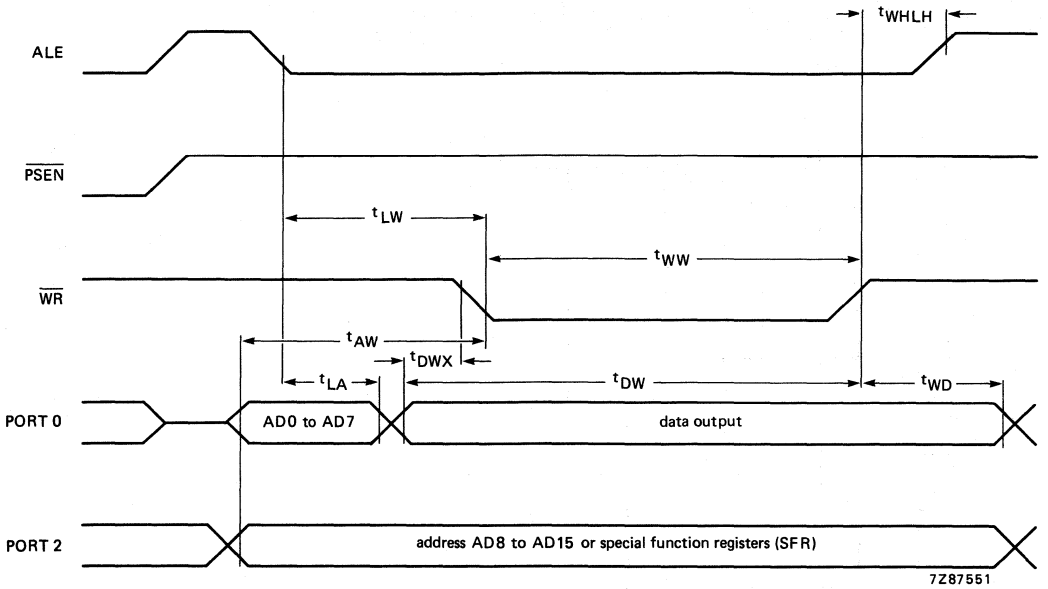


Fig. 16 Write to data memory via Port 0 and Port 2.

DEVELOPMENT DATA

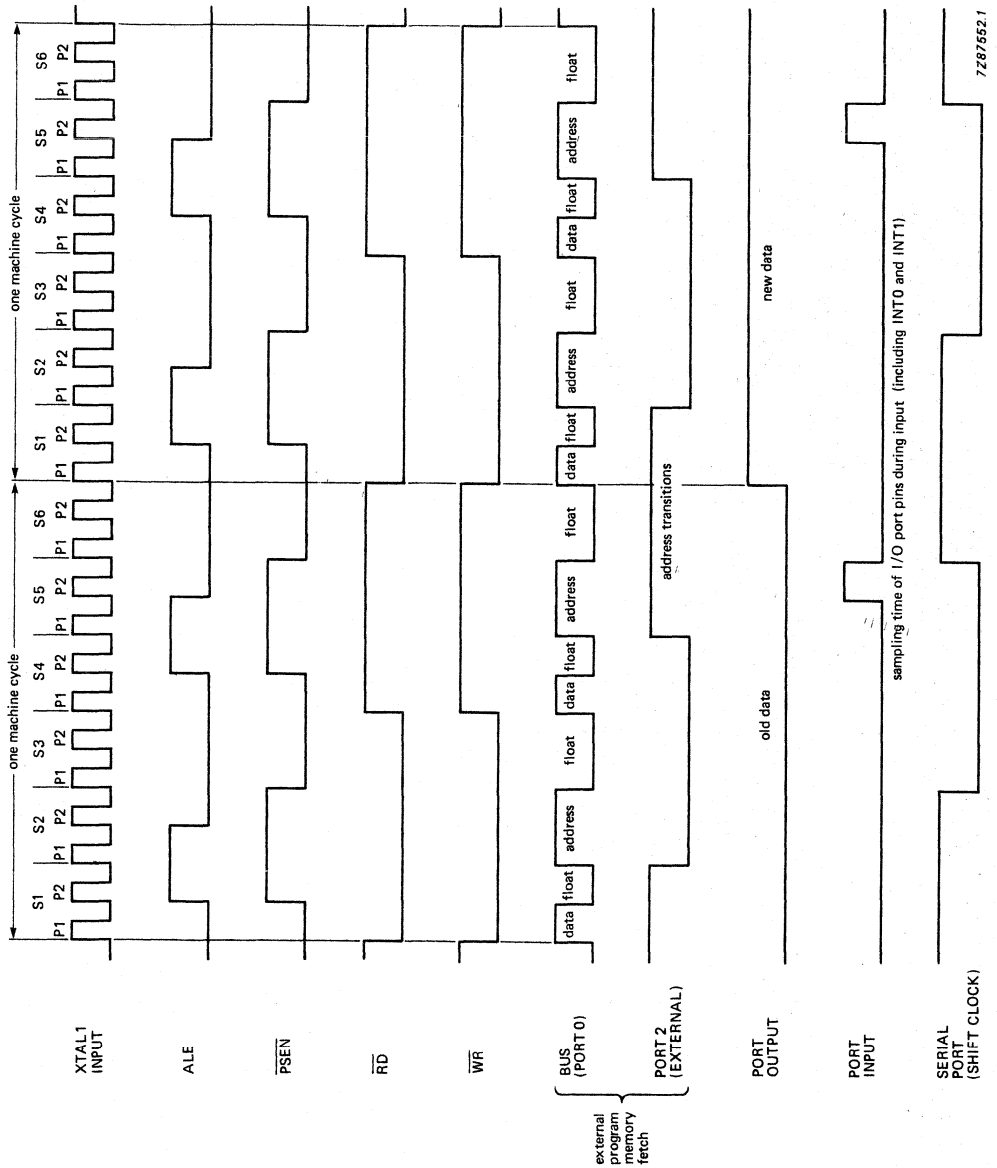


Fig. 17 Instruction cycle timing.

D.C. characteristics for non-standard signals (PSENE, A0-A13, DO-D7)

PCB85C51 (V_{CC} = 4,5 to 5,5 V)

Non-standard pins	min.	max.	unit	conditions
V_{OL} output LOW voltage		0,45	V	I_{OL} = 1,6 mA
V_{OH} output HIGH voltage	2,4		V	I_{OH} = −400 μA
V_{IL} input LOW voltage	−0,5	0,8	V	
V_{IH} input HIGH voltage	2,0	V_{CC}	V	V_{CC} < 5,5 V
→ **$\overline{\text{PSENE}}$**				
V_{OL} output LOW voltage		0,45	V	I_{OL} = 3,2 mA
V_{OH} output HIGH voltage	2,4		V	I_{OH} = −400 μA

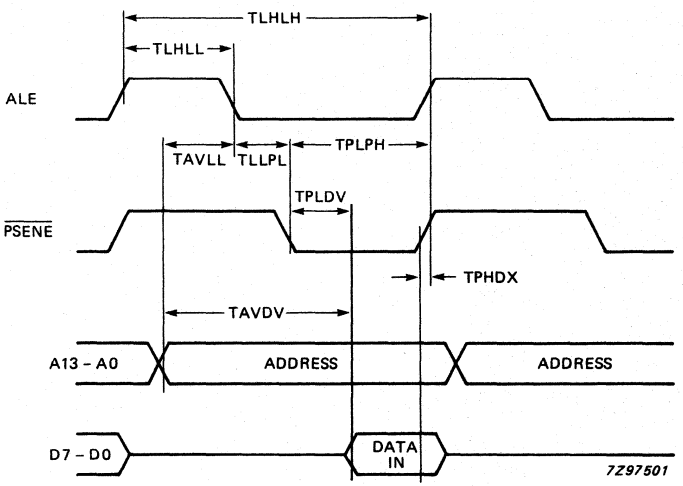


Fig. 18 Reading from quasi-internal program memory.

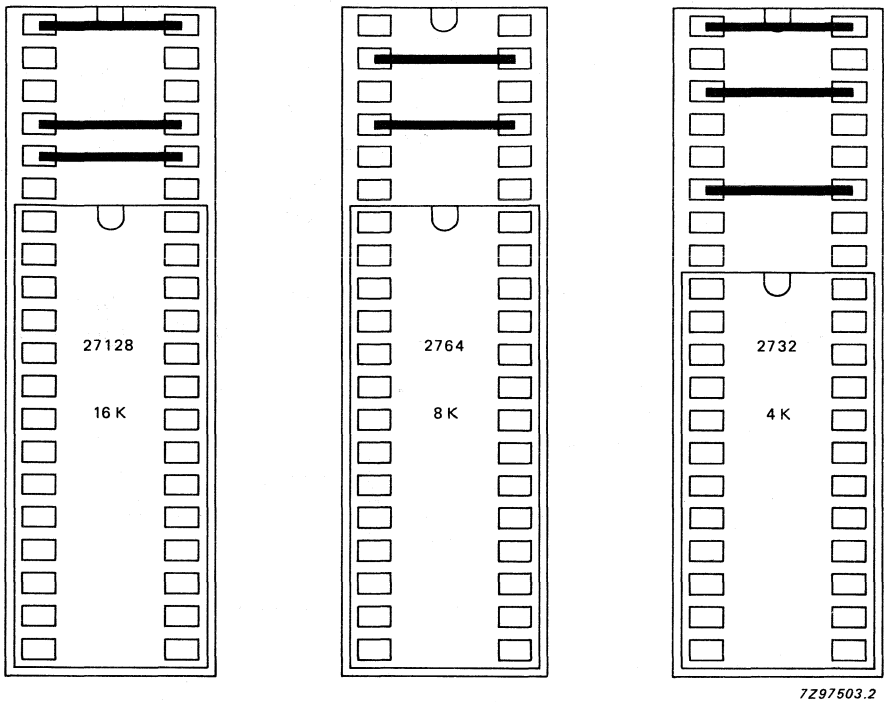


Fig. 19 Connection diagram for piggy-back package.

Jumpers with a length of 15,24 mm for connecting M1 and M2 to V_{SS} or V_{CC} may be ordered via catalogue number 4312 065 01138.

CMOS MICROCONTROLLER FOR TELEPHONE SETS

GENERAL DESCRIPTION

The PCD3315C is a single-chip 8-bit microcontroller fabricated in CMOS and is a member of the PCD3343 family. It has special on-chip features for application in telephone sets.

Features

- 8-bit CPU, ROM, RAM, I/O in a single 28-lead DIL or SO package
- 1536 ROM bytes
- 160 RAM bytes
- 20 quasi-bidirectional I/O port lines
- Two test inputs: one of which is also the external interrupt input (CE/$\overline{T0}$)
- Single-level vectored interrupts: external, timer/event counter
- 8-bit programmable timer/event counter
- Over 80 instructions (based on MAB8048, MAB8400, PCD3343 and PCF8500)
- All instructions 1 or 2 cycles
- Clock frequency 100 kHz to 10 MHz
- Single supply voltage from 1,8 V to 6 V
- Low standby voltage and current
- STOP and IDLE mode
- On-chip oscillator with output drive capability for peripherals
- Configuration of all I/O port lines individually selected by mask: pull-up, open drain or push-pull
- Power-on-reset circuit and low supply voltage detection
- Reset state of all ports individually selected by mask
- Operating temperature range: −25 to + 70 °C

PACKAGE OUTLINES

PCD3315CP: 28-lead DIL; plastic (SOT-117).
PCD3315CT: 28-lead mini-pack; plastic (SO-28; SOT-136A).

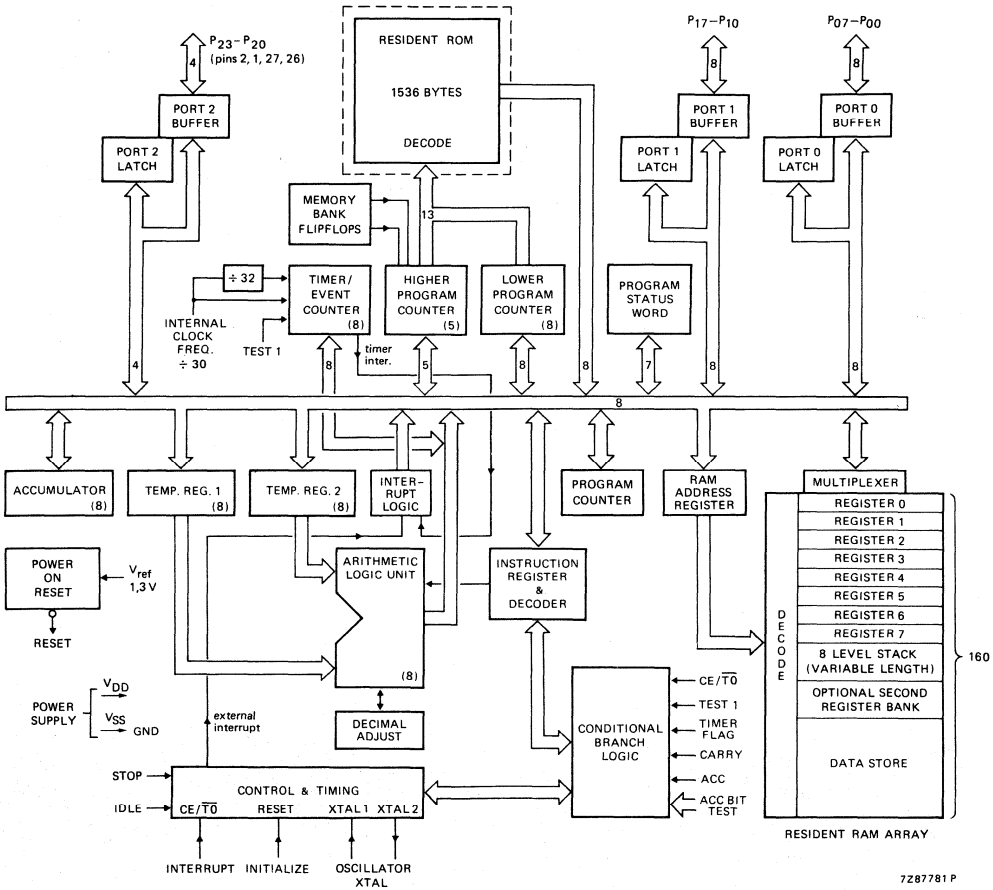
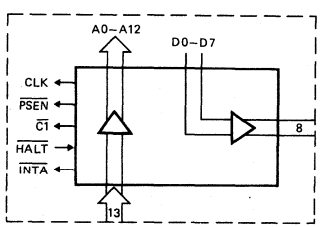
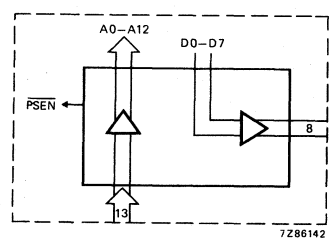


Fig. 1 Block diagram; PCD3315C.



(a)

Fig. 1a Replacement of dotted part in Fig. 1, for the PCD8500F bond-out version.



(b)

Fig. 1b Replacement of dotted part in Fig. 1, for the PCF8500B 'Piggy-back' version.

DEVELOPMENT DATA

PINNING

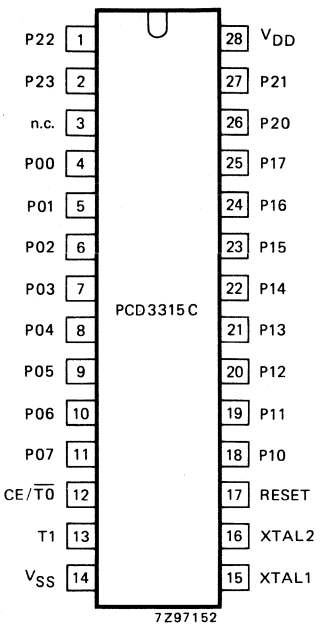


Fig. 2 Pinning diagram: PCD3315C.

PIN DESIGNATION

3	n.c.	not connected
4-11	P00-P07	**Port 0**: 8-bit quasi-bidirectional I/O port.
12	CE/$\overline{T0}$	**Interrupt/Test 0**: external interrupt input (sensitive to positive-going edge edge)/test input pin; when used as a test input directly tested by conditional branch instructions JT0 and JNT0.
13	T1	**Test 1**: test input pin, directly tested by conditional branch instructions JT1 and JNT1. T1 also functions as an input to the 8-bit timer/event counter, using the STRT CNT instruction.
14	V_{SS}	**Ground**: circuit earth potential.
15	XTAL 1	**Crystal input**: connection to timing component (crystal) which determines the frequency of the internal oscillator; also the input for an external clock source.
16	XTAL 2	connection to the other side of the timing component.
17	RESET	**Reset input**: used to initialize the processor (active HIGH), or output of power-on-reset circuit.
18-25	P10-P17	**Port 1**: 8-bit quasi-bidirectional I/O port.
26, 27, 1, 2	P20-P23	**Port 2**: 4-bit quasi-bidirectional I/O port.
28	V_{DD}	**Power supply**: 1,8 V to 6 V.

D.C. CHARACTERISTICS

V_{DD} = 2,5 to 6 V; V_{SS} = 0 V; T_{amb} = −25 to + 70 °C; all voltages with respect to V_{SS}; f = 3,58 MHz with R_S = 50 Ω; unless otherwise specified.

parameter	symbol	min.	typ.	max.	unit
Supply voltage					
operating	V_{DD}	1,8	–	6	V
STOP mode for RAM retention	V_{DD}	1,0	–	6	V
Supply current					
operating					
at V_{DD} = 3 V	I_{DD}	–	350	–	μA
IDLE mode					
at V_{DD} = 3 V	I_{DD}	–	150	–	μA
STOP mode (note 1)					
at V_{DD} = 1,8 V; T_{amb} = 25 °C	I_{DD}	–	1,2	2,5	μA
at V_{DD} = 1,8 V; T_{amb} = 55 °C	I_{DD}	–	–	5	μA
at V_{DD} = 1,8 V; T_{amb} = 70 °C	I_{DD}	–	–	10	μA
RESET I/O					
Switching level	V_{RESET}	–	1,2	–	V
Sink current					
at $V_{DD} > V_{RESET}$	I_{OL}	–	7	–	μA
Inputs					
Input voltage LOW	V_{IL}	0	–	$0{,}3V_{DD}$	V
Input voltage HIGH	V_{IH}	$0{,}7V_{DD}$	–	V_{DD}	V
Input leakage current					
at $V_{SS} < V_I < V_{DD}$	$\pm I_{IL}$	–	–	1	μA
Outputs					
Output voltage LOW					
at $V_I = V_{SS}$ or V_{DD}; $\|I_O\| < 1$ μA	V_{OL}	–	–	0,05	V
Output sink current LOW					
at V_{DD} = 3 V; V_O = 0,4 V	I_{OL}	0,6	1,5	–	mA
Pull-up output source current HIGH					
at V_{DD} = 3 V; $V_O = 0{,}9V_{DD}$	$-I_{OH}$	10	–	–	μA
at V_{DD} = 3 V; $V_O = V_{SS}$	$-I_{OH}$	–	–	200	μA
Push-pull output source current HIGH					
at V_{DD} = 3 V; $V_O = V_{DD}$−0,4 V	$-I_{OH}$	0,6	1,5	–	mA

Note 1

Crystal connected between XTAL 1 and XTAL 2; pin 2 pulled to V_{DD} via 5,6 kΩ resistor; CE and T1 at V_{SS}.

CMOS MICROCONTROLLER FOR TELEPHONE SETS

GENERAL DESCRIPTION

The PCD3343 is a single-chip 8-bit microcontroller fabricated in CMOS. It has special on-chip features for application in telephone sets.
The device is mask programmable, designed to provide telephone dialling facilities such as redial, repertory dial, emergency call, keyboard scan and control for liquid crystal display, pulse dial and/or DTMF dial via dedicated peripheral.

Features

- 8-bit CPU, ROM, RAM, I/O in a single 28-lead DIL or SO package
- 3 K ROM bytes
- 224 RAM bytes
- 20 quasi-bidirectional I/O port lines
- Two test inputs: one of which is also the external interrupt input (CE/$\overline{T0}$)
- Single-level vectored interrupts: external, timer/event counter, serial I/O
- Serial I/O which can be used in bus systems with more than one master (serial I/O data via an existing port line and clock via a dedicated line)
- 8-bit programmable timer/event counter
- Over 80 instructions (based on MAB8048, MAB8400 and PCF8500)
- All instructions 1 or 2 cycles
- Clock frequency 100 kHz to 10 MHz
- Single supply voltage from 1,8 V to 6 V
- Low standby voltage and current
- STOP and IDLE mode
- On-chip oscillator with output drive capability for peripherals (e.g. PCD3312 DTMF generator)
- Configuration of all I/O port lines individually selected by mask: pull-up, open drain or push-pull
- Power-on-reset circuit and low supply voltage detection
- Reset state of all ports individually selected by mask
- Operating temperature range: −25 to + 70 °C

PACKAGE OUTLINES

PCD3343P : 28-lead DIL; plastic (SOT-117).
PCD3343D: 28-lead DIL; ceramic (CERDIP) (SOT-135A).
PCD3343T : 28-lead mini-pack; plastic (SO-28; SOT-136A).

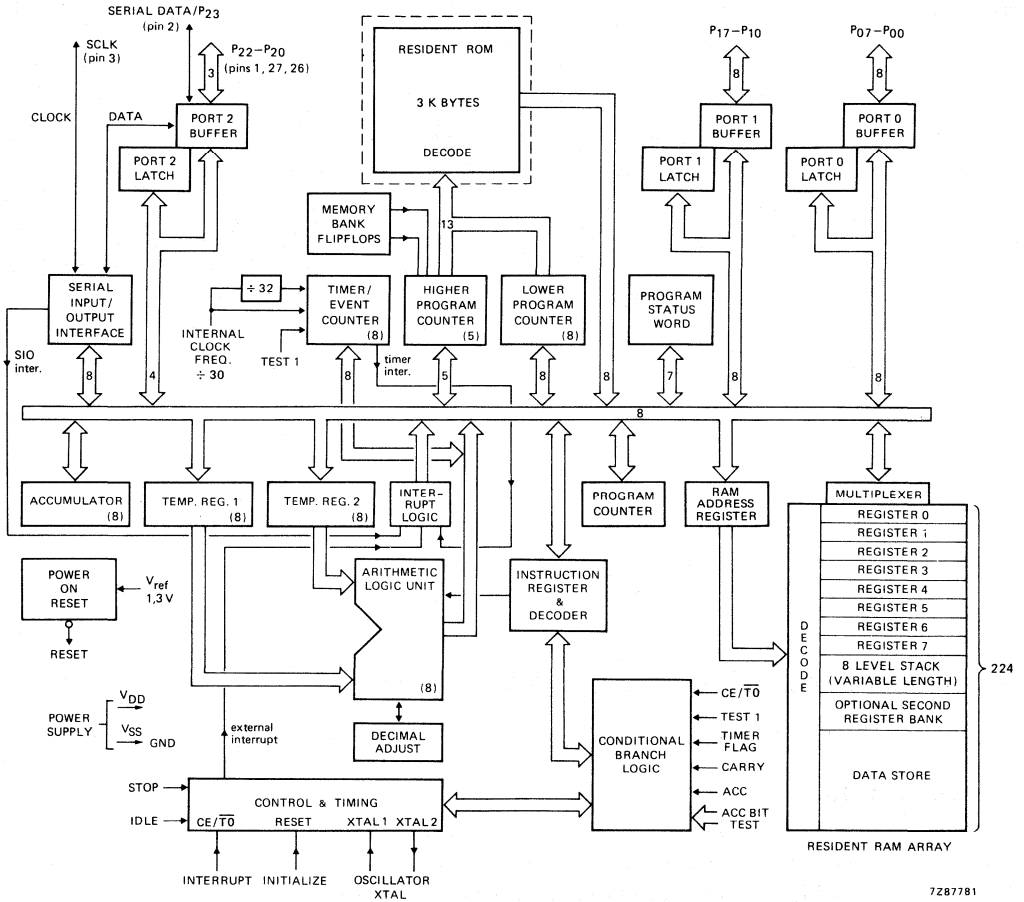
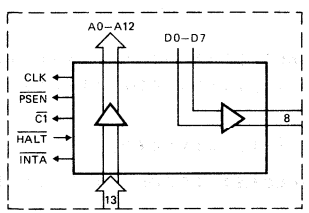
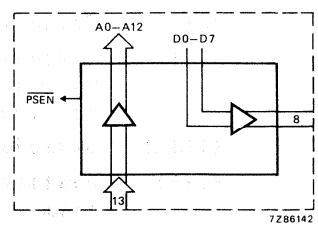


Fig. 1 Block diagram; PCD3343.



(a)

Fig. 1a Replacement of dotted part in Fig. 1, for the PCF8500F bond-out version.

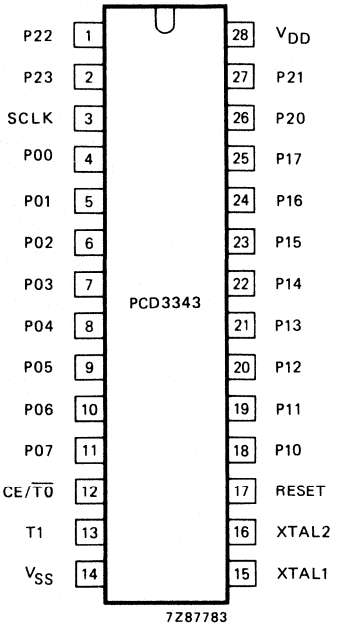


(b)

Fig. 1b Replacement of dotted part in Fig. 1, for the PCF8500B 'Piggy-back' version.

DEVELOPMENT DATA

PINNING



Note CE/$\overline{T0}$ is labelled $\overline{INT}$/T0 on the PCF8500B and has inverted polarity.

Fig. 2 Pinning diagram: PCD3343 and bottom pinning PCF8500B.

PIN DESIGNATION

3	SCLK	**Clock**: bidirectional clock for serial I/O.
4-11	P00-P07	**Port 0**: 8-bit quasi-bidirectional I/O port.
12	CE/$\overline{T0}$	**Interrupt/Test 0**: external interrupt input (sensitive to positive-going edge)/test input pin; when used as a test input directly tested by conditional branch instructions JT0 and JNT0.
13	T1	**Test 1**: test input pin, directly tested by conditional branch instructions JT1 and JNT1. T1 also functions as an input to the 8-bit timer/event counter, using the STRT CNT instruction.
14	V_{SS}	**Ground**: circuit earth potential.
15	XTAL 1	**Crystal input**: connection to timing component (crystal) which determines the frequency of the internal oscillator; also the input for an external clock source.
16	XTAL 2	connection to the other side of the timing component.
17	RESET	**Reset input**: used to initialize the processor (active HIGH), or output of power-on-reset circuit.
18-25	P10-P17	**Port 1**: 8-bit quasi-bidirectional I/O port.
26, 27, 1, 2	P20-P23	**Port 2**: 4-bit quasi-bidirectional I/O port. P23 is the serial data input/output in serial I/O mode.
28	V_{DD}	**Power supply**: 1,8 V to 6 V.

PINNING (continued)

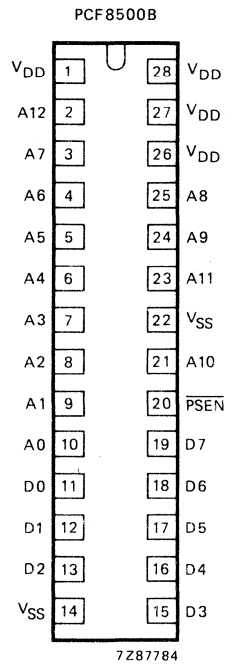


Fig. 3 Pinning diagram: PCF8500B 'Piggy-back' version top pinning; to access a 2732 or 2764 EPROM.

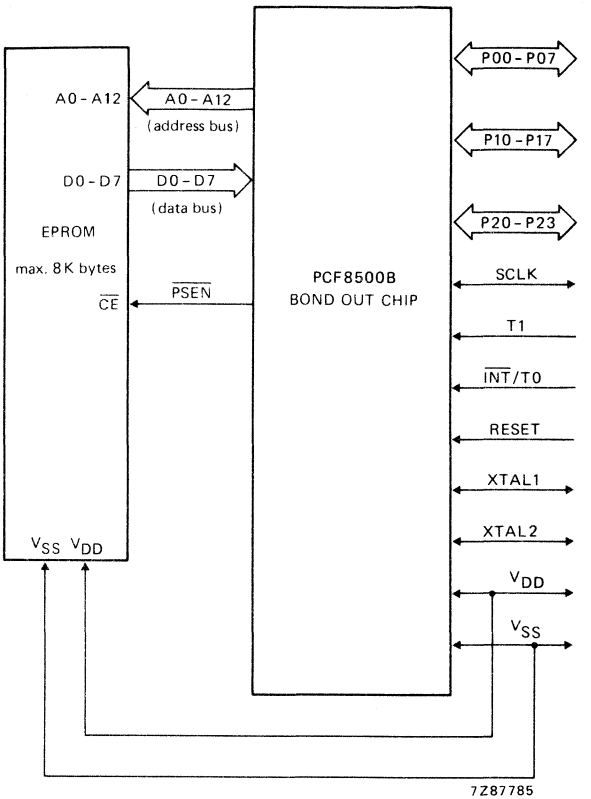


Fig. 3a Connection of EPROM to 'Piggy-back' package PCF8500B.

PIN DESIGNATION

14, 22	V_{SS}	**Ground**
1, 26-28	V_{DD}	**Power supply**
10-3, 25, 24, 21, 23, 2	A0-A12	**Address outputs**
11-13, 15-19	D0-D7	**Data**
20	$\overline{PSEN}$	**Program store enable**

Notes

1. RAM capacity of PCF8500B is 256 bytes.
2. Access time for ROMS/EPROMS to be below 7 x $1/f_{XTAL}$.
3. Pin 12 CE/$\overline{T0}$ is on the PCF8500B, inverted and labelled $\overline{INT}$/T0.

FUNCTIONAL DESCRIPTION

Bond-out version PCF8500F

The PCF8500F is a microcontroller that contains no on-board ROM, but has all address and data lines brought-out to access an external ROM or EPROM. This version has more pins than the PCD3343 with on-board ROM (see Fig. 1a). The RAM has 256 bytes. It can address 8 K bytes of ROM.

'Piggy-back' version PCF8500B

The PCF8500B is a special package that has standard pinning to the bottom which facilitates insertion as a mask-programmed device. An EPROM can be mounted on top in an additional socket. The total package height is greater than the standard DIL package. The RAM has 256 bytes and can also address 8 K bytes of program memory.

Program memory PCD3343

The program memory consists of 3072 bytes (8-bit words), in a read-only memory (ROM). Each location is directly addressable by the program counter. The memory is mask-programmed at the factory. Figure 4 shows the program memory map.

Four program memory locations are of special importance:

- Location 0; contains the first instruction to be executed after the processor is initialized (RESET),
- Location 3; contains the first byte of an external interrupt service subroutine,
- Location 5; contains the first byte of a serial I/O interrupt service subroutine,
- Location 7; contains the first byte of a timer/event counter interrupt service subroutine.

Program memory is arranged in banks of 2 K bytes, which are selected by SEL MB instructions. The program memory is further divided into location 'pages', each of 256 bytes. This latter division applies only for conditional branches. Memory bank boundaries can be crossed only by using the unconditional branch instructions after the appropriate memory bank has been selected. A CALL instruction can transfer control to a subroutine on any 'page'; RET and RETR instructions can transfer control from a subroutine back to the main program.

Data memory PCD3343

Data memory consists of 224 bytes (8-bit words), random-access data memory (RAM). All locations are indirectly addressable using RAM pointer registers; up to 16 designated locations are directly addressable. Memory also includes an 8-level program counter stack addressed by a 3-bit stack pointer. Figure 5 shows the data memory map.

Working registers

Locations 0 to 7 are designated as working registers, directly addressable by the direct register instructions. Ease of addressing, and a minimum requirement of instruction bytes to manipulate their contents, makes these locations suitable for storing frequently addressed intermediate results. This bank of registers can be selected by the SEL RB0 instruction.

Executing the select register bank instruction SEL RB1, designates locations 24 to 31 as working registers, instead of locations 0 to 7, and these are then directly addressable. This second bank of working registers may be used as an extension of the first or reserved for use during interrupt service subroutines saving the first bank for use in the main program. If the second bank is not used, locations 24 to 31 may serve as general purpose RAM.

The first locations of each bank contain the RAM pointer registers R0, R1, R0' and R1', which indirectly address all RAM locations.

All RAM locations make efficient program loop counters when used with the decrement register and test instruction DJNZ.

FUNCTIONAL DESCRIPTION (continued)

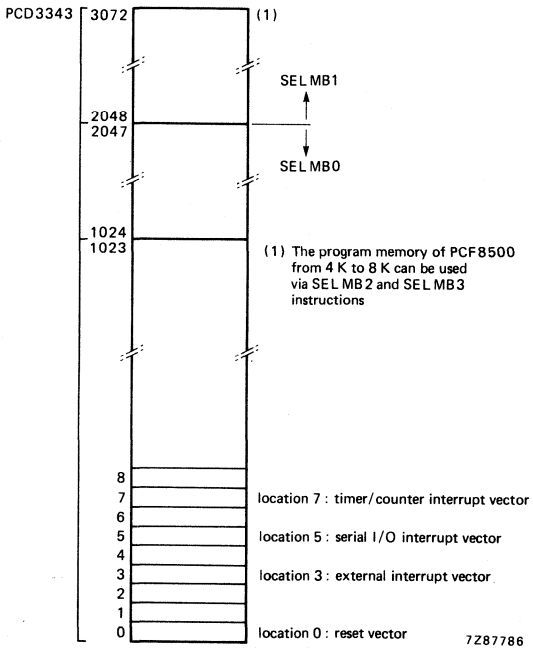


Fig. 4 Program memory map.

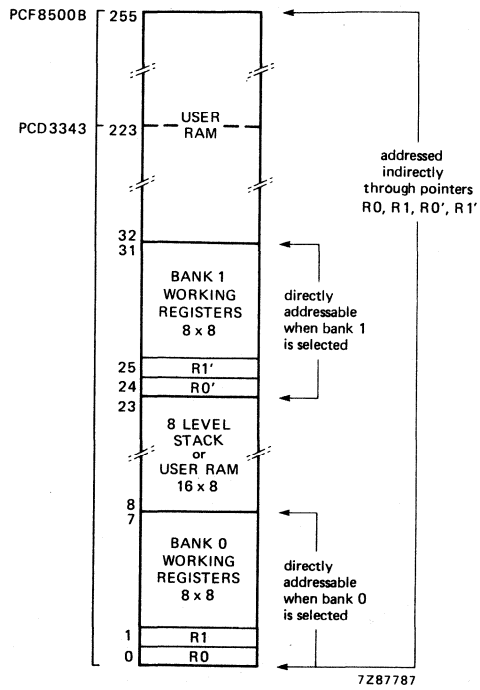


Fig. 5 Data memory map.

Program counter stack

Locations 8 to 23 may be designated as an 8-level program counter stack (2 locations per level), or as general purpose RAM. The program counter stack (Fig. 6) enables the processor to keep track of the return addresses and status generated by interrupts or CALL instructions by storing the contents of the program counter prior to servicing the subroutine. A 3-bit stack pointer determines which of the eight register pairs of the program counter stack will be loaded with next generated return address.

The stack pointer, when initialized to 000 by RESET, points to RAM locations 8 and 9. On the first subroutine CALL or interrupt, the contents of the program counter and bits 4, 6 and 7 of the program status word (PSW) are transferred to locations 8 and 9. The stack pointer increments by one and points to locations 10 and 11 ready for another CALL. Because an address may be up to 13 bits long, two bytes must be used to store each address.

At the end of a subroutine, which is signalled by a return instruction (RET or RETR), the stack pointer decrements by one and the contents of the register pair on top of the stack are transferred to the program counter. The saved PSW bits are transferred to the PSW only by the RETR instruction.

If not all 8 levels of subroutine and interrupt nesting are used, the unused portion of the stack may be used as any other indirectly addressable RAM locations. Possible locations from 32 to 223 may be used for storage of program variables or data.

Nesting of subroutines within subroutines can continue up to 8 times without overflowing the stack. If overflow does occur the deepest address stored (locations 8 and 9) will be overwritten and lost since the stack pointer overflows from 111 to 000. It also underflows from 000 to 111.

The value of the saved contents of the program counter is different for an interrupt CALL compared to a normal CALL to subroutine. With an interrupt CALL, the program counter return address is saved; with a subroutine CALL, the saved program counter value is one less than the program counter return address.

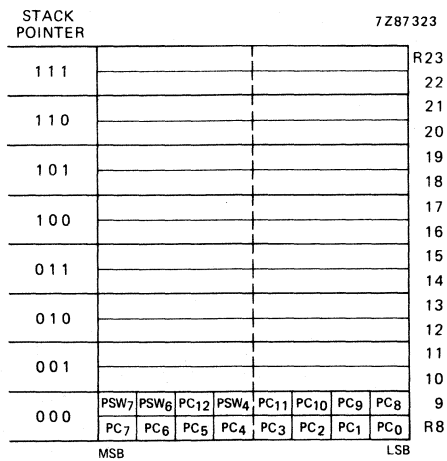


Fig. 6 Program counter stack.

IDLE and STOP modes

IDLE mode

When the microcontroller enters the IDLE mode via the IDLE instruction (H'01') the oscillator, timer/counter and serial I/O are kept running. The microcontroller exits from the IDLE mode by one of three interrupts if they are enabled or by activating a RESET. If the interrupt is not enabled the processor will remain in the IDLE mode. An active signal on the RESET pin restarts the microcontroller and a normal RESET sequence is executed (see Fig. 7).

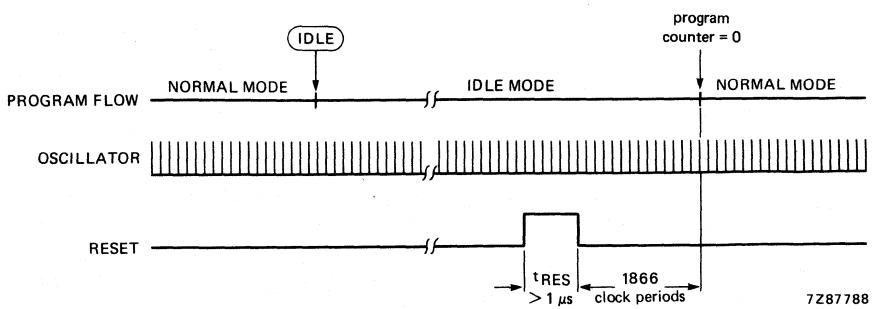


Fig. 7 Exit from IDLE mode via a RESET.

FUNCTIONAL DESCRIPTION (continued)

An active signal coming from an enabled interrupt causes the execution of the normal interrupt routine since normal interrupt scanning is still being carried out. A LOW-to-HIGH transition on the external interrupt pin (CE/$\overline{T0}$) reactivates the microcontroller. A HIGH level applied to CE/$\overline{T0}$ will reactivate the microcontroller only in the STOP mode. Thus, if CE/$\overline{T0}$ was HIGH before the microcontroller entered the IDLE mode, it must go LOW before the microcontroller can be reactivated (see Fig. 8).

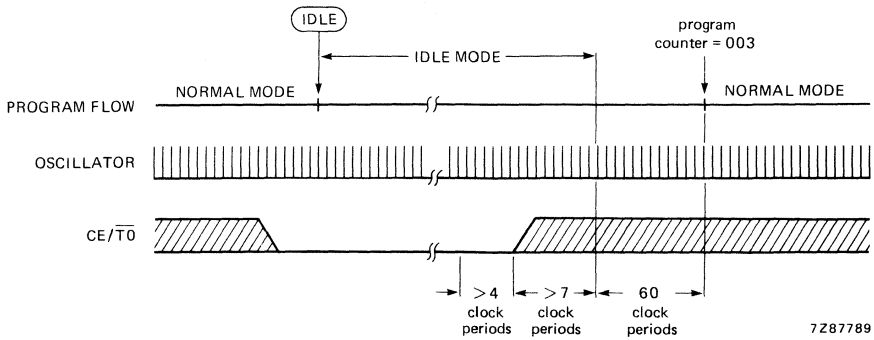


Fig. 8 Exit from IDLE mode via an interrupt.

Wake-up from the IDLE mode is ensured when CE/$\overline{T0}$ is LOW for 4 CP (clock periods) followed by a HIGH for 7 CP. After the initial forced CALL H'003' operation (60 CP) the program continues with the external interrupt service routine.

STOP mode

The microcontroller enters the STOP mode by the STOP instruction (H'22'). The oscillator is switched off. The internal status of the CPU, RAM contents and the state of I/O ports are not affected. The microcontroller can be brought-out of the STOP mode by an active signal at the external interrupt input or by an external RESET signal. When one of these two signals is applied an internal delay of 1866 CP is provided to ensure that all internal clocks are operating correctly before restart (see Fig. 9).

If the microcontroller exits from the STOP mode by activating RESET, a normal RESET sequence is executed.

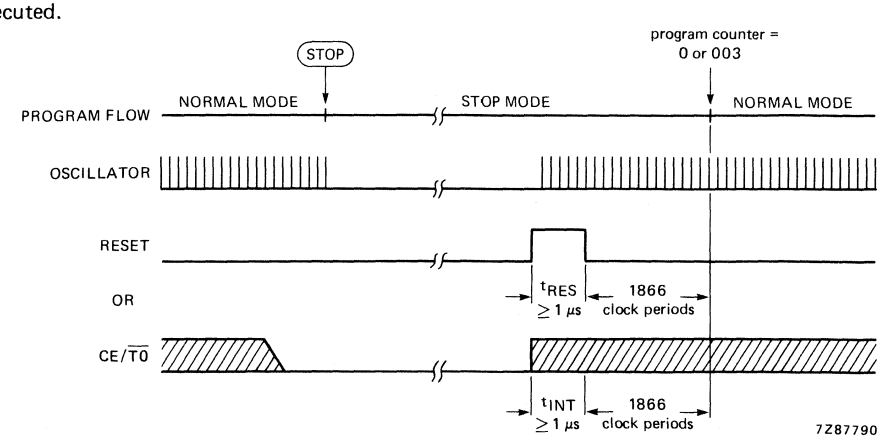


Fig. 9 Entering and exiting the STOP mode.

If the microcontroller exits the STOP mode by pulling the external interrupt input pin HIGH, an interrupt sequence is executed only if the external interrupt is enabled. In this event the microcontroller resumes the normal program sequence after returning from the interrupt routine, as in the normal mode. If the interrupt is not enabled, it continues the normal program sequence, executing the instruction following the STOP instruction.

The microcontroller is restarted by a HIGH level applied at the CE/$\overline{T0}$ pin, and not by a LOW-to-HIGH transition as in a normal interrupt mechanism.

When the CE/$\overline{T0}$ level is active during the STOP instruction then no STOP is executed.

A HIGH level on the external interrupt input of at least 1 μs will cause the microcontroller to exit the STOP mode.

I/O facilities

The PCD3343 family has 23 I/O lines arranged as:

- Port 0 parallel port of 8 lines (P00 to P07)
- Port 1 parallel port of 8 lines (P10 to P17)
- Port 2 parallel port of 4 lines (P20 to P23)
- SCLK serial I/O consisting of a data line shared with a parallel port line (P23) and a separate clock line SCLK
- CE/$\overline{T0}$ external interrupt and test input. When used as a test input can be directly tested by conditional branch instructions JT0 and JNT0
- T1 test input which can alter program sequences when tested by conditional jump instructions JT1 and JNT1. T1 also functions as an input to the 8-bit timer/event counter.

Parallel ports

All parallel ports can be used as outputs or inputs, their structure is quasi-bidirectional.
Output data written to a port is latched and remains unchanged until rewritten.
Input data is not latched and so must be present until read by an input instruction.

Input lines are fully CMOS compatible, output lines can drive one LS-TTL or CMOS load.

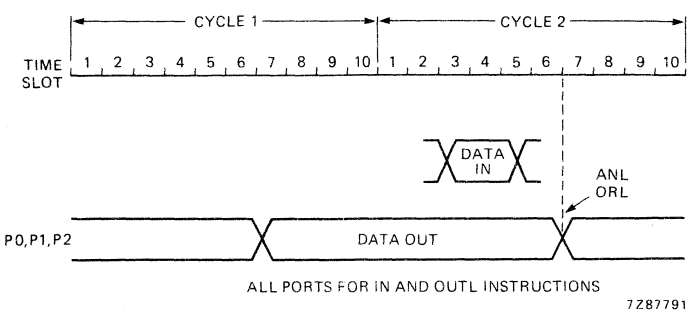


Fig. 10 Timing diagram of all ports on IN and OUTL instructions; for ANL and ORL instructions, the ports change on the time slot 7 of cycle 2.

Fig. 11 shows the quasi-bidirectional I/O interface with push-pull output and switched pull-up current source.
Each line is pulled up to V_{DD} via a constant current source (TR4), which is enabled via TR3 whenever one of the two output latches contains a logic 1. This current provides sufficient source current for a TTL HIGH level, yet can be pulled LOW by an external CMOS device, thus allowing the same pin to be used for both input and output.

FUNCTIONAL DESCRIPTION (continued)

When a logic 1 is written to the line for the first time (MQ = 1, SQ = 0), TR2 is switched on for the duration of the internal write pulse (one oscillator period), to provide a fast transition from logic 0 to logic 1. Subsequent writing of a logic 1 to the port lines will not switch TR2 on. This prevents unnecessary current through external components connected to the port lines of the same port which might be in the input mode and also connected to ground.

When a logic 0 is written to the line, TR3 switches off the current source. Current sinking capability is provided by TR1, which is now switched on. When used as an input, a logic 1 must first be written to the line, otherwise TR1 will remain low impedance.

In telephone applications this switched pull-up source may not be sufficient. Therefore the PCD3343 offers the possibility to select individually 19 of the 20 parallel port pins (not P23), by the following mask options:

Option 1- STANDARD PORT; quasi-bidirectional I/O with switched pull-up current source of 100 μA (typ.) and P-channel booster transistor TR2 (1,5 mA). TR2 is only active during 1 clock cycle (0,28 μs at 3,58 MHz).

Option 2- OPEN DRAIN; quasi-bidirectional I/O with only an N-channel open drain output. Application as an output requires connection of an external pull-up resistor (Fig. 12).

Option 3- PUSH-PULL OUTPUT; drive capability of the output will be 1,5 mA (typ.) at V_{DD} = 3 V in both polarities. To avoid a large current flowing through the output transistors during the input mode, these push-pull pins must only be used as outputs (Fig. 13).

Also, individual mask selection of the RESET state of these port pins can be achieved by appending the following options S and R to options 1, 2 or 3.

Option S-SET; after RESET this pin will be initialized to HIGH.

Option R-RESET; after RESET this pin will be initialized to LOW.

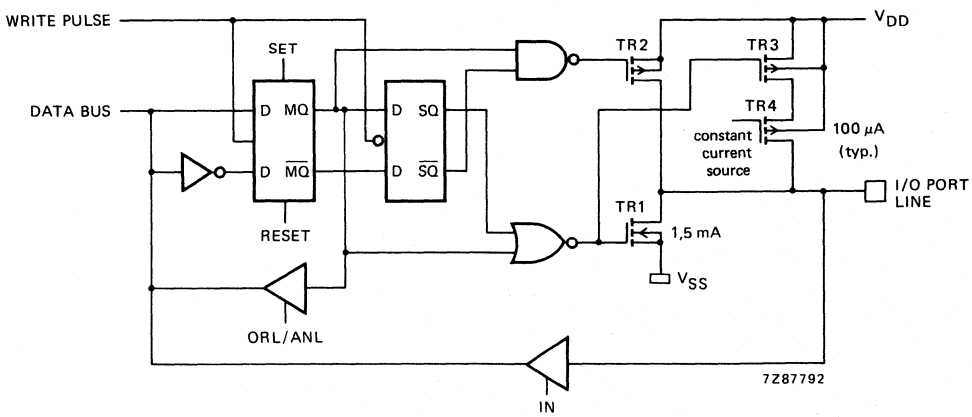


Fig. 11 Standard output with switched pull-up current source.

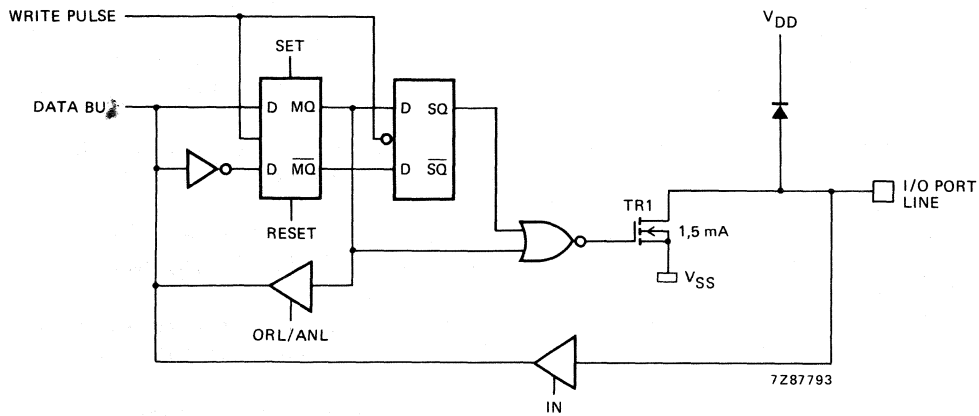


Fig. 12 Open drain output.

DEVELOPMENT DATA

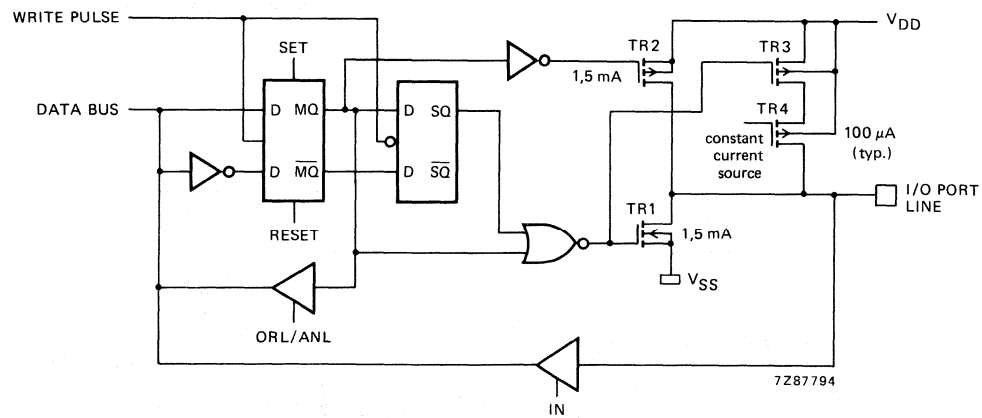


Fig. 13 Push-pull output.

FUNCTIONAL DESCRIPTION (continued)

Serial I/O (SIO)

The PCD3343 has an on-chip serial I/O interface. This SIO interface is a versatile feature in an intelligent telephone set, as shown in application diagram Fig. 32.

In this application the SIO is used to communicate with the different peripherals, such as:

- DTMF generator (PCD3312)
- LCD drivers (PCF8577)
- External RAM (PCD8571)
- Clock calendar (PCB8573)

No extra hardware is required for decoding, addressing and data processing.

Whereas a normal microcontroller must regularly monitor the serial data bus for the presence of data, the serial I/O interface detects, receives and converts the serial data stream into parallel format without interrupting the execution of the current program. An interrupt is sent to the PCD3343 only when a complete byte is received. It then reads the data byte in one instruction. Likewise during transmission the serial I/O interface performs parallel to serial conversion and subsequent serial output of the data. The microcontroller is only interrupted in the execution of its programmed tasks when a complete byte has been transmitted.

The design of the PCD3343 serial I/O system allows any number of devices from PCF8500 family (clips) to be connected via the two-line serial bus. The ability of any devices to communicate, without interrupting the operation of any other devices on the bus, is an outstanding attribute of the system. This is achieved by allocating a specific 7-bit address to each device and providing a system whereby a device reacts only to a message prefixed with its own address or the 'general CALL' address. Address recognition is performed by the interface hardware so that operation of the microcontroller need only be interrupted when a valid address has been received. This saves significant processing time and memory space compared with a conventional microcontroller employing a software serial interface. When the addressing facility is not required, for instance in a system with only two microcontrollers, direct data transfer without addressing can be performed. In multi-master systems, an automatically invoked arbitration procedure prevents two or more devices from continuing simultaneous transmission.

In NORMAL (running) and IDLE mode, the serial I/O logic remains active; its internal system clock will be switched off when there is no activity on the serial bus.

After execution of the STOP instruction, the oscillator of the PCD3343 is switched off. This means that the serial I/O logic will remain in the state it was at the occurance of the STOP instruction. To avoid "bus block" problems and to assure correct start-up of the bus after exit from the STOP mode, the user should disable the serial logic (ESO = 0) prior to the execution of the STOP instruction. This must be carried out only when the PCD3343 has finished a serial data transfer.

Serial I/O interface

Figure 14 shows the serial I/O interface. The clock line of the serial bus has exclusive use of pin 3 (SCLK) while the data line shares pin 2 (serial data) with the I/O line P23 of port 2. When the serial I/O is enabled, P23 is disabled as a parallel port line; (P23 and SCLK only open drain).

The microcontroller and interface communicate via the internal microcontroller bus and the Serial Interrupt Request line. Data and information controlling the operation of the interface are stored in four registers:

- Data shift register (S0)
- Serial I/O interface status word (S1)
- Serial clock control word (S2)
- Address register

Data shift register (S0)

Register S0 converts serial data to parallel format and vice versa. A pending interrupt is generated only after a complete byte has been transmitted, or after a complete data byte, specific address or 'general CALL' address has been received. The most significant bit is transmitted first.

Serial I/O interface status word (S1)

Register S1 provides information concerning the state of the interface and stores information from the microcontroller. Bits 0 to 3 are duplicated: control bits in these positions can only be written by the microcontroller, while interface bits can only be read.

MST and TRX (see Table 1)

These bits determine the operating mode of the serial I/O interface.

Table 1 Operating modes of the serial I/O interface

MST	TRX	operating mode
0	0	slave receiver
1	0	master receiver
0	1	slave transmitter
1	1	master transmitter

BB: Bus Busy.

This is the flag which indicates the status of the bus.

PIN: Pending Interrupt Not

PIN = '0' indicates the presence of a pending interrupt, which will cause a Serial Interrupt Request when the serial interrupt mechanism is enabled.

ESO: Enable Serial output

The ESO flag enables/disables the serial I/O interface: ESO = '1' enables, ESO = '0' disables. ESO can only be written by software.

BC0, BC1 and BC2

Bits BC0, BC1 and BC2 indicate the number of bits received or transmitted in a data stream. These bits can only be written by software.

AL: Arbitration Lost

The arbitration lost flag is set by hardware when the serial I/O interface, as master transmitter, loses a bus arbitration procedure.

AAS: Addressed As Slave

This flag is set by hardware when the interface detects either its own specific address or the 'general CALL' address as the first byte of a transfer and the interface has been programmed to operate in the address recognition mode.

AD0: Address Zero

This flag is set by hardware after detection of the 'general CALL' address when the interface is operating in the address recognition mode.

LRB: Last Received Bit

This contains either the last data bit received or, for a transmitting device in the acknowledgement mode, the acknowledgement signal from the receiving device.

Bits AL, AAS, AD0 and LRB can only be read by software.

FUNCTIONAL DESCRIPTION (continued)

Serial clock control word (S2)

Bits 0 to 4 of the clock control register S2 are used to set the frequency of the serial clock signal. When a 3,58 MHz crystal is used, the frequency of the serial clock can be varied between 92 kHz and 580 Hz (see Table 2). An asymmetrical clock with a HIGH-to-LOW ratio of 3 : 1 can be generated using bit 5. The asymmetrical clock allows a microcontroller more time per clock period for sampling the data line, making the timing of this action less critical. Bit 6 can be used to activate the acknowledge mode of the serial I/O. S2 is a write only register.

Address register

The address register contains the 7-bit address back-up latches and the bit (ALS) used to enable/disable the address recognition mode. The address register can be written using the MOV S0, A and MOV S0, # data instructions, but only when ES0 = '0'.

Serial I/O interrupt logic

An EN SI instruction enables and a DIS SI instruction disables the interrupt logic. When the logic is enabled, a pending interrupt results in a serial I/O interrupt to the processor, causing a CALL to location 5 in the ROM. When disabled, the presence of an interrupt is still indicated by PIN in S1, allowing the interrupt to be serviced. However, vectored interrupt will not occur.

DEVELOPMENT DATA

Table 2 SIO clock pulse frequency control when using a 3,58 MHz crystal

hexadecimal S20-S24 code	divisor	f_{SCLK} (kHz) (approximate)
0	not allowed	
1	39	92
2	45	80
3	51	70
4	63	57
5	75	48
6	87	41
7	99	36
8	123	29
9	147	24
A	171	21
B	195	18
C	243	15
D	291	12
E	339	11
F	387	9,2
10	483	7,4
11	579	6,2
12	675	5,3
13	771	4,6
14	963	3,7
15	1155	3,1
16	1347	2,7
17	1539	2,3
18	1923	1,9
19	2307	1,6
1A	2691	1,3
1B	3075	1,2
1C	3843	0,93
1D	4611	0,78
1E	5379	0,67
1F	6147	0,58

FUNCTIONAL DESCRIPTION (continued)

Table 3 Serial I/O addresses for telephony peripherals

type	address 7	6	5	4	3	2	1	0	description
PCF8570A	1	0	1	0	A2	A1	X	R/$\overline{W}$	2 K RAM
PCF8570	1	0	1	0	A2	A1	A0	R/$\overline{W}$	2 K RAM
PCD8571	1	0	1	0	A2	A1	A0	R/$\overline{W}$	1 K RAM
PCD3311	0	1	0	0	1	0	A0	R/$\overline{W}$	DTMF dialler
PCD3312	0	1	0	0	1	0	A0	R/$\overline{W}$	DTMF dialler
PCE2111	0	0	0	0	0	0	1	0	LCD driver *
PCD8573	1	1	0	1	0	A1	A0	R/$\overline{W}$	clock calendar
PCF8574	0	0	1	1	A2	A1	A0	R/$\overline{W}$	8-bit I/O expander
PCF8576	0	1	1	1	0	0	SA0	R/$\overline{W}$	1 : 4 LCD driver
PCF8577	0	1	1	1	0	1	0	R/$\overline{W}$	1 : 2 LCD driver

* LCD driver requires an additional enable line.

DEVELOPMENT DATA

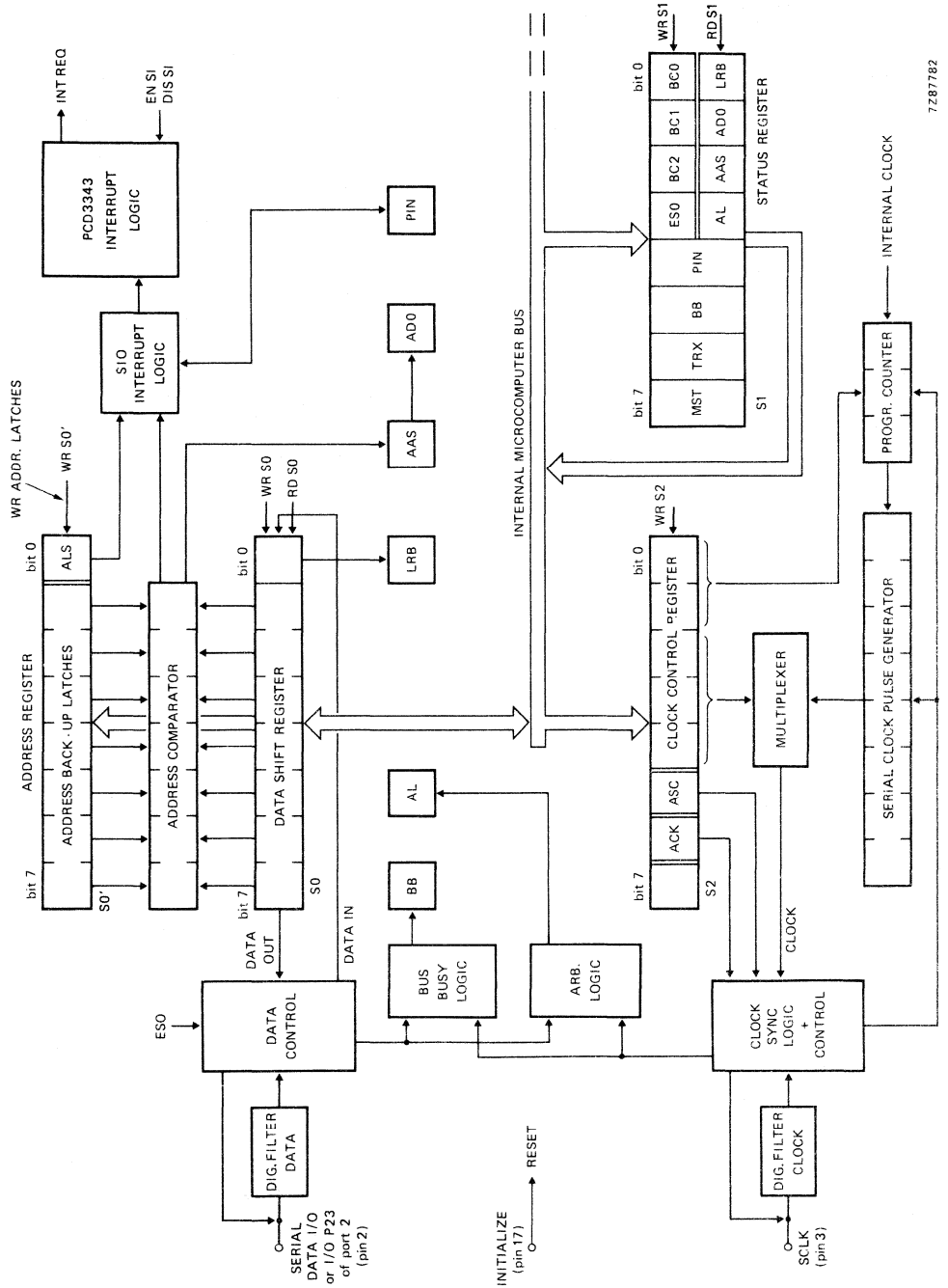


Fig. 14 Serial I/O interface.

FUNCTIONAL DESCRIPTION (continued)

Interrupts (see Fig. 15)

When the external interrupt is enabled, a LOW-to-HIGH transition on the CE/$\overline{T0}$ input initiates an external interrupt subroutine which causes a CALL to program memory location 3 following completion of the current instruction.
The interrupt must remain enabled until the interrupt instruction is completed, otherwise the next instruction of the main program will be executed. Serial I/O interrupt, when enabled, causes a CALL to location 5, and a timer/event counter overflow forces a CALL to location 7 when the timer interrupt is enabled.

When an interrupt subroutine starts, the contents of the program counter and bits 4, 6 and 7 of the PSW have been saved in the program counter stack. Accumulator contents have to be saved by software. Interrupt acknowledgement can be carried out by software via port pins. All interrupt subroutines must reside in memory bank 0.

Since the interrupt system is single level, once an interrupt is detected, all further interrupt requests are latched, but ignored, pending a RETR instruction to re-enable the interrupt input logic. After executing RETR, the program continues in the main part; this is independent of the occurance of a second interrupt during the running of the first routine. If 2 or 3 interrupts occur simultaneously, their priority is:

- (1) external
- (2) serial I/O
- (3) timer/event counter

An external interrupt can be generated by using the timer/counter in the event counter mode. The counter is first preset to (H'FF'), then EN TCNTI instruction is executed. A LOW-to-HIGH transition of the T1 input will then initiate an interrupt subroutine and cause a CALL to location 7.

On execution of a DIS I instruction, the PCD3343 always clears the digital filter/latch and the External Interrupt Flag.

The Timer Flag (TF) is reset only when the JTF or JNTF instruction is executed or after RESET.

The Timer Interrupt Flag is set when timer overflow occurs, only if the timer interrupt is enabled.

The microcontroller will exit the IDLE mode when any one of the following three interrupts is enabled:

- External
- Serial I/O
- Timer/event counter

There is no internal pull-up or pull-down device connected to the external interrupt input (pin 12). If required pin 12 must be externally connected to a resistor ($R \leqslant 100\ k\Omega$). When the external interrupt is not used pin 12 must be connected to V_{SS}.

DEVELOPMENT DATA

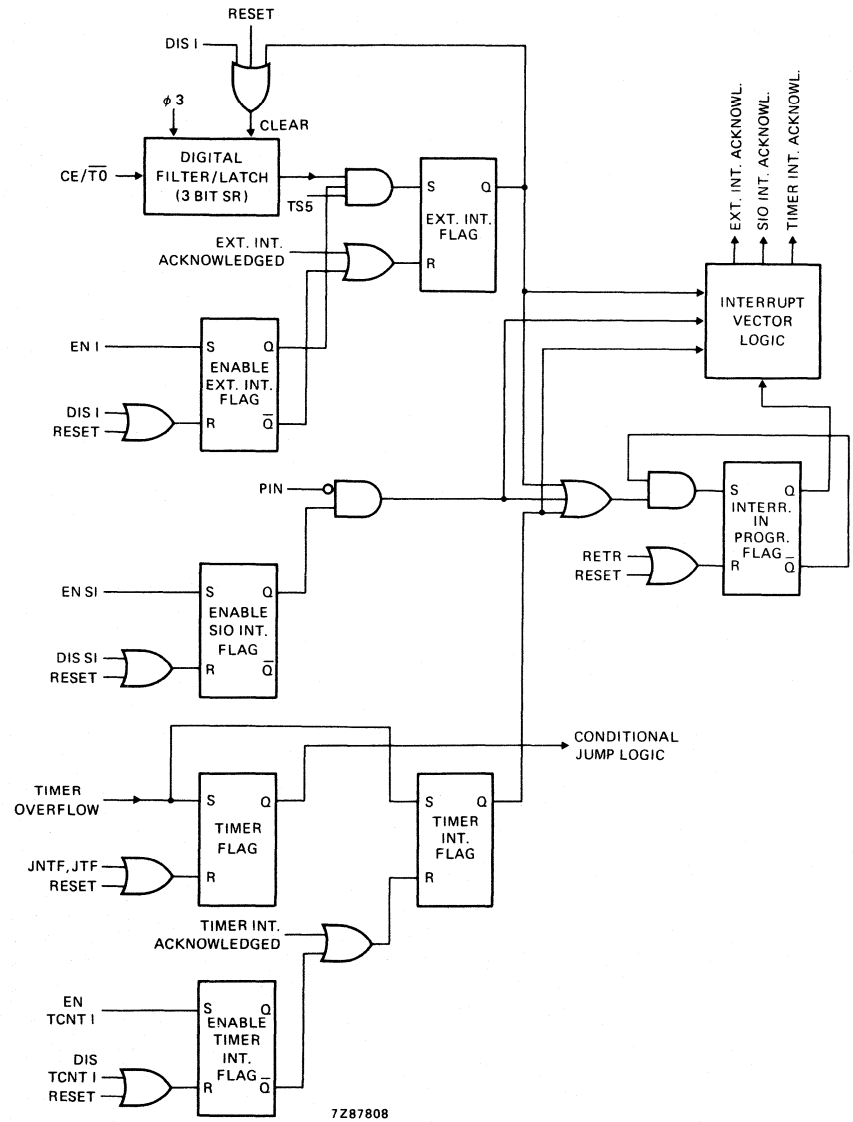


Fig. 15 Interrupt logic.

Notes to figure 15

1. CE/$\overline{T0}$ positive edge is always latched in the digital filter/latch.
2. Correct interrupt timing in ensured when CE/$\overline{T0}$ is LOW for >4 CP followed by a HIGH for >7 CP.
3. When the interrupt in progress flag is set, further interrupts are latched but ignored, until RETR is executed.
4. A DIS I instruction always clears a pending external interrupt.

FUNCTIONAL DESCRIPTION (continued)

Oscillator (see Fig. 16)

The 3,58 MHz oscillator can be inhibited by the STOP instruction under software control. It is also inhibited when a low-voltage condition is present to prevent discharge of a weak back-up battery.

Provided the supply voltage is within the operating range, the oscillator will be restarted after a STOP instruction by a HIGH level at either the CE/$\overline{T0}$ or RESET pin.

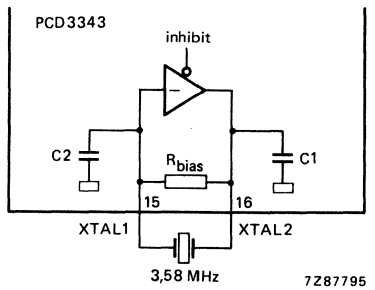


Fig. 16 Oscillator with integrated elements.

The oscillator has the output drive capability for the DTMF generator (PCD3311/3312) via pin 16 (XTAL 2). An external clock can be applied to pin 15 (XTAL 1). A machine cycle consists of 10 time slots, each time slot being 3 oscillator periods.

In telephony applications the 3,58 MHz crystal provides a 8,4 μs machine cycle. The range of the clock frequency is from 100 kHz up to a maximum which is a function of the supply voltage (see Fig. 23).

Timer/event counter (see Fig. 17)

An internal 8-bit up-counter is provided. This can count external events, modulo-32 machine cycles, or machine cycles directly. Table 4 gives the instructions that control the counter and the prescaler, and the functions performed.

When used as a timer, the input to the counter is either the overflow or input of a 5-bit prescaler. When used as an event counter, LOW-to-HIGH transitions on pin 13 (T1) are counted. The maximum rate at which the counter may be incremented is once every machine cycle (182,6 kHz for a 8,4 μs machine cycle). When the counter overflows, the timer flag is set. The flag can be tested and reset using the JTF (jump if timer flag = 1) or JNTF instruction. Overflow also generates an interrupt to the processor via setting of the Timer Interrpt Flag when the timer/event counter interrupt is enabled.

Table 4 Timer/event counter control

function	timer mode modulo-1, modulo-32*	counter mode
CLEAR	MOV T,A (A) = 0 or RESET	MOV T,A (A) = 0 or RESET
PRESET	MOV T,A	MOV T,A
START	STRT T	STRT CNT
STOP	STOP TCNT or RESET	STOP TCNT or RESET
TEST	JTF/JNTF	JTF/JNTF
READ**	MOV A,T	MOV A,T

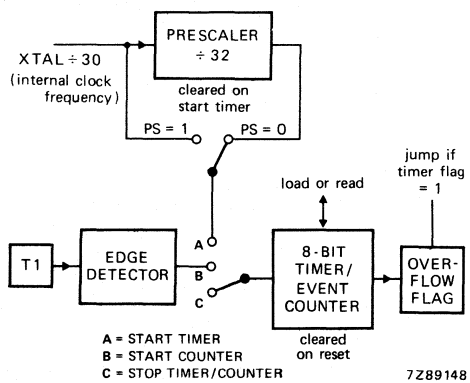


Fig. 17 Timer/event counter.

DEVELOPMENT DATA

Program status word (see Fig. 18)

The program status word (PSW) is an 8-bit word (1 byte) in the CPU which stores information about the current status of the microcontroller.
The PSW bits are:

- Bits 0 to 2 — stack pointer bits (SP_0, SP_1, SP_2)
- Bit 3 — prescaler select (PS);
  0 = modulo-32; 1 = modulo-1 (no prescaling)
- Bit 4 — working register bank select (RBS);
  0 = register bank 0; 1 = register bank 1
- Bit 5 — not used (1)
- Bit 6 — auxiliary carry (AC); half-carry bit generated by an ADD instruction and used by the decimal adjust instruction DA A
- Bit 7 — carry (CY); the carry flag indicates that previous operation has resulted in an overflow of the accumulator.

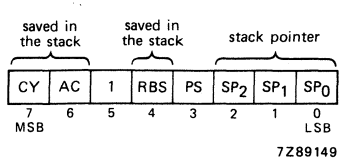


Fig. 18 Program status word.

* With prescaler select, PS = 0, the timer counts modulo-32 machine cycles, with PS = 1 it counts modulo-1 cycles (prescaler not used); prescaler cleared with STRT T, prescaler not readable.

** READ does not disturb the counting process.

FUNCTIONAL DESCRIPTION (continued)

Program status word (continued)

All bits can be read using the MOV A, PSW instruction. Bits 7 and 6 are set and cleared by CPU operation. Bit 4 can be changed by a SEL RB instruction, bit 3 by the MOV PSW, A instruction, and bits 0, 1 and 2 by the CALL, RET or RETR instructions and in the event of an interrupt. Bits 7, 6 and 4 are stored in the program counter stack during subroutine and interrupt calls. These bits are restored in the PSW with a RETR (return and restore) instruction which must be used at the end of an interrupt and can be used at the end of a normal subroutine. The RET instruction has no restore feature and cannot be used at the end of an interrupt.

Program counter (see Fig. 19)

A 13-bit program counter is used to facilitate 8 K bytes of ROM being addressed. The arrangement of the bits is shown in figure 19. During an interrupt subroutine PC_{11} and PC_{12} are forced to logic 0. All 13 bits are saved in the stack during CALL and interrupt routines.

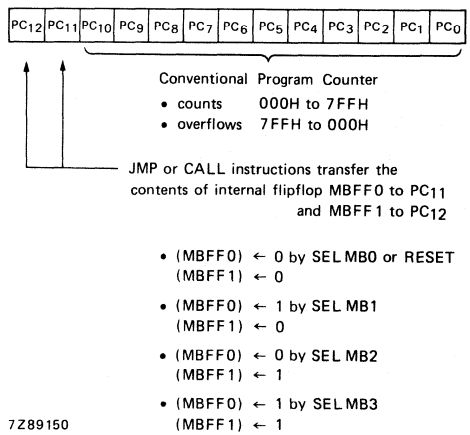


Fig. 19 Program counter.

Central processing unit

The PCD3343 has arithmetic, logical and branching capabilities. The DA A, SWAP A and XCHD instructions simplify BCD arithmetic and the handling of nibbles. The MOVP A,@A instruction permits efficient table look-up from the current ROM page.

Conditional branch logic

The conditional branch logic within the processor enables several conditions, internal and external to the processor, to be tested by the user's program. Table 5 lists the conditional jump instructions used to change the program sequence. The DJNZ instruction decrements a designated register or data memory location and branches if the contents are not zero. This instruction is useful for looping control. The JMPP@A instruction allows multiway branches to destinations determined by the contents of the accumulator.

Table 5 Conditional branches

test	jump condition	jump instruction
accumulator	all bits zero	JZ
	any bit non-zero	JNZ
accumulator bit test	1	JB0 to JB7
carry flag	1	JC
	0	JNC
timer overflow flag	1	JTF
	0	JNTF
test input T0	1	JNT0
	0	JT0*
test input T1	1	JT1
	0	JNT1
register	non-zero	DJNZ

Test input T1 (pin 13)

The T1 input line can be used as:

- A test input for branch instructions JT1 and JNT1
- An external input to the event counter

When used as a test input:

- JT1 instruction tests for logic 1 level
- JNT1 instruction tests for logic 0 level

When used as an input to the event counter, T1 must be LOW for > 4 CP, followed by a HIGH for > 4 CP. The transition can be recognized with a repetition rate of once per 30 oscillator clock periods (1 machine cycle).

There is no internal pull-up or pull-down resistor connected to the T1 input. If required it must be externally connected to a resistor ($R = \leqslant 100\ k\Omega$). When T1 is not used pin 13 must be connected to V_{DD} or V_{SS}.

Reset (pin 17)

A positive-going signal on the RESET input/output:

- Sets the program counter to zero
- Selects location 0 of memory band 0 and register bank 0
- Sets the stack pointer to zero (000); pointing to RAM address 8
- Disables the interrupts (external, timer and serial I/O)
- Stops the timer/event counter, then sets it to zero
- Sets the timer prescaler to modulo-32
- Resets the timer flag
- Sets all ports according to reset states
- Sets the serial I/O to slave receiver mode and disables the serial I/O
- Cancels IDLE and STOP mode

After the voltage is applied to RESET an internal delay of 1866 CP is introduced before the microcontroller commences operation.

* Because of the inverted interrupt input CE/$\overline{T0}$ the conditional jump JT0 is also inverted.

FUNCTIONAL DESCRIPTION (continued)

Power-on-reset and low-voltage detection (see Fig. 20)

In telephony applications, correct operation of the PCD3343 during moments of slowly changing supply voltages and low-voltage conditions is essential. This is achieved by the addition of an internal power-on-reset and low-voltage detection circuit.
To allow an external RESET signal being fed into the PCD3343, the reset pin (pin 17) has been configured as an input/output.

While a reset condition exists in the detection circuit, pin 17 is pulled HIGH by TR1 controlled by the reset circuit.
When the reset condition is not present a pull-down current source (TR2) will be activated. TR2 forces pin 17 LOW thus removing the RESET signal from the microcontroller.

Since the level at pin 17 is recognized by the microcontroller, the reset time constant can be stretched by connecting an external capacitor between V_{DD} and pin 17 (see Fig. 22).

The signal at pin 17 can also be used as an output to reset other devices in the system.

The internal reset circuit monitors the PCD3343 supply voltage. If the voltage drops below the switching level (typ. 1,3 V), a reset (HIGH) is applied to pin 17. This reset is removed (pin 17 goes LOW), after a fixed delay (t_d), when the supply voltage rises above the switching level again. The delay ensures a complete reset even when the supply voltage quickly rises above switching level after initial switch-on.

During a low-voltage condition the oscillator is inhibited to prevent complete discharge of a weak battery. The timing of the power-on-reset and low-voltage detection circuit is shown in figure 21.

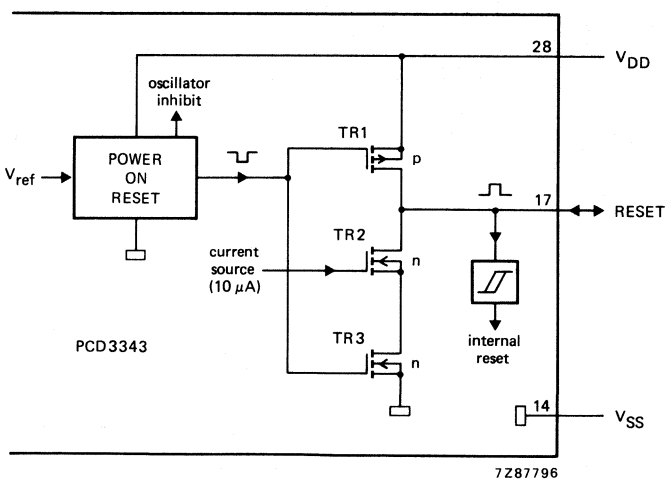
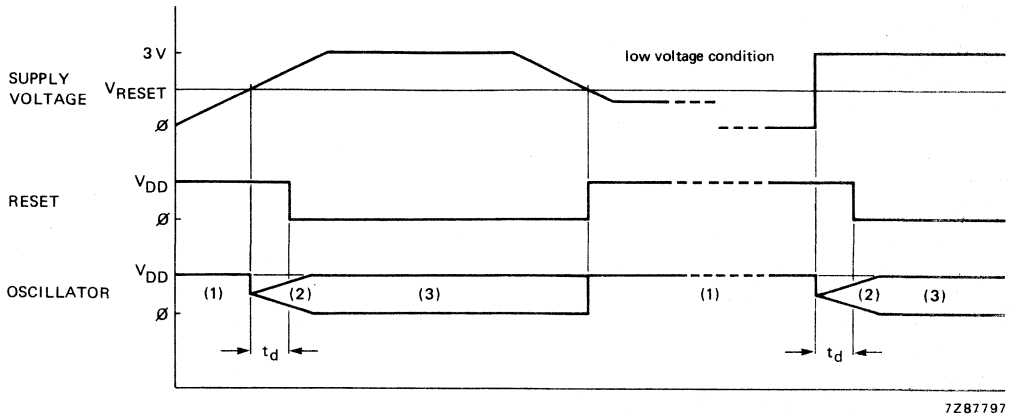


Fig. 20 Power-on-reset configuration.



Where: (1) Oscillator inhibited
(2) Oscillator starting
(3) Oscillator running, but may be stopped with a STOP condition

Fig. 21 Timing of power-on-reset and low-voltage detection.

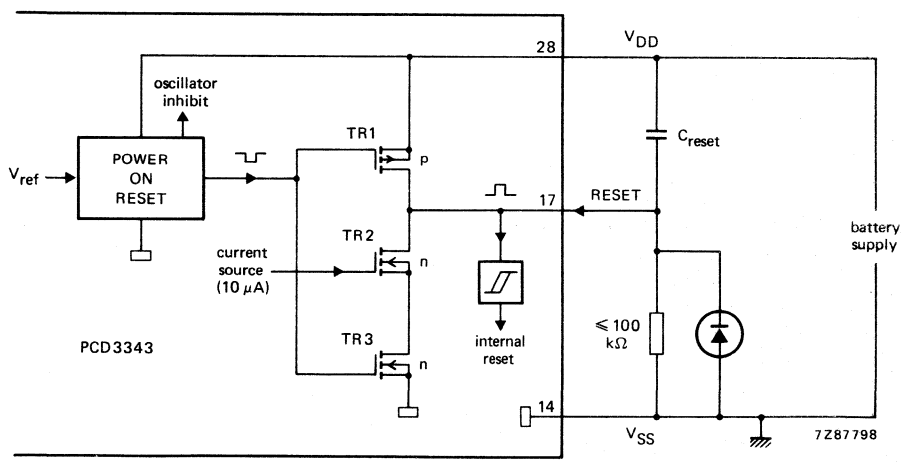


Fig. 22 Stretched power-on-reset with external capacitor.

INSTRUCTION SET

The PCD3343 instruction set consists of over 80 one and two byte instructions and is based on the MAB8048 instruction set. New instructions include those for serial I/O operation and memory bank selection. Program code efficiency is high because all RAM locations and all ROM locations on a 256 byte page require only a single byte address.

Table 8 gives the instruction set of the PCD3343. Table 7 shows the instruction map and Table 6 details the symbols and definition descriptions that are used.

Table 6 Symbols and definitions used in Table 8

symbol	definition description
A	accumulator
addr	program memory address
Bb	bit designation (b = 0-7)
RBS	register bank select
C	carry bit (bit CY)
CNT	event counter
D	mnemonic for 4-bit digit (nibble)
data	8-bit number or expression
I	interrupt
MB	memory bank
MBFF	memory bank flip-flop
P	mnemonic for 'in-page' operation
PC	program counter
Pp	port designation (p = 0, 1 or 2)
PSW	program status word
RB	register bank
Rr	register designation (r = 0-7)
Sn	serial I/O register
SP	stack pointer
T	timer
TF	timer flag
T0, T1	test 0 and 1 inputs
#	immediate data prefix
@	indirect address prefix
(X)	contents of X
((X))	contents of location addressed by X
←	is replaced by
↔	is exchanged with

Table 7 PCD3343 instruction map

first hexadecimal character of opcode ↓	second hexadecimal character of opcode															
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NOP	IDLE		ADD A,#data	JMP page 0	EN I	JNTF addr	DEC A	IN A,Pp 0 1 2				MOV A,Sn 0 1			
1	INC @Rr 0 1		JB0 addr	ADDC A,#data	CALL page 0	DIS I	JTF addr	INC A	INC Rr 0 1 2 3 4 5 6 7							
2	XCH A,@Rr 0 1		STOP	MOV A,#data	JMP page 1	EN TCNTI	JNTO addr	CLR A	XCH A,Rr 0 1 2 3 4 5 6 7							
3	XCHD A,@Rr 0 1		JB1 addr		CALL page 1	DIS TCNTI	JTO addr	CPL A	OUTL Pp,A 0 1 2				MOV Sn,A 0 1 2			
4	ORL A,@Rr 0 1		MOV A,T	ORL A,#data	JMP page 2	STRT CNT	JNT1 addr	SWAP A	ORL A,Rr 0 1 2 3 4 5 6 7							
5	ANL A,@Rr 0 1		JB2 addr	ANL A,#data	CALL page 2	STRT T	JT1 addr	DA A	ANL A,Rr 0 1 2 3 4 5 6 7							
6	ADD A,@Rr 0 1		MOV T,A		JMP page 3	STOP TCNT		RRC A	ADD A,Rr 0 1 2 3 4 5 6 7							
7	ADDC A,@Rr 0 1		JB3 addr		CALL page 3			RR A	ADDC A,Rr 0 1 2 3 4 5 6 7							
8				RET	JMP page 4	EN SI			ORL Pp,#data 0 1 2							
9			JB4 addr	RETR	CALL page 4	DIS SI	JNZ addr	CLR C	ANL Pp,#data 0 1 2				MOV Sn,#data 0 1 2			
A	MOV @Rr,A 0 1			MOVP A,@A	JMP page 5	SEL MB2		CPL C	MOV Rr,A 0 1 2 3 4 5 6 7							
B	MOV @Rr,#data 0 1		JB5 addr	JMPP @A	CALL page 5	SEL MB3			MOV Rr,#data 0 1 2 3 4 5 6 7							
C	DEC @Rr 0 1				JMP page 6	SEL RB0	JZ addr	MOV A,PSW	DEC Rr 0 1 2 3 4 5 6 7							
D	XRL A,@Rr 0 1		JB6 addr	XRL A,#data	CALL page 6	SEL RB1		MOV PSW,A	XRL A,Rr 0 1 2 3 4 5 6 7							
E	DJNZ @Rr,addr 0 1				JMP page 7	SEL MB0	JNC addr	RL A	DJNZ Rr, addr 0 1 2 3 4 5 6 7							
F	MOV A,@Rr 0 1		JB7 addr		CALL page 7	SEL MB1	JC addr	RLC A	MOV A,Rr 0 1 2 3 4 5 6 7							

INSTRUCTION SET (continued)

Table 8 Instruction set

	mnemonic	opcode (hex.)	bytes/ cycles	description	function		notes
ACCUMULATOR	ADD A, Rr	6*	1/1	Add register contents to A	(A)←(A) + (Rr)	r = 0–7	1
	ADD A, @Rr	60 61	1/1	Add RAM data, addressed by Rr, to A	(A)←(A) + ((R0)) (A)←(A) + ((R1))		1
	ADD A, #data	03 data	2/2	Add immediate data to A	(A)←(A) + data		1
	ADDC A, Rr	7*	1/1	Add carry and register contents to A	(A)←(A) + (Rr) + (C)	r = 0–7	1
	ADDC A, @Rr	70 71	1/1	Add carry and RAM data, addressed by Rr, to A	(A)←(A) + ((R0)) + (C) (A)←(A) + ((R1)) + (C)		1
	ADDC A, #data	13 data	2/2	Add carry and immediate data to A	(A)←(A) + data + (C)		1
	ANL A, Rr	5*	1/1	'AND' Rr with A	(A)←(A) AND (Rr)	r = 0–7	
	ANL A, @Rr	50 51	1/1	'AND' RAM data, addressed by Rr, with A	(A)←(A) AND ((R0)) (A)←(A) AND ((R1))		
	ANL A, #data	53 data	2/2	'AND' immediate data with A	(A)←(A) AND data		
	ORL A, Rr	4*	1/1	'OR' Rr with A	(A)←(A) OR (Rr)	r = 0–7	
	ORL A, @Rr	40 41	1/1	'OR' RAM data, addressed by Rr, with A	(A)←(A) OR ((R0)) (A)←(A) OR ((R1))		
	ORL A, #data	43 data	2/2	'OR' immediate data with A	(A)←(A) OR data		
	XRL A, Rr	D*	1/1	'XOR' Rr with A	(A)←(A) XOR (Rr)	r = 0–7	
	XRL A, @Rr	D0 D1	1/1	'XOR' RAM, addressed by Rr, with A	(A)←(A) XOR ((R0)) (A)←(A) XOR ((R1))		
	XRL A, #data	D3 data	2/2	'XOR' immediate data with A	(A)←(A) XOR data		
	INC A	17	1/1	increment A by 1	(A)←(A) + 1		
	DEC A	07	1/1	decrement A by 1	(A)←(A) −1		
	CLR A	27	1/1	clear A to zero	(A)←0		
	CPL A	37	1/1	one's complement A	(A)←NOT(A)		
	RL A	E7	1/1	rotate A left	$(A_{n+1})\leftarrow(A_n)$ $(A_0)\leftarrow(A_7)$	n = 0–6	

ACCUMULATOR (cont.)	RLC A	F7	1/1	rotate A left through carry	$(A_{n+1})\leftarrow A_n$ $(A_0)\leftarrow(C)$, $(C)\leftarrow(A_7)$	n = 0–6	2
	RR A	77	1/1	rotate A right	$(A_n)\leftarrow(A_{n+1})$ $(A_7)\leftarrow(A_0)$	n = 0–6	
	RRC A	67	1/1	rotate A right through carry	$(A_n)\leftarrow(A_{n+1})$ $(A_7)\leftarrow(C)$, $(C)\leftarrow(A_0)$	n = 0–6	2
	DA A	57	1/1	decimal adjust A			2
	SWAP A	47	1/1	swap nibbles of A	$(A_{4-7})\leftrightarrow(A_{0-3})$		
DATA MOVES	MOV A, Rr	F*	1/1	move register contents to A	(A)←(Rr)	r = 0–7	
	MOV A, @Rr	F0 F1	1/1	move RAM data, addressed by Rr, to A	(A)←((R0)) (A)←((R1))		
	MOV A, #data	23 data	2/2	move immediate data to A	(A)←data		
	MOV Rr, A	A*	1/1	move accumulator contents to register	(Rr)←(A)	r = 0–7	
	MOV @Rr, A	A0 A1	1/1	move accumulator contents to RAM location addressed by Rr	((R0))←(A) ((R1))←(A)		
	MOV Rr, #data	B* data	2/2	move immediate data to Rr	(Rr)←data		
	MOV @Rr, #data	B0 data B1 data	2/2	move immediate data to RAM location addressed by Rr	((R0))←data ((R1))←data		
	XCH A, Rr	2*	1/1	exchange accumulator contents with Rr	(A)↔(Rr)	r = 0–7	
	XCH A, @Rr	20 21	1/1	exchange accumulator contents with RAM data addressed by Rr	(A)↔((R0)) (A)↔((R1))		
	XCHD A, @Rr	30 31	1/1	exchange lower nibbles of A and RAM data addressed by Rr	$(A_{0-3})\leftrightarrow((R0_{0-3}))$ $(A_{0-3})\leftrightarrow((R1_{0-3}))$		
	MOV A, PSW	C7	1/1	move PSW contents to accumulator	(A)←(PSW)		
	MOV PSW, A	D7	1/1	move accumulator bit 3 to PSW_3	$(PSW_3)\leftarrow(A_3)$		3
	MOVP A, @A	A3	1/2	move indirectly addressed data in current page to A	$(PC_{0-7})\leftarrow(A)$, $(A)\leftarrow((PC))$		
FLAGS	CLR C	97	1/1	clear carry bit	(C)←0		2
	CPL C	A7	1/1	complement carry bit	(C)←NOT(C)		2

	mnemonic	opcode (hex.)	bytes/ cycles	description	function	notes
REGISTER	INC Rr	1*	1/1	increment register by 1	(Rr)←(Rr) + 1 r = 0–7	
	INC @Rr	10 11	1/1	increment RAM data, addressed by Rr, by 1	((R0))←((R0)) + 1 ((R1))←((R1)) + 1	
	DEC Rr	C*	1/1	decrement register by 1	(Rr)←(Rr) – 1 r = 0–7	
	DEC @Rr	C0 C1	1/1	decrement RAM data, addressed by Rr, by 1	((R0))←((R0)) – 1 ((R1))←((R1)) – 1	
BRANCH	JMP addr	● 4 address	2/2	unconditional jump within a 2 K bank	(PC_{8-10})←$addr_{8-10}$ (PC_{0-7})←$addr_{0-7}$ (PC_{11-12})←MBFF 0–1	
	JMPP @A	B3	1/2	indirect jump within a page	(PC_{0-7})←((A))	
	DJNZ Rr, addr	E* address	2/2	decrement Rr by 1 and jump if not zero to addr	(Rr)←(Rr) – 1 r = 0–7 if (Rr) not zero (PC_{0-7})←addr	
	DJNZ @Rr, addr	E0	2/2	decrement RAM data, addressed by Rr by 1 and jump if not zero to addr	((R0))←((R0)) – 1 if ((R0)) not zero (PC_{0-7})←addr	
		E1			((R1))←((R1)) – 1 if ((R1)) not zero (PC_{0-7})←addr	
	JBb addr	▲ 2 address	2/2	jump to addr if Acc. bit b = 1	if b = 1 : (PC_{0-7})←addr b = 0–7	
	JC addr	F6 address	2/2	jump to addr if C = 1	if C = 1 : (PC_{0-7})←addr	
	JNC addr	E6 address	2/2	jump to addr if C = 0	if C = 0 : (PC_{0-7})←addr	
	JZ addr	C6 address	2/2	jump to addr if A = 0	if A = 0 : (PC_{0-7})←addr	
	JNZ addr	96 address	2/2	jump to addr if A is NOT zero	if A≠0 : (PC_{0-7})←addr	
	JT0 addr	36 address	2/2	jump to addr if T0 = 0	if T0 = 0: (PC_{0-7})←addr	
	JNT0 addr	26 address	2/2	jump to addr if T0 = 1	if T0 = 1: (PC_{0-7})←addr	
	JT1 addr	56 address	2/2	jump to addr if T1 = 1	if T1 = 1: (PC_{0-7})←addr	
	JNT1 addr	46 address	2/2	jump to addr if T1 = 0	if T1 = 0: (PC_{0-7})←addr	
	JTF addr	16 address	2/2	jump to addr if Timer Flag = 1	if TF = 1: (PC_{0-7})←addr	4
	JNTF addr	06 address	2/2	jump to addr if Timer Flag = 0	if TF = 0: (PC_{0-7})←addr	

TIMER/EVENT COUNTER	MOV A, T	42	1/1	move timer/event counter contents to accumulator	(A)←(T)	
	MOV T, A	62	1/1	move accumulator contents to timer/event counter	(T)←(A)	
	STRT CNT	45	1/1	start event counter		
	STRT T	55	1/1	start timer		
	STOP TCNT	65	1/1	stop timer/event counter		
	EN TCNTI	25	1/1	enable timer/event counter interrupt		
	DIS TCNTI	35	1/1	disable timer/event counter interrupt		
CONTROL	EN I	05	1/1	enable external interrupt		
	DIS I	15	1/1	disable external interrupt		
	SEL RB0	C5	1/1	select register bank 0	(RBS)←0	5
	SEL RB1	D5	1/1	select register bank 1	(RBS)←1	5
	SEL MB0	E5	1/1	select program memory bank 0	(MBFF0)←0, (MBFF1)←0	
	SEL MB1	F5	1/1	select program memory bank 1	(MBFF0)←1, (MBFF1)←0	
	SEL MB2	A5	1/1	select program memory bank 2	(MBFF0)←0, (MBFF1)←1	
	SEL MB3	B5	1/1	select program memory bank 3	(MBFF0)←1, (MBFF1)←1	
	STOP	22	1/1	enter STOP mode		
	IDLE	01	1/1	enter IDLE mode		
SUBROUTINE	CALL addr	▲ 4 address	2/2	jump to subroutine	((SP))←(PC), ($PSW_{4,6,7}$) 6 (SP)←(SP) + 1 (PC_{8-10})←$addr_{8-10}$ (PC_{0-7})←$addr_{0-7}$ (PC_{11-12})←MBFF 0–1	6
	RET	83	1/2	return from subroutine	(SP)←(SP) – 1 (PC)←((SP))	6
	RETR	93	1/2	return from interrupt and restore bits 4, 6, 7 of PSW	(SP)←(SP) – 1 ($PSW_{4,6,7}$) + (PC)←((SP))	6

	mnemonic	opcode (hex.)	bytes/ cycles	description	function	notes
PARALLEL INPUT/OUTPUT	IN A, Pp	08 09 0A	1/2	input port p data to accumulator	(A)←(P0) (A)←(P1) (A)←(P2)	 7
	OUTL Pp, A	38 39 3A	1/2	output accumulator data to port p	(P0)←(A) (P1)←(A) (P2)←(A)	
	ANL Pp, #data	98 99 9A	2/2	AND port p data with immediate data	(P0)←(P0) AND data (P1)←(P1) AND data (P2)←(P2) AND data	
	ORL Pp, #data	88 89 8A	2/2	OR port p data with immediate data	(P0)←(P0) OR data (P1)←(P1) OR data (P2)←(P2) OR data	
SERIAL INPUT/OUTPUT	MOV A, S_n	0C 0D	1/2	move serial I/O register contents to accumulator	(A)←(S0) (A)←(S1)	 8
	MOV S_n, A	3C 3D 3E	1/2	move accumulator contents to serial I/O register	(S0)←(A) (S1)←(A) (S2)←(A)	 9
	MOV S_n, #data	9C 9D 9E	2/2	move immediate data to serial I/O register	(S0)←data (S1)←data (S2)←data	
	EN SI	85	1/1	enable serial I/O interrupt		
	DIS SI	95	1/1	disable serial I/O interrupt		
	NOP	00	1/1	no operation		

Notes to Table 8

1. PSW CY, AC affected
2. PSW CY affected
3. PSW PS affected
4. Execution of JTF and JNTF instructions resets the Timer Flag (TF).

* : 8, 9, A, B, C, D, E, F
● : 0, 2, 4, 6, 8, A, C, E
▲ : 1, 3, 5, 7, 9, B, D, F

5. PSW RBS affected
6. PSW SP_0, SP_1, SP_2 affected
7. (A) = 1111 P23, P22, P21, P20.
8. (S1) has a different meaning for read and write operation, see serial I/O interface.
9. (S2) is a write only register. Reading S2 will give value FFH.

RATINGS

Limiting values in accordance with the Absolute Maximum System (IEC 134)

Supply voltage (pin 28)	V_{DD}		−0,8 to + 8	V
All input voltages	V_I		0,8 to V_{DD} + 0,8	V
D.C. current into any input or output	$\pm I_I, \pm I_O$	max.	10	mA
Total power dissipation (see note)	P_{tot}	max.	500	mW
Power dissipation per output				
except P23, SCLK	P_O	max.	50	mW
P23, SCLK	P_O	max.	180	mW
Storage temperature range	T_{stg}		−65 to + 150	°C
Operating ambient temperature range	T_{amb}		−25 to + 70	°C
Operating junction temperature	T_j	max.	125	°C

Note

Thermal resistance (junction to ambient)				
for SOT-117	$R_{th\ j\text{-}a}$	max.	120	K/W ←
for SOT-135A	$R_{th\ j\text{-}a}$	max.	60	K/W
for SOT-136A	$R_{th\ j\text{-}a}$	max.	150	K/W

DEVELOPMENT DATA

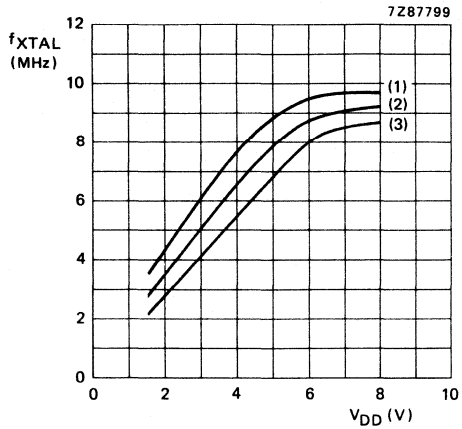
D.C. CHARACTERISTICS

V_{DD} = 2,75 to 6 V; V_{SS} = 0 V; T_{amb} = −25 to +70 °C; all voltages with respect to V_{SS}; f = 3,58 MHz with R_S = 50 Ω; unless otherwise specified.

parameter	symbol	min.	typ.	max.	unit
Supply voltage					
operating (see Fig. 23)	V_{DD}	1,8	–	6	V
STOP mode for RAM retention	V_{DD}	1,0	–	6	V
Supply current					
operating					
at V_{DD} = 3 V (see Fig. 24)	I_{DD}	–	600	–	μA
IDLE mode					
at V_{DD} = 3 V (see Fig. 25)	I_{DD}	–	300	–	μA
STOP mode (see Fig. 26 and note 1)					
at V_{DD} = 1,8 V; T_{amb} = 25 °C	I_{DD}	–	1,2	2,5	μA
at V_{DD} = 1,8 V; T_{amb} = 55 °C	I_{DD}	–	–	5	μA
at V_{DD} = 1,8 V; T_{amb} = 70 °C	I_{DD}	–	–	10	μA
RESET I/O					
Switching level	V_{RESET}	–	1,3	–	V
Sink current					
at $V_{DD} > V_{RESET}$	I_{OL}	–	7	–	μA
Inputs					
Input voltage LOW	V_{IL}	0	–	$0{,}3V_{DD}$	V
Input voltage HIGH	V_{IH}	$0{,}7V_{DD}$	–	V_{DD}	V
Input leakage current					
at $V_{SS} < V_I < V_{DD}$	$\pm I_{IL}$	–	–	1	μA
Outputs					
Output voltage LOW					
at $V_I = V_{SS}$ or V_{DD}; $\lvert I_O \rvert < 1$ μA	V_{OL}	–	–	0,05	V
Output sink current LOW					
at V_{DD} = 3 V; V_O = 0,4 V					
except P23/SDA, SCLK (see Fig. 27)	I_{OL}	0,75	1,5	–	mA
P23/SDA, SCLK (see Fig. 28)	I_{OL}	1,5	–	–	mA
Pull-up output source current HIGH (see Fig. 29)					
at V_{DD} = 3 V; $V_O = 0{,}9V_{DD}$	$-I_{OH}$	25	–	–	μA
at V_{DD} = 3 V; $V_O = V_{SS}$	$-I_{OH}$	–	–	200	μA
Push-pull output source current HIGH					
at V_{DD} = 3 V; $V_O = V_{DD}$−0,4 V	$-I_{OH}$	0,75	1,5	–	mA

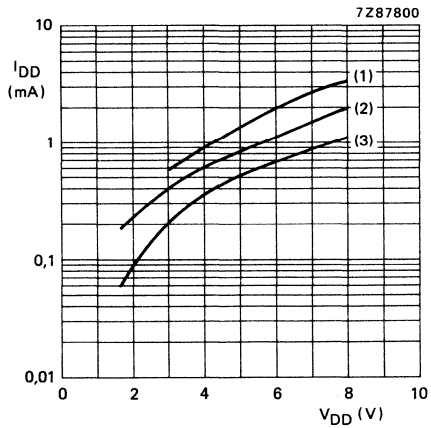
Note 1

Crystal connected between XTAL 1 and XTAL 2; SCL and SDA pulled to V_{DD} via 5,6 kΩ resistor; CE and T1 at V_{SS}.



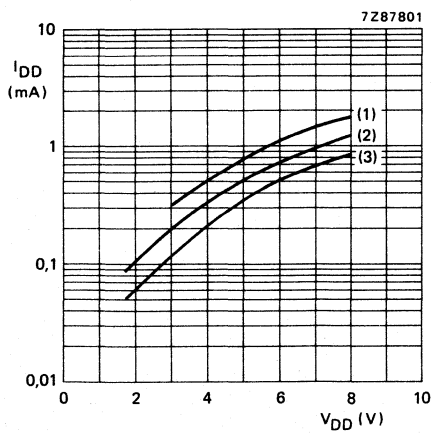
(1) $T_{amb} = -25\ ^{o}C$
(2) $T_{amb} = 25\ ^{o}C$
(3) $T_{amb} = 70\ ^{o}C$

Fig. 23 Maximum clock frequency (f_{XTAL}) as a function of the supply voltage (V_{DD}).



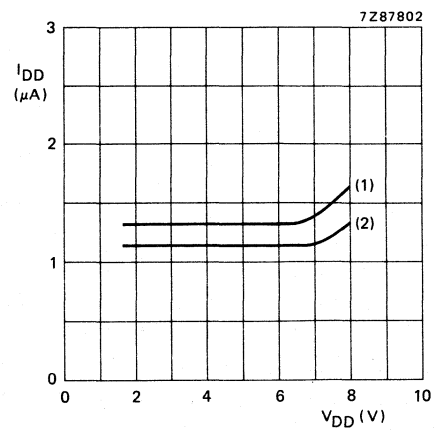
(1) clock frequency = 4 MHz
(2) clock frequency = 2 MHz
(3) clock frequency = 500 kHz

Fig. 24 Typical supply current (I_{DD}) in operating mode as a function of the supply voltage (V_{DD}); $T_{amb} = 25\ ^{o}C$.



(1) clock frequency = 4 MHz
(2) clock frequency = 2 MHz
(3) clock frequency = 500 kHz

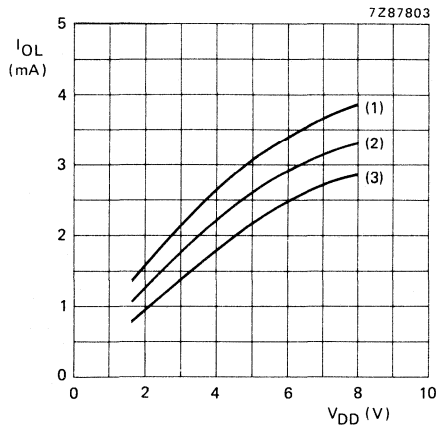
Fig. 25 Typical supply current (I_{DD}) in IDLE mode as a function of the supply voltage (V_{DD}); $T_{amb} = 25\ ^{o}C$.



(1) $T_{amb} = 70\ ^{o}C$
(2) $T_{amb} = 25\ ^{o}C$

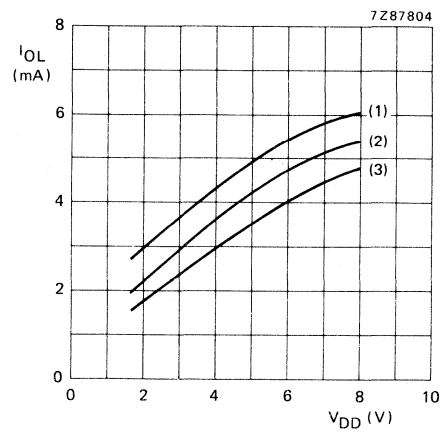
Fig. 26 Typical supply current (I_{DD}) in STOP mode as a function of the supply voltage (V_{DD}).

DEVELOPMENT DATA



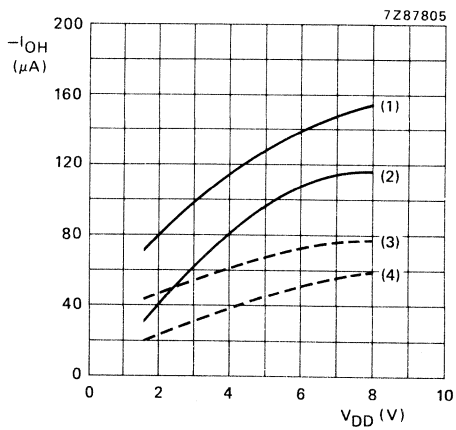
(1) $T_{amb} = -25\ ^{o}C$
(2) $T_{amb} = 25\ ^{o}C$
(3) $T_{amb} = 70\ ^{o}C$

Fig. 27 Output sink current LOW (I_{OL}), except outputs P23/SDA and SCLK, as a function of supply voltage (V_{DD}); V_O = 0,4 V.



(1) $T_{amb} = -25\ ^{o}C$
(2) $T_{amb} = +25\ ^{o}C$
(3) $T_{amb} = +70\ ^{o}C$

Fig. 28 Output current LOW (I_{OL}), outputs P23/SDA and SCLK, as a function of supply voltage (V_{DD}); V_O = 0,4 V.



(1) $T_{amb} = 25\ ^{o}C$; $V_O = V_{SS}$
(2) $T_{amb} = 25\ ^{o}C$; $V_O = 0{,}9V_{DD}$
(3) $T_{amb} = 70\ ^{o}C$; $V_O = V_{SS}$
(4) $T_{amb} = 70\ ^{o}C$; $V_O = 0{,}9V_{DD}$

Fig. 29 Output source current HIGH ($-I_{OH}$) as a function of supply voltage (V_{DD}).

A.C. CHARACTERISTICS

Rise and fall times between 10 and 90% levels; C_L = 50 pF

parameter	symbol	at 70 °C max. value			unit
	V_{DD}	1,8	3,0	6,0	V
Fall time	t_f	200	100	70	ns
Rise time	t_r	200	100	80	ns

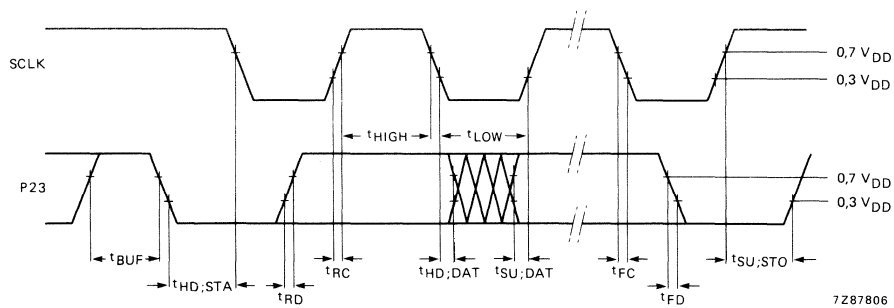


Fig. 30 PCD3343 timing requirements for the P23 and SCLK *input* signals.

Table 9 Input timing shown in figure 30

symbol	timing
t_{BUF}	$\geqslant 14t_{XTAL}$
$t_{HD;STA}$	$\geqslant 14t_{XTAL}$
t_{HIGH}	$\geqslant 17t_{XTAL}$
t_{LOW}	$\geqslant 17t_{XTAL}$
$t_{SY;STO}$	$\geqslant 14t_{XTAL}$
$t_{HD;DAT}$	> 0
$t_{SU;DAT}$	$\geqslant 250$ ns
t_{RD}	$\leqslant 1\ \mu s$
t_{RC}	$\leqslant 1\ \mu s$
t_{FD}	$\leqslant 1\ \mu s$
t_{FC}	$\leqslant 0{,}3\ \mu s$

Notes to Table 9

t_{XTAL} = one period of the XTAL input frequency (f_{XTAL})
= 280 ns for f_{XTAL} = 3,58 MHz.
These figures apply to all modes.

DEVELOPMENT DATA

A.C. CHARACTERISTICS (continued)

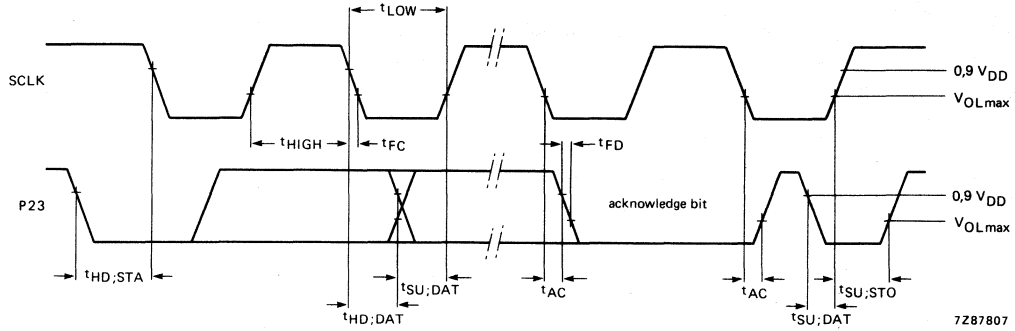


Fig. 31 PCD3343 timing requirements for the P23 and SCLK *output* signals.

Table 10 Output timing shown in figure 31

symbol	timing	
	normal mode (ASC in S2 = 0)	low-speed mode (ASC in S2 = 1)
$t_{HD;STA}$	½ (DF + 9) t_{XTAL}	¾ (DF + 9) t_{XTAL}
t_{HIGH}	½ (DF) t_{XTAL}	¾ (DF) t_{XTAL}
t_{LOW}	½ (DF) t_{XTAL}	¼ (DF) t_{XTAL}
$t_{SU;STO}$	½ (DF − 3) t_{XTAL}	¼ (DF − 3) t_{XTAL}
$t_{HD;DAT}$ (slave transmitter)		
any DF	$\geqslant 9t_{XTAL}$ $\leqslant 12t_{XTAL}$	$\geqslant 9t_{XTAL}$ $\leqslant 12t_{XTAL}$
$t_{HD;DAT}$ (master transmitter)		
for DF $\leqslant$ 51	$\geqslant 9t_{XTAL}$ $\leqslant 12t_{XTAL}$	– –
for DF $\leqslant$ 99	– –	$\geqslant 9t_{XTAL}$ $\leqslant 12t_{XTAL}$
$t_{SU;DAT}$ (master transmitter)		
for DF > 51	$\geqslant 15t_{XTAL}$ $\leqslant 24t_{XTAL}$	– –
for DF > 99	–	$\geqslant 15t_{XTAL}$ $\leqslant 24t_{XTAL}$
for DF $\leqslant$ 51	$\geqslant 9t_{XTAL}$	$\geqslant 9t_{XTAL}$
for DF $\leqslant$ 99	–	$\geqslant 9t_{XTAL}$
t_{AC}	$\geqslant 9t_{XTAL}$ $\leqslant 12t_{XTAL}$	$\geqslant 9t_{XTAL}$ $\leqslant 12t_{XTAL}$
t_{FD}, t_{FC}	$\leqslant$ 100 ns at C_b = 400 pF	$\leqslant$ 100 ns at C_b = 400 pF

Notes to Table 10

t_{XTAL} = one period of the XTAL input frequency (f_{XTAL})
= 280 ns for f_{XTAL} = 3,58 MHz.
DF = divisor (see Table 2 Serial I/O section).
C_b = the maximum bus capacitance for each line.

DEVELOPMENT DATA

APPLICATION INFORMATION

A block diagram of an electronic featurephone built around the PCD3343 is shown in figure 32. It comprises the following dedicated telephony IC's:

- TEA1060/1061 — transmission circuit for telephony
- PCD3312 — DTMF generator with Serial I/O
- PCE2111 or PCF8577 — 2 LCD drivers in LCD module MB7020160
- PCD8571 — 1 K RAM's with Serial I/O; the number of RAM's depends on the required amount of stored telephone numbers
- PCD3360/3361 — programmable multi-tone ringer

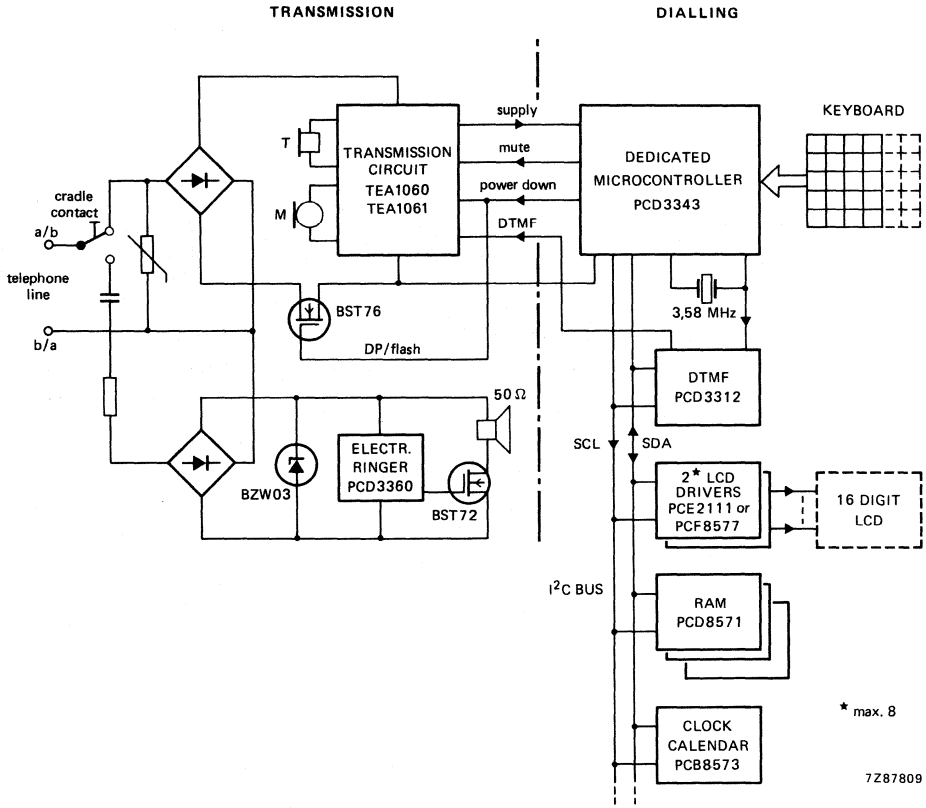


Fig. 32 Block diagram of electronic featurephone with common line interface.

A detailed application diagram of the PCD3343 with PCD3312 (DTMF), two PCD8571 (RAM) and two PCE2111 (LCD display drivers) is shown in figure 33.

Row 5 of the keyboard contains the following special keys:

- P — program and autodial
- FL — flash or register recall
- R — redial or extended redial
- AP — access pause

Row 6 contains the different diode options.

Columns 5 and 6 contain the button keys M0 to M9; single name keys for repertory telephone numbers.

APPLICATION INFORMATION (continued)

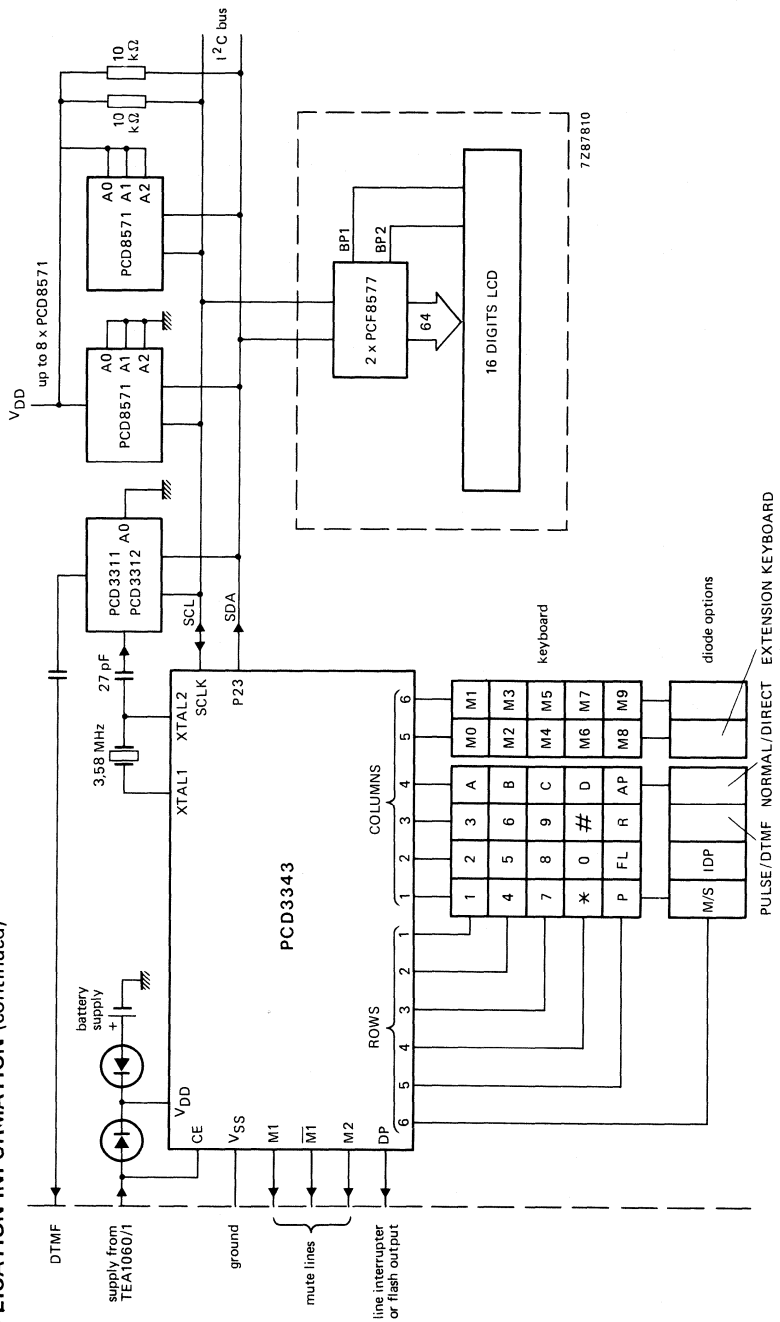


Fig. 33 Application diagram of PCD3343 for electronic featurephone with associated keyboard.

Additional information is available on request for the following:

- Serial I/O
- I^2C bus specification
- Interrupt logic
- Instruction set descriptions
- Software routines for an intelligent telephone set

SINGLE-CHIP 8-BIT MICROCONTROLLER

DESCRIPTION

The PCF84C12 microcontroller is manufactured in CMOS technology.
It has 13 quasi-bidirectional I/O port lines, one single-level vectored interrupt, an 8-bit timer event counter and on-board clock oscillator and clock circuits.

This microcontroller is an efficient controller as well as an arithmetic processor. The instruction set is based on that of the MAB8048 and is pin- and instruction set compatible with the MAB8400 family. The PCF84C12 has extensive bit handling abilities and facilities for both binary and BCD arithmetic.

For detailed information see the user manual "Single-chip 8-bit microcontrollers".

Features

- 8-bit CPU, ROM, RAM, I/O in a single 20-lead DIL or SO package
- 1 K ROM bytes
- 64 RAM bytes
- 13 quasi-bidirectional I/O port lines
- Two test inputs: one of which is also the external interrupt input
- Single-level vectored interrupts: external and timer/event counter
- 8-bit programmable timer/event counter
- Clock frequency 100 kHz to 10 MHz
- Over 80 instructions (based on MAB8048) all of 1 or 2 cycles
- Single supply voltage from 2,5 V to 5,5 V
- STOP and IDLE mode
- Power-on-reset circuit
- Operating temperature range: −40 to + 85 °C

PACKAGE OUTLINES

PCF84C12P: 20-lead DIL; plastic (SOT-146).
PCF84C12T: 20-lead mini-pack; plastic (SO-20; SOT-163A).

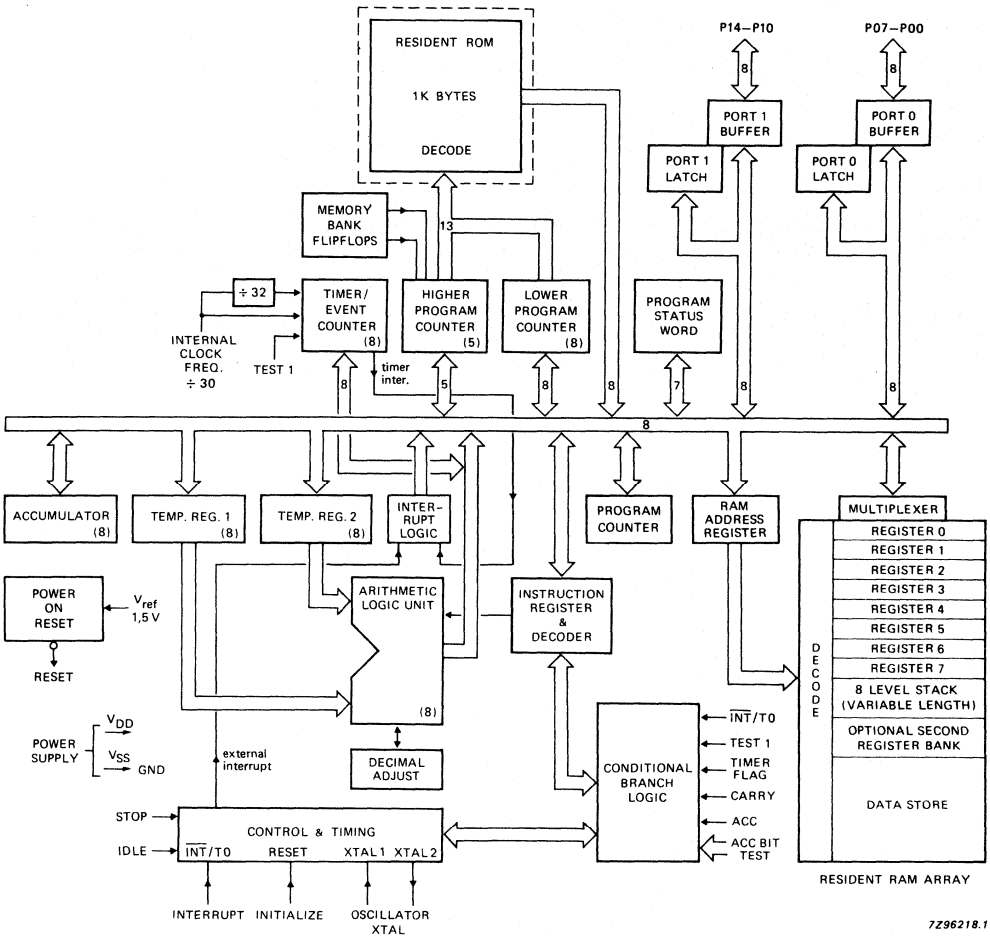


Fig. 1 Block diagram.

PINNING

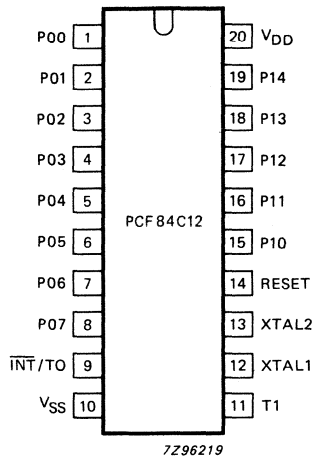


Fig. 2 Pinning diagram.

DEVELOPMENT DATA

PIN DESIGNATION

1-8	P00-P07	**Port 0:** 8-bit quasi-bidirectional I/O port.
9	$\overline{INT}$/T0	**Interrupt/Test 0:** external interrupt input (sensitive to negative-going edge)/test input pin; when used as a test input directly tested by conditional branch instructions JT0 and JNT0.
10	V_{SS}	**Ground:** circuit earth potentional.
11	T1	**Test 1:** test input pin, directly tested by conditional branch instructions JT1 and JNT1. T1 also functions as an input to the 8-bit timer/event counter, using the STRT CNT instruction.
12	XTAL 1	**Oscillator input:** crystal which determines the internal oscillator frequency or the external clock generator.
13	XTAL 2	**Oscillator output**
14	RESET	**Reset input:** used to initialize the processor (active HIGH), or output of power-on-reset circuit.
15-19	P10-P14	**Port 1:** quasi-bidirectional port in parallel port mode.
20	V_{DD}	**Power supply:** 2,5 V to 5,5 V.

FUNCTIONAL DESCRIPTION

Program memory

The program memory consists of 1 K bytes, in a read-only memory (ROM). Each location is directly addressable by the program counter. The memory is mask-programmed at the factory. Figure 3 shows the program memory map.

Three program memory locations are of special importance:

- Location 0; contains the first instruction to be executed after the processor is initialized (RESET),
- Location 3; contains the first byte of an external interrupt service subroutine,
- Location 7; contains the first byte of a timer/event counter interrupt service subroutine.

Data memory

Data memory consists of 64 bytes, random-access data memory (RAM). All locations are indirectly addressable using RAM pointer registers; up to 16 designated locations are directly addressable. Memory also includes an 8-level program counter stack addressed by a 3-bit stack pointer. Figure 4 shows the data memory map.

Working registers

Locations 0 to 7 are designated as working registers, directly addressable by the direct register instructions. Ease of addressing, and a minimum requirement of instruction bytes to manipulate their contents, makes these locations suitable for storing frequently addressed intermediate results. This bank of registers can be selected by the SEL RB0 instruction.

Executing the select register bank instruction SEL RB1, designates locations 24 to 31 as working registers, instead of locations 0 to 7, and these are then directly addressable. This second bank of working registers may be used as an extension of the first or reserved for use during interrupt service subroutines saving the first bank for use in the main program. If the second bank is not used, locations 24 to 31 may serve as general purpose RAM.

The first locations of each bank contain the RAM pointer registers R0, R1, R0' and R1', which indirectly address all RAM locations.

All RAM locations make efficient program loop counters when used with the decrement register and test instruction DJNZ.

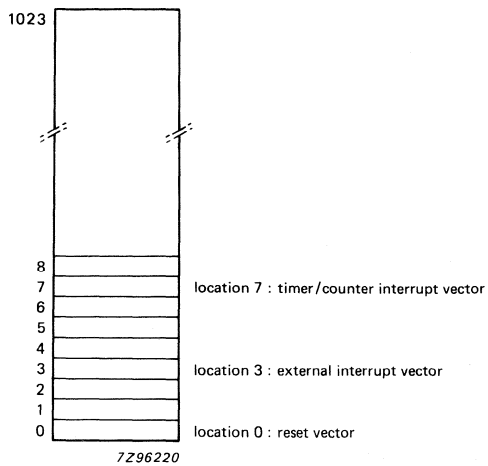


Fig. 3 Program memory map.

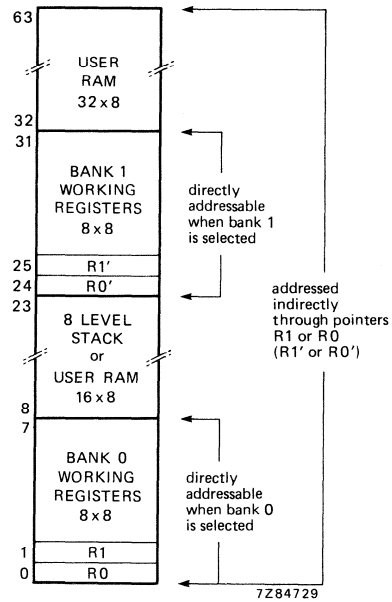


Fig. 4 Data memory map.

DEVELOPMENT DATA

Program counter stack

Locations 8 to 23 may be designated as an 8-level program counter stack (2 locations per level), or as general purpose RAM. The program counter stack (Fig. 5) enables the processor to keep track of the return addresses and status generated by interrupts or CALL instructions by storing the contents of the program counter prior to servicing the subroutine. A 3-bit stack pointer determines which of the eight register pairs of the program counter stack will be loaded with next generated return address.

The stack pointer, when initialized to 000 by RESET, points to RAM locations 8 and 9. On the first subroutine CALL or interrupt, the contents of the program counter and bits 4, 6 and 7 of the program status word (PSW) are transferred to locations 8 and 9. The stack pointer increments by one and points to locations 10 and 11 ready for another CALL. Because an address may be up to 13 bits long, two bytes must be used to store each address.

At the end of a subroutine, which is signalled by a return instruction (RET or RETR), the stack pointer decrements by one and the contents of the register pair on top of the stack are transferred to the program counter. The saved PSW bits are transferred to the PSW only by the RETR instruction.

If not all 8 levels of subroutine and interrupt nesting are used, the unused portion of the stack may be used as any other indirectly addressable RAM locations.

Nesting of subroutines within subroutines can continue up to 8 times without overflowing the stack. If overflow does occur the deepest address stored (locations 8 and 9) will be overwritten and lost since the stack pointer overflows from 111 to 000. It also underflows from 000 to 111.

FUNCTIONAL DESCRIPTION (continued)

The value of the saved contents of the program counter is different for an interrupt CALL compared to a normal CALL to subroutine. With an interrupt CALL, the program counter return address is saved; with a subroutine CALL, the saved program counter value is one less than the program counter return address.

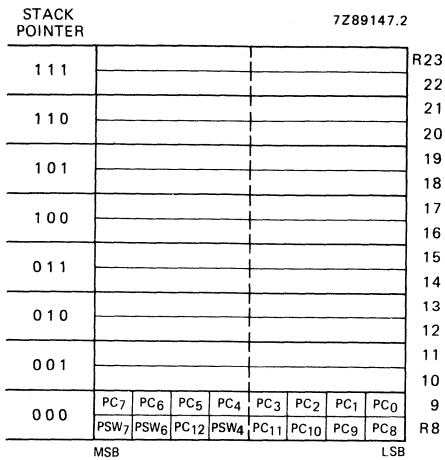


Fig. 5 Program counter stack.

IDLE and STOP modes

IDLE mode

When the microcontroller enters the IDLE mode via the IDLE instruction (H'01') the oscillator and timer/counter are kept running. The microcontroller exits from the IDLE mode by one of three interrupts if they are enabled or by activating a RESET. If the interrupt is not enabled the processor will remain in the IDLE mode. An active signal on the RESET pin restarts the microcontroller and a normal RESET sequence is executed (see Fig. 6).

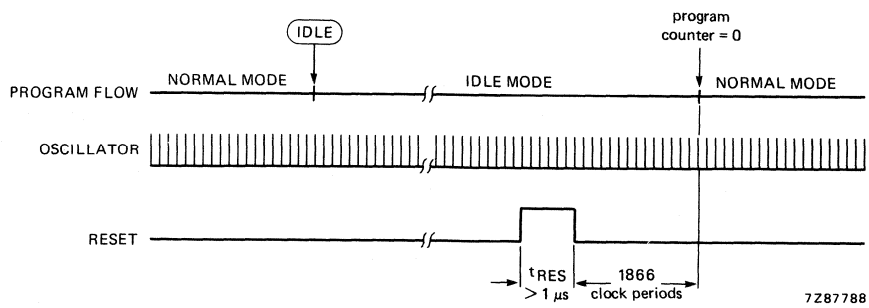


Fig. 6 Exit from IDLE mode via a RESET.

An active signal coming from an enabled interrupt causes the execution of the normal interrupt routine since normal interrupt scanning is still being carried out. A HIGH-to-LOW transition on the external interrupt pin ($\overline{INT}$/T0) reactivates the microcontroller. A LOW level applied to $\overline{INT}$/T0 will reactivate the microcontroller only in the STOP mode. Thus, if $\overline{INT}$/T0 was LOW before the microcontroller entered the IDLE mode, it must go HIGH before the microcontroller can be reactivated (see Fig. 7).

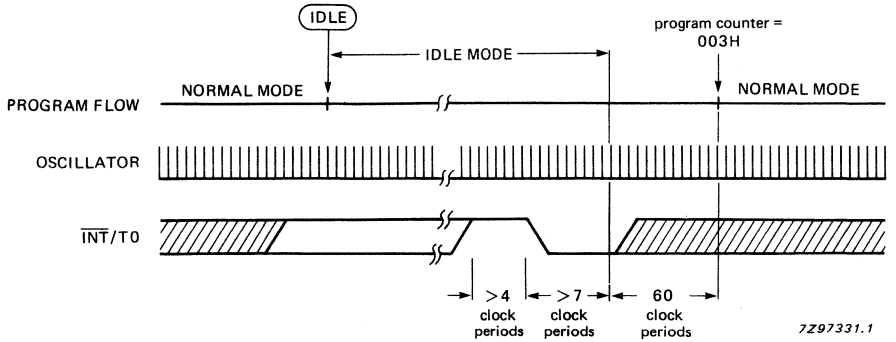


Fig. 7 Exit from IDLE mode via an interrupt.

Wake-up from the IDLE mode is ensured when $\overline{INT}$/T0 is HIGH for 4 CP (clock periods) followed by a LOW for 7 CP. After the initial forced CALL 003 H operation (60 CP) the program continuea with the external interrupt service routine.

STOP mode

The microcontroller enters the STOP mode by the STOP instruction (22 H). The oscillator is switched off. The internal status of the CPU, RAM contents and the state of I/O ports are not affected. The microcontroller can be brought-out of the STOP mode by an active signal at the external interrupt input or by an external RESET signal. When one of these two signals is applied an internal delay of 1866 CP is provided to ensure that all internal clocks are operating correctly before restart (see Fig. 8).

Note; the start-up time of a crystal oscillator is measured in milliseconds, and the 1866 CP count begins after this start-up time.

If the microcontroller exists from the STOP mode by activating RESET, a normal RESET sequence is executed.

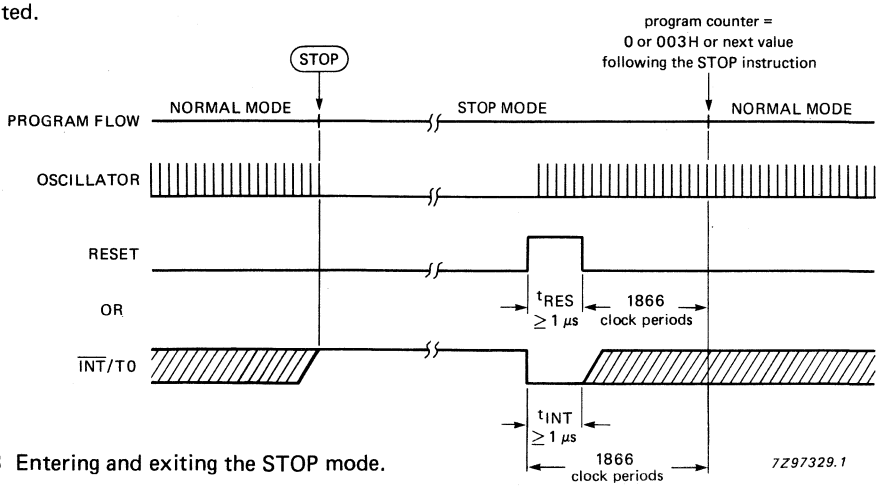


Fig. 8 Entering and exiting the STOP mode.

FUNCTIONAL DESCRIPTION (continued)

If the microcontroller exits the STOP mode by pulling the external interrupt input pin LOW, an interrupt sequence is executed only if the external interrupt is enabled. In this event the microcontroller resumes the normal program sequence after returning from the interrupt routine, as in the normal mode. If the interrupt is not enabled, it continues the normal program sequence, executing the instruction following the STOP instruction.

The microcontroller is restarted by a LOW level applied at the $\overline{INT}$/T0 pin, and not by a HIGH-to-LOW transition as in a normal interrupt mechanism.

When the $\overline{INT}$/T0 level is active during the STOP instruction then no STOP is executed.

A LOW level on the external interrupt input of at least 1 μs will cause the microcontroller to exit the STOP mode.

I/O facilities

The PCF84C12 family has 15 I/O lines arranged as:

- Port 0 parallel port of 8 lines (P00 to P07)
- Port 1 parallel port of 5 lines (P10 to P14)
- $\overline{INT}$/T0 external interrupt and test input. When used as a test input can be directly tested by conditional branch instructions JT0 and JNT0
- T1 test input which can alter program sequences when tested by conditional jump instructions JT1 and JNT1. T1 also functions as an input to the 8-bit timer/event counter.

Parallel ports

All parallel ports can be used as outputs or inputs, their structure is quasi-bidirectional.
Output data written to a port is latched and remains unchanged until rewritten.
Input data is not latched and so must be present until read by an input instruction.

Input lines are fully CMOS compatible, output lines can drive one TTL or CMOS load.

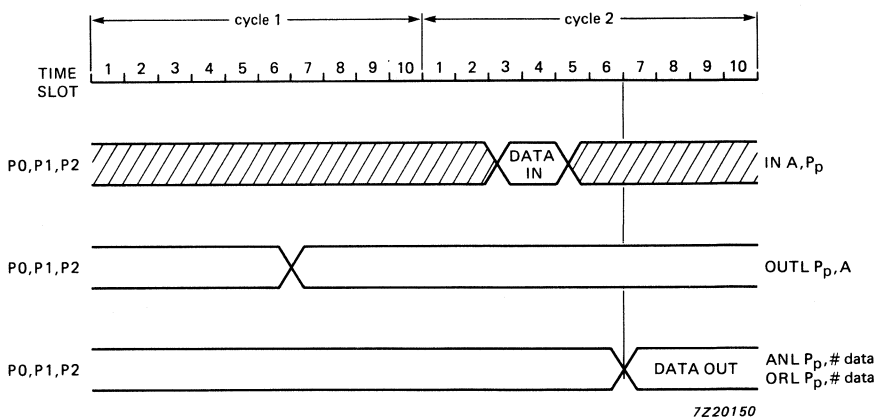


Fig. 9 Shows the timing diagram for all ports using IN, OUTL, ANL and ORL instructions. For the OUTL instruction data changes on time slot 7 of cycle 1. For the MOV, ANL and ORL instructions, the ports change on time slot 7 of cycle 2.

Fig. 10 shows the quasi-bidirectional I/O interface with push-pull output and switched pull-up current source.
Each line is pulled up to V_{DD} via a constant current source (TR4), which is enabled via TR3 whenever one of the two output latches contains a logic 1. This current provides sufficient source current for a TTL HIGH level, yet can be pulled LOW by an external CMOS device, thus allowing the same pin to be used for both input and output.

When a logic 1 is written to the line for the first time (MQ = 1, SQ = 0), TR2 is switched on for the duration of the internal write pulse (one oscillator period), to provide a fast transition from logic 0 to logic 1. Subsequent writing of a logic 1 to the port lines will not switch TR2 on. This prevents unnecessary current through external components connected to the port lines of the same port which might be in the input mode and also connected to ground.

When a logic 0 is written to the line, TR3 switches off the current source. Current sinking capability is provided by TR1, which is now switched on. When used as an input, a logic 1 must first be written to the line, otherwise TR1 will remain low impedance.

The PCF84C12 offers the possibility to select individually all the port pins by the following mask options:

Option 1 – STANDARD PORT; quasi-bidirectional I/O with switched pull-up current source of 100 μA (typ.) and P-channel booster transistor TR2.
TR2 is only active during 1 clock cycle (Fig. 10).

Option 2 – OPEN DRAIN; quasi-bidirectional I/O with only an N-channel open drain output. Application as an output requires connection of an external pull-up resistor (Fig. 11).

Option 3 – PUSH-PULL OUTPUT; drive capability of the output will be 1,6 mA (min.) at V_{DD} = 5 V in both polarities. To avoid a large current flowing through the output transistors during the input mode, these push-pull pins must only be used as outputs (Fig. 12).

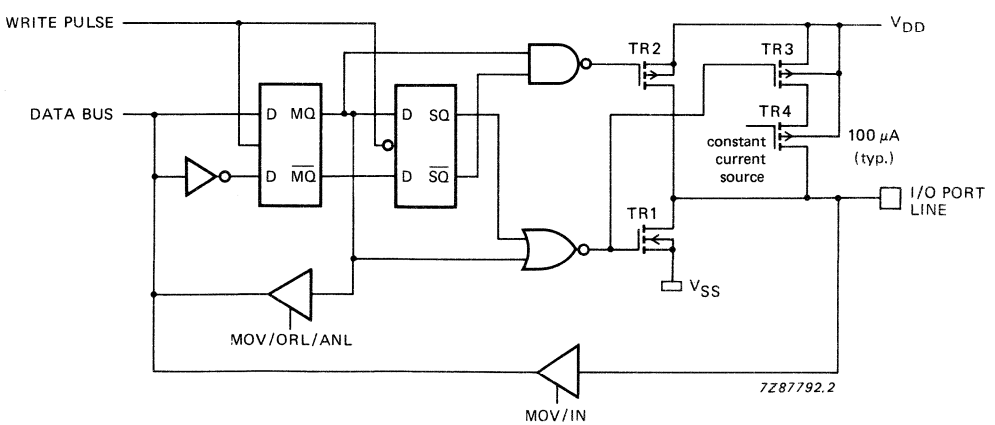


Fig. 10 Standard output with switched pull-up current source.

FUNCTIONAL DESCRIPTION (continued)

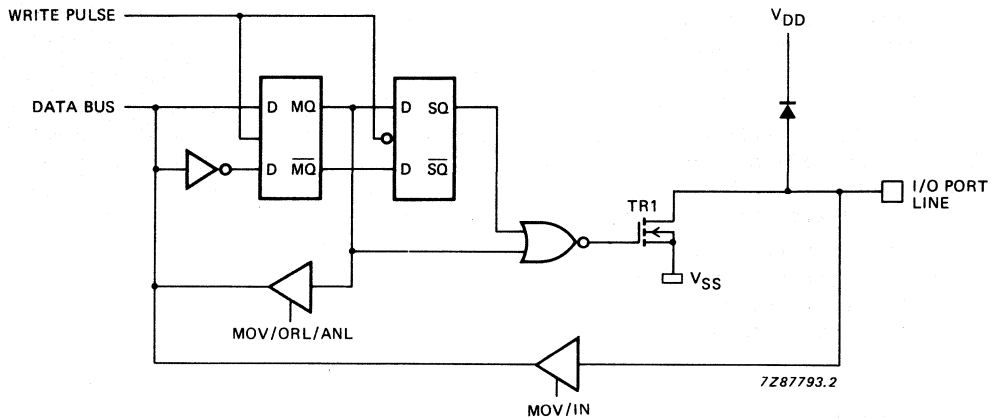


Fig. 11 Open drain output.

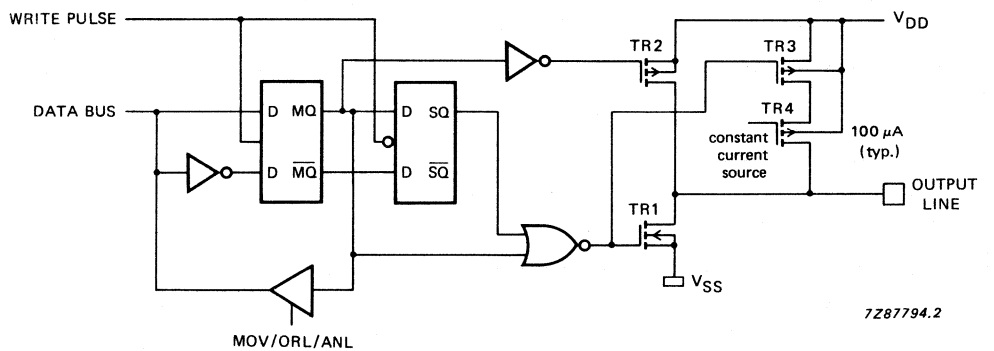


Fig. 12 Push-pull output.

Interrupts (see Fig. 13)

When the external interrupt is enabled, a HIGH-to-LOW transition on the $\overline{INT}$/T0 input initiates an external interrupt subroutine which causes a CALL to program memory location 3 following completion of the current instruction.
The interrupt must remain enabled until the interrupt instruction is completed, otherwise the next instruction of the main program will be executed. A timer/event counter overflow forces a CALL to location 7 when the timer interrupt is enabled.

When an interrupt subroutine starts, the contents of the program counter and bits 4, 6 and 7 of the PSW have been saved in the program counter stack. Accumulator contents have to be saved by software. Interrupt acknowledgement can be carried out by software via port pins. All interrupt subroutines must reside in memory bank 0.

Since the interrupt system is single level, once an interrupt is detected, all further interrupt requests are latched, but ignored, pending a RETR instruction to re-enable the interrupt input logic. After executing RETR, the program continues in the main part; this is independent of the occurance of a second interrupt during the running of the first routine. If 2 interrupts occur simultaneously, their priority is:

- (1) external
- (2) timer/event counter

An external interrupt can be generated by using the timer/counter in the event counter mode. The counter is first preset to (H'FF'), then EN TCNTI instruction is executed. A LOW-to-HIGH transition of the T1 input will then initiate an interrupt subroutine and cause a CALL to location 7.

On execution of a DIS I instruction, the PCF84C12 always clears the digital filter/latch and the External Interrupt Flag.

The Timer Flag (TF) is reset only when the JTF or JNTF instruction is executed or after RESET.

The Timer Interrupt Flag is set when timer overflow occurs, only if the timer interrupt is enabled.

The microcontroller will exit the IDLE mode when one of the following two interrupts is enabled.

- External
- Timer/event counter

There is no internal pull-up or pull-down device connected to the external interrupt input (pin 9). If required pin 9 must be externally connected to a resistor ($R \leqslant 100\ k\Omega$). When the external interrupt is not used pin 9 must be connected to V_{DD}.

Improvements to interrupt and timer logic with respect to the MAB8400 family

For detailed information see the user manual "Single-chip 8-bit microcontrollers", PCF84C00 section.

FUNCTIONAL DESCRIPTION (continued)

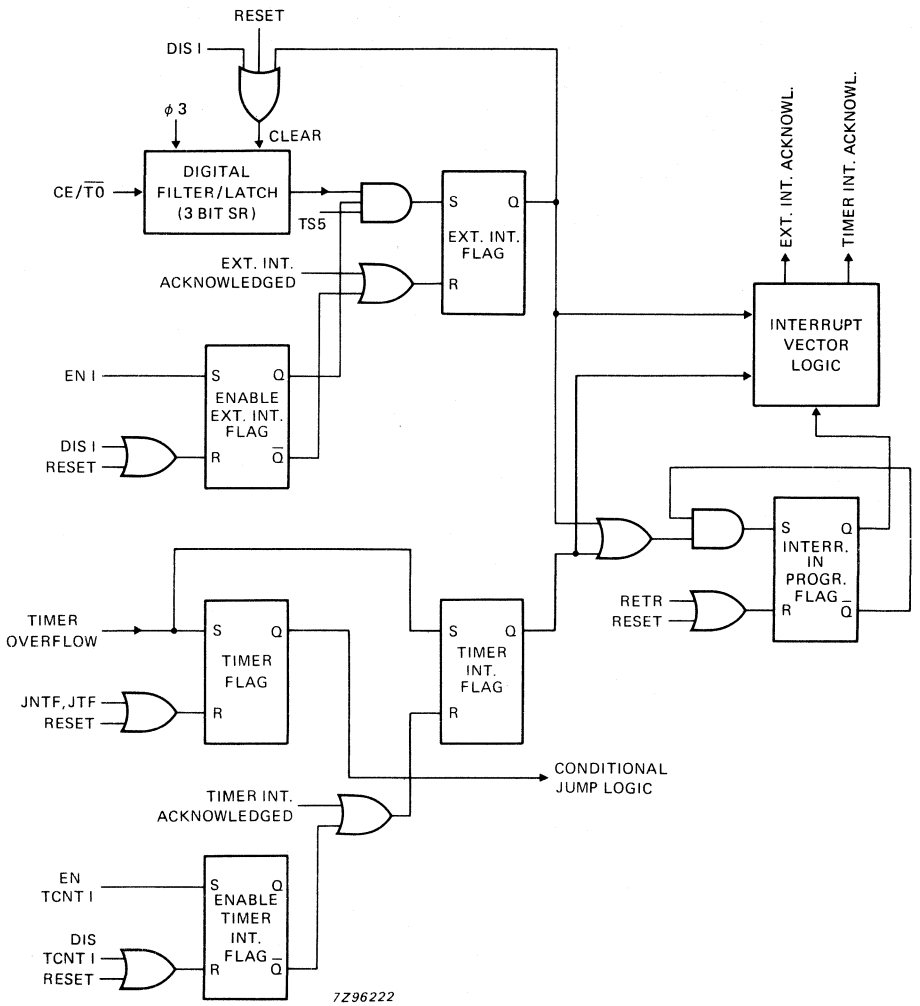


Fig. 13 Interrupt logic.

Notes to figure 13

1. $\overline{INT}$/T0 negative edge is always latched in the digital filter/latch.
2. Correct interrupt timing is ensured when $\overline{INT}$/T0 is HIGH for >4 CP followed by a LOW for >7 CP.
3. When the interrupt in progress flag is set, further interrupts are latched but ignored, until RETR is executed.
4. A DIS I instruction always clears a pending external interrupt.
5. For all flip-flops, RESET overrules SET.

Oscillator (see Fig. 14)

The oscillator can be inhibited by the STOP instruction under software control. It is also inhibited when a low-supply voltage condition is present to prevent discharge of a weak back-up battery. Provided the supply voltage is within the operating range the oscillator will be restarted after a STOP instruction by a LOW level at the $\overline{INT}$/T0 pin or a HIGH level at the RESET pin.

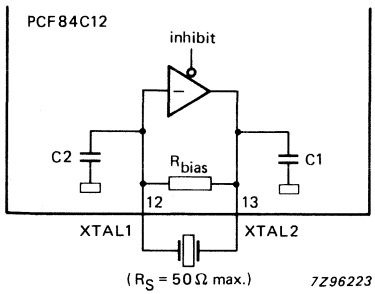


Fig. 14(a) Oscillator with integrated elements.

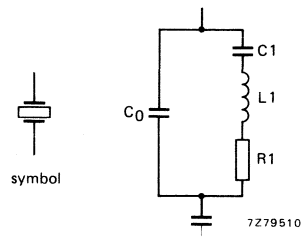


Fig. 14(b) Crystal unit equivalent circuit.

The values of crystal series resistance R1 and the crystal's total load capacitance C_L (C_0 + wiring + external capacitors) must not be above the curve (Fig. 14(c)) for the corresponding frequency. Note; if external capacitors are connected to XTAL 1 and XTAL they must be of equal value.

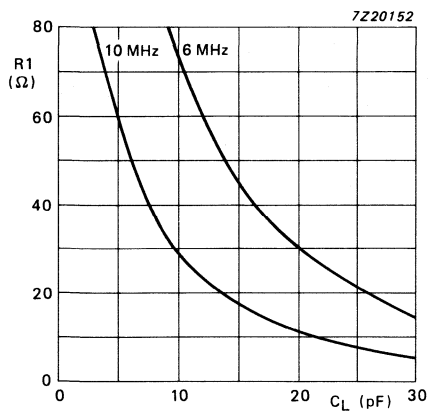


Fig. 14(c) Crystal circuit criteria.

The oscillator has the output drive capability via pin 13 (XTAL2). An external clock can be applied to pin 12 (XTAL1). A machine cycle consists of 10 time slots, each time slot being 3 oscillator periods.

Timer/event counter (see Fig. 15)

An internal 8-bit up-counter is provided. This can count external events, modulo-32 machine cycles, or machine cycles directly. Table 1 gives the instructions that control the counter and the prescaler, and the functions performed.

When used as a timer, the input to the counter is either the overflow or input of a 5-bit prescaler. When used as an event counter, LOW-to-HIGH transitions on pin 11 (T1) are counted. The maximum rate at which the counter may be incremented is once every machine cycle. When the counter overflows, the timer flag is set. The flag can be tested and reset using the JTF (jump is timer flag = 1) or JNTF instruction. Overflow also generates an interrupt to the processor via setting of the Timer Interrupt Flag when the timer/event counter interrupt is enabled.

FUNCTIONAL DESCRIPTION (continued)

Table 1 Timer/event counter control

function	timer mode modulo-1, modulo-32*	counter mode
CLEAR	MOV T,A (A) = 0 or RESET	MOV T,A (A) = 0 or RESET
PRESET	MOV T,A	MOV T,A
START	STRT T	STRT CNT
STOP	STOP TCNT or RESET	STOP TCNT or RESET
TEST	JTF/JNTF	JTF/JNTF
READ**	MOV A,T	MOV A,T

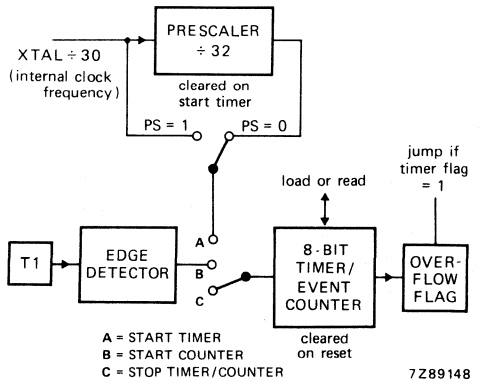


Fig. 15 Timer/event counter.

Program status word (see Fig. 16)

The program status word (PSW) is an 8-bit word (1 byte) in the CPU which stores information about the current status of the microcontroller.
The PSW bits are:

- Bits 0 to 2 — stack pointer bits (SP_0, SP_1, SP_2)
- Bit 3 — prescaler select (PS); 0 = modulo-32; 1 = modulo-1 (no prescaling)
- Bit 4 — working register bank select (RBS); 0 = register bank 0; 1 = register bank 1
- Bit 5 — not used (1)
- Bit 6 — auxiliary carry (AC); half-carry bit generated by an ADD instruction and used by the decimal adjust instruction DA A
- Bit 7 — carry (CY); the carry flag indicates that previous operation has resulted in an overflow of the accumulator.

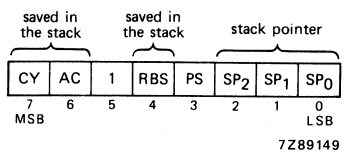


Fig. 16 Program status word.

* With prescaler select, PS = 0, the timer counts modulo-32 machine cycles, with PS = 1 it counts modulo-1 cycles (prescaler not used); prescaler cleared with STRT T, prescaler not readable.

** READ does not disturb the counting process.

Program status word (continued)

All bits can be read using the MOV A, PSW instruction. Bits 7 and 6 are set and cleared by CPU operation. Bit 4 can be changed by a SEL RB instruction, bit 3 by the MOV PSW, A instruction, and bits 0, 1 and 2 by the CALL, RET or RETR instructions and in the event of an interrupt. Bits 7, 6 and 4 are stored in the program counter stack during subroutine and interrupt calls. These bits are restored in the PSW with a RETR (return and restore) instruction which must be used at the end of an interrupt and can be used at the end of a normal subroutine. The RET instruction has no restore feature and cannot be used at the end of an interrupt.

Program counter (see Fig. 17)

The 13-bit program counter is able to address 8 K bytes of ROM. The arrangement of the bits is shown in figure 17. During an interrupt subroutine PC_{11} and PC_{12} are forced to logic 0. All 13 bits are saved in the stack during CALL and interrupt routines.

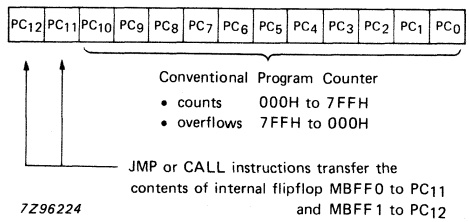


Fig. 17 Program counter.

Central processing unit

The PCF84C12 has arithmetic, logical and branching capabilities. The DA A, SWAP A and XCHD instructions simplify BCD arithmetic and the handling of nibbles. The MOVP A,@A instruction permits efficient table look-up from the current ROM page.

Conditional branch logic

The conditional branch logic within the processor enables several conditions, internal and external to the processor, to be tested by the user's program. Table 2 lists the conditional jump instructions used to change the program sequence. The DJNZ instruction decrements a designated register or data memory location and branches if the contents are not zero. This instruction is useful for looping control. The JMPP@A instruction allows multiway branches to destinations determined by the contents of the accumulator.

FUNCTIONAL DESCRIPTION (continued)

Table 2 Conditional branches

test	jump condition	jump instruction
accumulator	all bits zero	JZ
	any bit non-zero	JNZ
accumulator bit test	1	JB0 to JB7
carry flag	1	JC
	0	JNC
timer overflow flag	1	JTF
	0	JNTF
test input T0	1	JT0
	0	JNT0
test input T1	1	JT1
	0	JNT1
register	non-zero	DJNZ

Test input T1 (pin 11)

The T1 input line can be used as:

- A test input for branch instructions JT1 and JNT1
- An external input to the event counter

When used as a test input:

- JT1 instruction tests for logic 1 level
- JNT1 instruction tests for logic 0 level

When used as an input to the event counter, T1 must be LOW for > 4 CP, followed by a HIGH for > 4 CP. The transition can be recognized with a repetition rate of once per 30 oscillator clock periods (1 machine cycle).

There is no internal pull-up or pull-down resistor connected to the T1 input. If required it must be externally connected to a resistor ($R = \leqslant 100\ k\Omega$). When T1 is not used pin 11 must be connected to V_{DD} or V_{SS}.

Reset (pin 14)

A positive-going signal on the RESET input:

- Sets the program counter to zero
- Selects location 0 of memory band 0 and register bank 0
- Sets the stack pointer to zero (000); pointing to RAM address 8
- Disables the interrupts (external and timer)
- Stops the timer/event counter, then sets it to zero
- Sets the timer prescaler to modulo-32
- Resets the timer flag
- Sets all ports to input mode
- Cancels IDLE and STOP mode

Power-on-reset and low-voltage detection (see Fig. 18)

Correct operation of the PCF84C12 during moments of slowly changing supply voltages and low-voltage conditions is essential. This is achieved by the addition of an internal power-on-reset and low-voltage detection circuit. To allow an external RESET signal being fed into the PCF84C12, the reset pin (pin 14) has been configured as an input/output.

While a reset condition exists in the detection circuit, pin 14 is pulled HIGH by TR1 controlled by the reset circuit.

When the reset condition is not present a pull-down current source (TR2) will be activated. TR2 forces pin 14 LOW thus removing the RESET signal from the microcontroller.

Since the level at pin 14 is recognized by the microcontroller, the reset time constant can be stretched by connecting an external capacitor between V_{DD} and pin 14 (see Fig. 19).

The signal at pin 14 can also be used as an output to reset other devices in the system.

The internal reset circuit monitors the PCF84C12 supply voltage. If the voltage drops below the switching level (typ. 1,5 V), a reset (HIGH) is applied to pin 14. This reset is removed (pin 14 goes LOW), after a fixed delay (t_d), when the supply voltage rises above the switching level again. The delay ensures a complete reset even when the supply voltage quickly rises above switching level after initial switch-on.

If the supply voltage does not reach its minimum value for the frequency of operation (see Fig. 21) within 1860 clock periods, the internal RESET delay is not sufficient.

During a low-voltage condition the oscillator is inhibited to prevent complete discharge of a weak battery. The timing of the power-on-reset and low-voltage detection circuit is shown in Fig. 19.

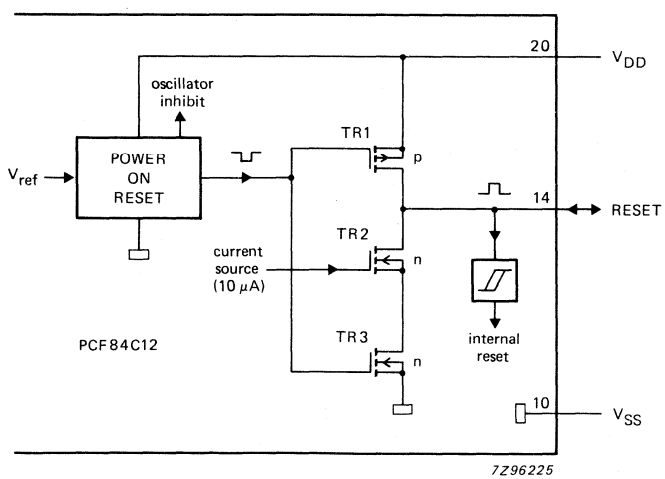
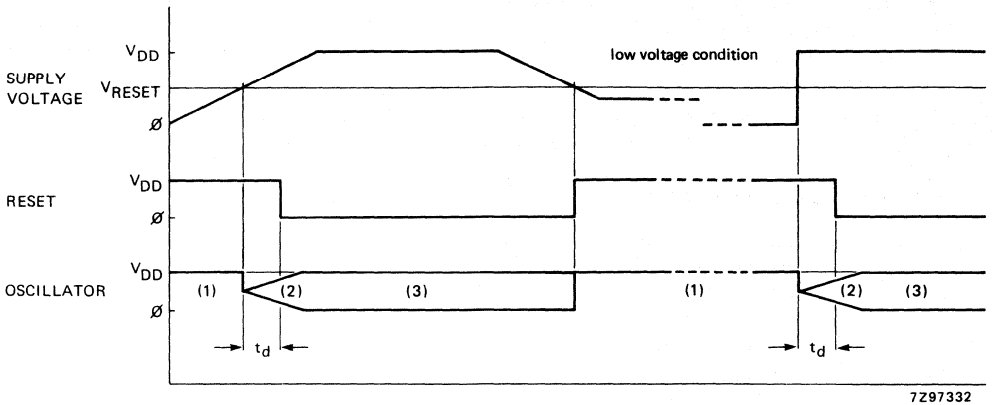


Fig. 18 Power-on-reset configuration.

FUNCTIONAL DESCRIPTION (continued)



Where: (1) Oscillator inhibited
(2) Oscillator starting
(3) Oscillator running, but may be stopped with a STOP condition

Fig. 19 Timing of power-on-reset and low-voltage detection.

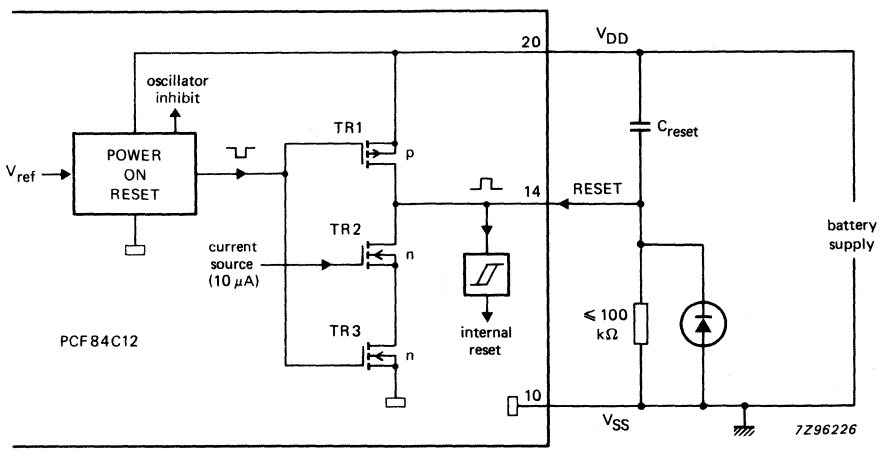


Fig. 20 Stretched power-on-reset with external capacitor.

INSTRUCTION SET

The PCF84C12 instruction set consists of over 80 one and two byte instructions and is identical to the MAB8400 instruction set. New instructions are added for STOP and IDLE mode. Program code efficiency is high because all RAM locations and all ROM locations on a 256 byte page require only a single byte address.

Table 5 gives the instruction set of the PCF84C12. Table 4 shows the instruction map and Table 3 details the symbols and definition descriptions that are used.

Table 3 Symbols and definitions used in Table 5

symbol	definition description
A	accumulator
addr	program memory address
Bb	bit designation (b = 0-7)
RBS	register bank select
C	carry bit (bit CY)
CNT	event counter
D	mnemonic for 4-bit digit (nibble)
data	8-bit number or expression
I	interrupt
MBFF	memory bank flip-flop
P	mnemonic for 'in-page' operation
PC	program counter
Pp	port designation (p = 0, 1 or 2)
PSW	program status word
RB	register bank
Rr	register designation (r = 0-7)
SP	stack pointer
T	timer
TF	timer flag
T0, T1	test 0 and 1 inputs
#	immediate data prefix
@	indirect address prefix
(X)	contents of X
((X))	contents of location addressed by X
←	is replaced by
↔	is exchanged with

DEVELOPMENT DATA

Table 4 PCF84C12 instruction map

first hexadecimal character of opcode ↓ / second hexadecimal character of opcode →	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NOP	IDLE		ADD A,#data	JMP page 0	EN I	JNTF addr	DEC A	IN A,Pp							
1	INC @Rr 0	INC @Rr 1	JBO addr	ADDC A,#data	CALL page 0	DIS I	JTF addr	INC A	INC Rr 0	INC Rr 1	INC Rr 2	INC Rr 3	INC Rr 4	INC Rr 5	INC Rr 6	INC Rr 7
2	XCH A,@Rr 0	XCH A,@Rr 1	STOP	MOV A,#data	JMP page 1	EN TCNTI	JNTO addr	CLR A	XCH A,Rr 0	XCH A,Rr 1	XCH A,Rr 2	XCH A,Rr 3	XCH A,Rr 4	XCH A,Rr 5	XCH A,Rr 6	XCH A,Rr 7
3	XCHD A,@Rr 0	XCHD A,@Rr 1	JB1 addr		CALL page 1	DIS TCNTI	JTO addr	CPL A	OUTL Pp,A							
4	ORL A,@Rr 0	ORL A,@Rr 1	MOV A,T	ORL A,#data	JMP page 2	STRT CNT	JNT1 addr	SWAP A	ORL A,Rr 0	ORL A,Rr 1	ORL A,Rr 2	ORL A,Rr 3	ORL A,Rr 4	ORL A,Rr 5	ORL A,Rr 6	ORL A,Rr 7
5	ANL A,@Rr 0	ANL A,@Rr 1	JB2 addr	ANL A,#data	CALL page 2	STRT T	JT1 addr	DA A	ANL A,Rr 0	ANL A,Rr 1	ANL A,Rr 2	ANL A,Rr 3	ANL A,Rr 4	ANL A,Rr 5	ANL A,Rr 6	ANL A,Rr 7
6	ADD A,@Rr 0	ADD A,@Rr 1	MOV T,A		JMP page 3	STOP TCNT		RRC A	ADD A,Rr 0	ADD A,Rr 1	ADD A,Rr 2	ADD A,Rr 3	ADD A,Rr 4	ADD A,Rr 5	ADD A,Rr 6	ADD A,Rr 7
7	ADDC A,@Rr 0	ADDC A,@Rr 1	JB3 addr		CALL page 3			RR A	ADDC A,Rr 0	ADDC A,Rr 1	ADDC A,Rr 2	ADDC A,Rr 3	ADDC A,Rr 4	ADDC A,Rr 5	ADDC A,Rr 6	ADDC A,Rr 7
8				RET	JMP page 4				ORL Pp,#data 0	ORL Pp,#data 1						
9			JB4 addr	RETR	CALL page 4		JNZ addr	CLR C	ANL Pp,#data 0	ANL Pp,#data 1						
A	MOV @Rr,A 0	MOV @Rr,A 1		MOVP A,@A	JMP page 5			CPL C	MOV Rr,A 0	MOV Rr,A 1	MOV Rr,A 2	MOV Rr,A 3	MOV Rr,A 4	MOV Rr,A 5	MOV Rr,A 6	MOV Rr,A 7
B	MOV @Rr,#data 0	MOV @Rr,#data 1	JB5 addr	JMPP @A	CALL page 5				MOV Rr,#data 0	MOV Rr,#data 1	MOV Rr,#data 2	MOV Rr,#data 3	MOV Rr,#data 4	MOV Rr,#data 5	MOV Rr,#data 6	MOV Rr,#data 7
C	DEC @Rr 0	DEC @Rr 1			JMP page 6	SEL RB0	JZ addr	MOV A,PSW	DEC Rr 0	DEC Rr 1	DEC Rr 2	DEC Rr 3	DEC Rr 4	DEC Rr 5	DEC Rr 6	DEC Rr 7
D	XRL A,@Rr 0	XRL A,@Rr 1	JB6 addr	XRL A,#data	CALL page 6	SEL RB1		MOV PSW,A	XRL A,Rr 0	XRL A,Rr 1	XRL A,Rr 2	XRL A,Rr 3	XRL A,Rr 4	XRL A,Rr 5	XRL A,Rr 6	XRL A,Rr 7
E	DJNZ @Rr,addr 0	DJNZ @Rr,addr 1			JMP page 7		JNC addr	RL A	DJNZ Rr, addr 0	DJNZ Rr, addr 1	DJNZ Rr, addr 2	DJNZ Rr, addr 3	DJNZ Rr, addr 4	DJNZ Rr, addr 5	DJNZ Rr, addr 6	DJNZ Rr, addr 7
F	MOV A,@Rr 0	MOV A,@Rr 1	JB7 addr		CALL page 7		JC addr	RLC A	MOV A,Rr 0	MOV A,Rr 1	MOV A,Rr 2	MOV A,Rr 3	MOV A,Rr 4	MOV A,Rr 5	MOV A,Rr 6	MOV A,Rr 7

Table 5 Instruction set

	mnemonic	opcode (hex.)	bytes/ cycles	description	function		notes
ACCUMULATOR	ADD A, Rr	6*	1/1	Add register contents to A	(A)←(A) + (Rr)	r = 0–7	1
	ADD A, @Rr	60 61	1/1	Add RAM data, addressed by Rr, to A	(A)←(A) + ((R0)) (A)←(A) + ((R1))		1
	ADD A, #data	03 data	2/2	Add immediate data to A	(A)←(A) + data		1
	ADDC A, Rr	7*	1/1	Add carry and register contents to A	(A)←(A) + (Rr) + (C)	r = 0–7	1
	ADDC A, @Rr	70 71	1/1	Add carry and RAM data, addressed by Rr, to A	(A)←(A) + ((R0)) + (C) (A)←(A) + ((R1)) + (C)		1
	ADDC A, #data	13 data	2/2	Add carry and immediate data to A	(A)←(A) + data + (C)		1
	ANL A, Rr	5*	1/1	'AND' Rr with A	(A)←(A) AND (Rr)	r = 0–7	
	ANL A, @Rr	50 51	1/1	'AND' RAM data, addressed by Rr, with A	(A)←(A) AND ((R0)) (A)←(A) AND ((R1))		
	ANL A, #data	53 data	2/2	'AND' immediate data with A	(A)←(A) AND data		
	ORL A, Rr	4*	1/1	'OR' Rr with A	(A)←(A) OR (Rr)	r = 0–7	
	ORL A, @Rr	40 41	1/1	'OR' RAM data, addressed by Rr, with A	(A)←(A) OR ((R0)) (A)←(A) OR ((R1))		
	ORL A, #data	43 data	2/2	'OR' immediate data with A	(A)←(A) OR data		
	XRL A, Rr	D*	1/1	'XOR' Rr with A	(A)←(A) XOR (Rr)	r = 0–7	
	XRL A, @Rr	D0 D1	1/1	'XOR' RAM, addressed by Rr, with A	(A)←(A) XOR ((R0)) (A)←(A) XOR ((R1))		
	XRL A, #data	D3 data	2/2	'XOR' immediate data with A	(A)←(A) XOR data		
	INC A	17	1/1	increment A by 1	(A)←(A) + 1		
	DEC A	07	1/1	decrement A by 1	(A)←(A) −1		
	CLR A	27	1/1	clear A to zero	(A)←0		
	CPL A	37	1/1	one's complement A	(A)←NOT(A)		
	RL A	E7	1/1	rotate A left	$(A_{n+1})\leftarrow(A_n)$ $(A_0)\leftarrow(A_7)$	n = 0–6	

INSTRUCTION SET (continued)

ACCUMULATOR (cont.)	RLC A	F7	1/1	rotate A left through carry	$(A_{n+1}) \leftarrow A_n$ $(A_0) \leftarrow (C), (C) \leftarrow (A_7)$	n = 0–6	2
	RR A	77	1/1	rotate A right	$(A_n) \leftarrow (A_{n+1})$ $(A_7) \leftarrow (A_0)$	n = 0–6	
	RRC A	67	1/1	rotate A right through carry	$(A_n) \leftarrow (A_{n+1})$ $(A_7) \leftarrow (C), (C) \leftarrow (A_0)$	n = 0–6	2
	DA A	57	1/1	decimal adjust A			2
	SWAP A	47	1/1	swap nibbles of A	$(A_{4-7}) \leftrightarrow (A_{0-3})$		
DATA MOVES	MOV A, Rr	F*	1/1	move register contents to A	(A)←(Rr)	r = 0–7	
	MOV A, @Rr	F0 F1	1/1	move RAM data, addressed by Rr, to A	(A)←((R0)) (A)←((R1))		
	MOV A, #data	23 data	2/2	move immediate data to A	(A)←data		
	MOV Rr, A	A*	1/1	move accumulator contents to register	(Rr)←(A)	r = 0–7	
	MOV @Rr, A	A0 A1	1/1	move accumulator contents to RAM location addressed by Rr	((R0))←(A) ((R1))←(A)		
	MOV Rr, #data	B* data	2/2	move immediate data to Rr	(Rr)←data		
	MOV @Rr, #data	B0 data B1 data	2/2	move immediate data to RAM location addressed by Rr	((R0))←data ((R1))←data		
	XCH A, Rr	2*	1/1	exchange accumulator contents with Rr	(A)↔(Rr)	r = 0–7	
	XCH A, @Rr	20 21	1/1	exchange accumulator contents with RAM data addressed by Rr	(A)↔((R0)) (A)↔((R1))		
	XCHD A, @Rr	30 31	1/1	exchange lower nibbles of A and RAM data addressed by Rr	$(A_{0-3}) \leftrightarrow ((R0_{0-3}))$ $(A_{0-3}) \leftrightarrow ((R1_{0-3}))$		
	MOV A, PSW	C7	1/1	move PSW contents to accumulator	(A)←(PSW)		
	MOV PSW, A	D7	1/1	move accumulator bit 3 to PSW_3	$(PSW_3) \leftarrow (A_3)$		3
	MOVP A, @A	A3	1/2	move indirectly addressed data in current page to A	$(PC_{0-7}) \leftarrow (A), (A) \leftarrow ((PC))$		
FLAGS	CLR C	97	1/1	clear carry bit	(C)←0		2
	CPL C	A7	1/1	complement carry bit	(C)←NOT(C)		2

DEVELOPMENT DATA

	mnemonic	opcode (hex.)	bytes/ cycles	description	function	notes
REGISTER	INC Rr	1*	1/1	increment register by 1	(Rr)←(Rr) + 1 r = 0–7	
	INC @Rr	10 11	1/1	increment RAM data, addressed by Rr, by 1	((R0))←((R0)) + 1 ((R1))←((R1)) + 1	
	DEC Rr	C*	1/1	decrement register by 1	(Rr)←(Rr) – 1 r = 0–7	
	DEC @Rr	C0 C1	1/1	decrement RAM data, addressed by Rr, by 1	((R0))←((R0)) – 1 ((R1))←((R1)) – 1	
BRANCH	JMP addr	• 4 address	2/2	unconditional jump within a 2 K bank	(PC_{8-10})←$addr_{8-10}$ (PC_{0-7})←$addr_{0-7}$ (PC_{11-12})←MBFF 0–1	
	JMPP @A	B3	1/2	indirect jump within a page	(PC_{0-7})←((A))	
	DJNZ Rr, addr	E* address	2/2	decrement Rr by 1 and jump if not zero to addr	(Rr)←(Rr) – 1 r = 0–7 if (Rr) not zero (PC_{0-7})←addr	
	DJNZ @Rr, addr	E0	2/2	decrement RAM data, addressed by Rr by 1 and jump if not zero to addr	((R0))←((R0)) – 1 if ((R0)) not zero (PC_{0-7})←addr	
		E1			((R1))←((R1)) – 1 if ((R1)) not zero (PC_{0-7})←addr	
	JBb addr	▲ 2 address	2/2	jump to addr if Acc. bit b = 1	if b = 1 : (PC_{0-7})←addr b = 0–7	
	JC addr	F6 address	2/2	jump to addr if C = 1	if C = 1 : (PC_{0-7})←addr	
	JNC addr	E6 address	2/2	jump to addr if C = 0	if C = 0 : (PC_{0-7})←addr	
	JZ addr	C6 address	2/2	jump to addr if A = 0	if A = 0 : (PC_{0-7})←addr	
	JNZ addr	96 address	2/2	jump to addr if A is NOT zero	if A≠0 : (PC_{0-7})←addr	
	JT0 addr	36 address	2/2	jump to addr if T0 = 1	if T0 = 1: (PC_{0-7})←addr	
	JNT0 addr	26 address	2/2	jump to addr if T0 = 0	if T0 = 0: (PC_{0-7})←addr	
	JT1 addr	56 address	2/2	jump to addr if T1 = 1	if T1 = 1: (PC_{0-7})←addr	
	JNT1 addr	46 address	2/2	jump to addr if T1 = 0	if T1 = 0: (PC_{0-7})←addr	
	JTF addr	16 address	2/2	jump to addr if Timer Flag = 1	if TF = 1: (PC_{0-7})←addr	4
	JNTF addr	06 address	2/2	jump to addr if Timer Flag = 0	if TF = 0: (PC_{0-7})←addr	

INSTRUCTION SET (continued)

TIMER/EVENT COUNTER	MOV A, T	42	1/1	move timer/event counter contents to accumulator	(A)←(T)	
	MOV T, A	62	1/1	move accumulator contents to timer/event counter	(T)←(A)	
	STRT CNT	45	1/1	start event counter		
	STRT T	55	1/1	start timer		
	STOP TCNT	65	1/1	stop timer/event counter		
	EN TCNTI	25	1/1	enable timer/event counter interrupt		
	DIS TCNTI	35	1/1	disable timer/event counter interrupt		
CONTROL	EN I	05	1/1	enable external interrupt		
	DIS I	15	1/1	disable external interrupt		
	SEL RB0	C5	1/1	select register bank 0	(RBS)←0	5
	SEL RB1	D5	1/1	select register bank 1	(RBS)←1	5
	STOP	22	1/1	enter STOP mode		
	IDLE	01	1/1	enter IDLE mode		
SUBROUTINE	CALL addr	▲4 address	2/2	jump to subroutine	((SP))←(PC), ($PSW_{4, 6, 7}$) 6 (SP)←(SP) + 1 (PC_{8-10})←$addr_{8-10}$ (PC_{0-7})←$addr_{0-7}$ (PC_{11-12})←MBFF 0–1	6
	RET	83	1/2	return from subroutine	(SP)←(SP) −1 (PC)←((SP))	6
	RETR	93	1/2	return from interrupt and restore bits 4, 6, 7 of PSW	(SP)←(SP) −1 ($PSW_{4, 6, 7}$) + (PC)←((SP))	6

	mnemonic	opcode (hex.)	bytes/ cycles	description	function	notes
PARALLEL INPUT/OUTPUT	IN A, Pp	08 09	1/2	input port p data to accumulator	(A)←(P0) (A)←(P1)	7
	OUTL Pp, A	38 39	1/2	output accumulator data to port p	(P0)←(A) (P1)←(A)	
	ANL Pp, #data	98 99	2/2	AND port p data with immediate data	(P0)←(P0) AND data (P1)←(P1) AND data	
	ORL Pp, #data	88 89	2/2	OR port p data with immediate data	(P0)←(P0) OR data (P1)←(P1) OR data	
	NOP	00	1/1	no operation		

Notes to Table 8

1. PSW CY, AC affected
2. PSW CY affected
3. PSW PS affected
4. Execution of JTF and JNTF instructions resets the Timer Flag (TF).
5. PSW RBS affected
6. PSW SP_0, SP_1, SP_2 affected

* : 8, 9, A, B, C, D, E, F
● : 0, 2, 4, 6, 8, A, C, E
▲ : 1, 3, 5, 7, 9, B, D, F

RATINGS

Limiting values in accordance with the Absolute Maximum System (IEC 134)

Supply voltage (pin 20)	V_{DD}		−0,8 to + 8	V
All input voltages	V_I		0,8 to V_{DD} + 0,8	V
D.C. current into any input or output	$\pm I_I, \pm I_O$	max.	10	mA
Total power dissipation (see note)	P_{tot}	max.	500	mW
Power dissipation per output	P_O	max.	50	mW
Storage temperature range	T_{stg}		−65 to + 150	°C
Operating ambient temperature range	T_{amb}		−40 to +85	°C
Operating junction temperature	T_j	max.	125	°C

HANDLING

Inputs and outputs are protected against electrostatic discharge in normal handling. However, to be totally safe, it is desirable to take normal precautions appropriate to handling MOS devices (see 'Handling MOS devices').

Note

Thermal resistance (junction to ambient)

for SOT-146	$R_{th\ j\text{-}a}$	max.	120	K/W
for SOT-163A	$R_{th\ j\text{-}a}$	max.	150	K/W

D.C. CHARACTERISTICS

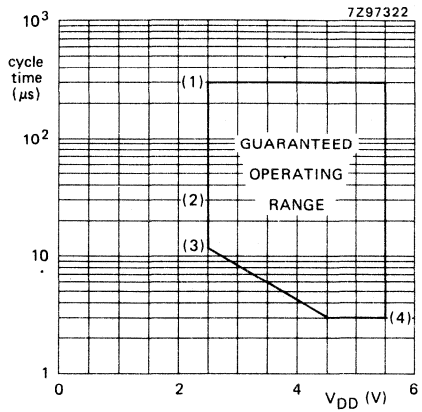
V_{DD} = 2,5 to 5,5 V; V_{SS} = 0 V; T_{amb} = −40 to + 85 °C; all voltages with respect to V_{SS}; unless otherwise specified.

parameter	symbol	min.	typ.	max.	unit
Supply voltage					
operating (see Fig. 21)	V_{DD}	2,5	–	5,5	V
Supply current					
operating (see Fig. 22)					
at V_{DD} = 5 V; f_{XTAL} = 6 MHz	I_{DD}	–	2	tbf	mA
at V_{DD} = 3 V; f_{XTAL} = 3,58 MHz	I_{DD}	–	300	tbf	μA
IDLE mode (see Fig. 23)					
at V_{DD} = 5 V; f_{XTAL} = 6 MHz	I_{DD}	–	1	tbf	mA
at V_{DD} = 3 V; f_{XTAL} = 3,58 MHz	I_{DD}	–	150	tbf	μA
STOP mode (see Fig. 24 and note 1)					
at V_{DD} = 2,5 V; T_{amb} = 25 °C	I_{DD}	–	1,2	2,5	μA
at V_{DD} = 2,5 V; T_{amb} = 85 °C	I_{DD}	–	–	10	μA
RESET I/O					
Switching level	V_{RESET}	–	1,5	2	V
Sink current					
at $V_{DD} > V_{RESET}$	I_{OL}	–	7	–	μA
Inputs					
Input voltage LOW	V_{IL}	0	–	$0{,}3V_{DD}$	V
Input voltage HIGH	V_{IH}	$0{,}7V_{DD}$	–	V_{DD}	V
Input leakage current					
as $V_{SS} < V_I < V_{DD}$	$\pm I_{IL}$	–	–	1	μA
Outputs					
Output voltage LOW					
at $V_I = V_{SS}$ or V_{DD}; $\lvert I_O \rvert < 1$ μA	V_{OL}	–	–	0,05	V
Output sink current LOW					
at V_{DD} = 5 V ± 10%; V_O = 0,4 V	I_{OL}	1,6	3	–	mA
Pull-up output source current HIGH					
at V_{DD} = 5 V ± 10%; V_O = 0,7 V_{DD}	$-I_{OH}$	40	–	–	μA
at V_{DD} = 5 V ± 10%; $V_O = V_{SS}$	$-I_{OH}$	–	–	400	μA
Push-pull output source current HIGH					
at V_{DD} = 5 V ± 10%; $V_O = V_{DD}$−0,4 V	$-I_{OH}$	1,6	3	–	mA

Note 1

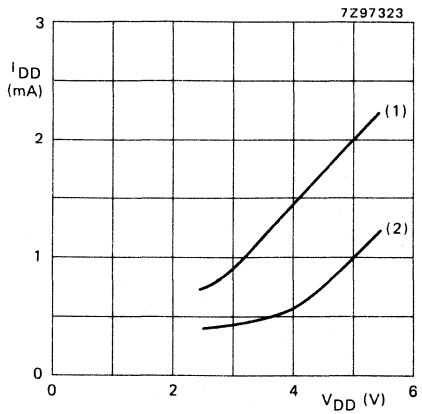
Crystal connected between XTAL1 and XTAL2; T1 at V_{SS}, $\overline{INT}$ at V_{DD}.

DEVELOPMENT DATA



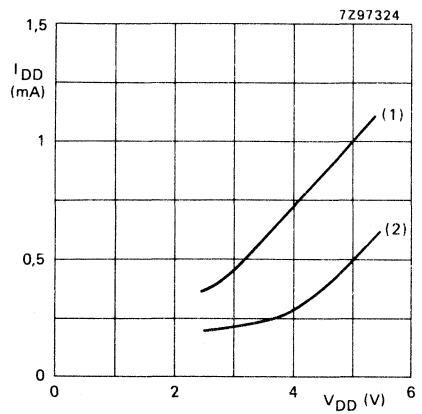
(1) clock frequency = 100 kHz
(2) clock frequency = 1 MHz
(3) clock frequency = 3 MHz
(4) clock frequency = 10 MHz

Fig. 21 Maximum clock frequency (f_{XTAL}) as a function of the supply voltage (V_{DD}).



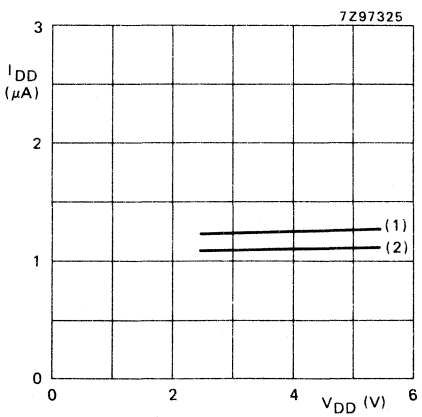
(1) clock frequency = 6 MHz
(2) clock frequency = 3,58 MHz

Fig. 22 Typical supply current (I_{DD}) in operating mode as a function of the supply voltage (V_{DD}); T_{amb} = 25 °C.



(1) clock frequency = 6 MHz
(2) clock frequency = 3,58 MHz

Fig. 23 Typical supply current (I_{DD}) in IDLE mode as a function of the supply voltage (V_{DD}); T_{amb} = 25 °C.



(1) T_{amb} = 85 °C
(2) T_{amb} = 25 °C

Fig. 24 Typical supply current (I_{DD}) in STOP mode as a function of the supply voltage (V_{DD}).

SINGLE-CHIP 8-BIT MICROCONTROLLER

DESCRIPTION

The PCF84CXX family of microcontrollers is manufactured in CMOS technology. The family consists of the following devices:

- PCF84C00 – 256 RAM bytes, external program memory
- PCF84C21 – 2 K ROM/64 RAM bytes
- PCF84C41 – 4 K ROM/128 RAM bytes
- PCF84C81 – 8 K ROM/256 RAM bytes

Each version has 20 quasi-bidirectional I/O port lines, a serial I/O interface, a single-level vectored interrupt circuit, an 8-bit timer event counter and on-board clock oscillator and clock circuits.

This microcontroller family is an efficient controller as well as an arithmetic processor. The instruction set is based on that of the MAB8048 and is pin- and instruction set compatible with the MAB8400 family.
The microcontrollers have bit handling abilities and facilities for both binary and BCD arithmetic.

For detailed information see the user manual "Single-chip 8-bit microcontrollers".

Features

- 8-bit CPU, ROM, RAM, I/O in a single 28-lead DIL or SO package
- 2 K, 4 K or 8 K ROM bytes plus a ROM-less version
- 64, 128 or 256 RAM bytes
- 20 quasi-bidirectional I/O port lines
- Two test inputs: one of which is also the external interrupt input
- Single-level vectored interrupts: external, timer/event counter, serial I/O
- I^2C hardware interface for serial data transfer on two lines; (serial I/O data via an existing port line and clock via a dedicated line)
- 8-bit programmable timer/event counter
- Clock frequency 100 kHz to 10 MHz
- Over 80 instructions (based on MAB8048) all of 1 or 2 cycles
- Single supply voltage from 2,5 V to 5,5 V
- STOP and IDLE mode
- Power-on-reset circuit
- Operating temperature range: –40 to +85 °C

PACKAGE OUTLINES

PCF84C21/41/81P: 28-lead DIL; plastic (SOT-117).
PCF84C21/41/81T: 28-lead mini-pack; plastic (SO-28; SOT-136A).
PCF84C00B : 28-lead 'Piggy-back' package (with up to 28-pin EPROM on top).
PCF84C00T : 56-lead mini-pack; plastic (VSO-56; SOT-190).

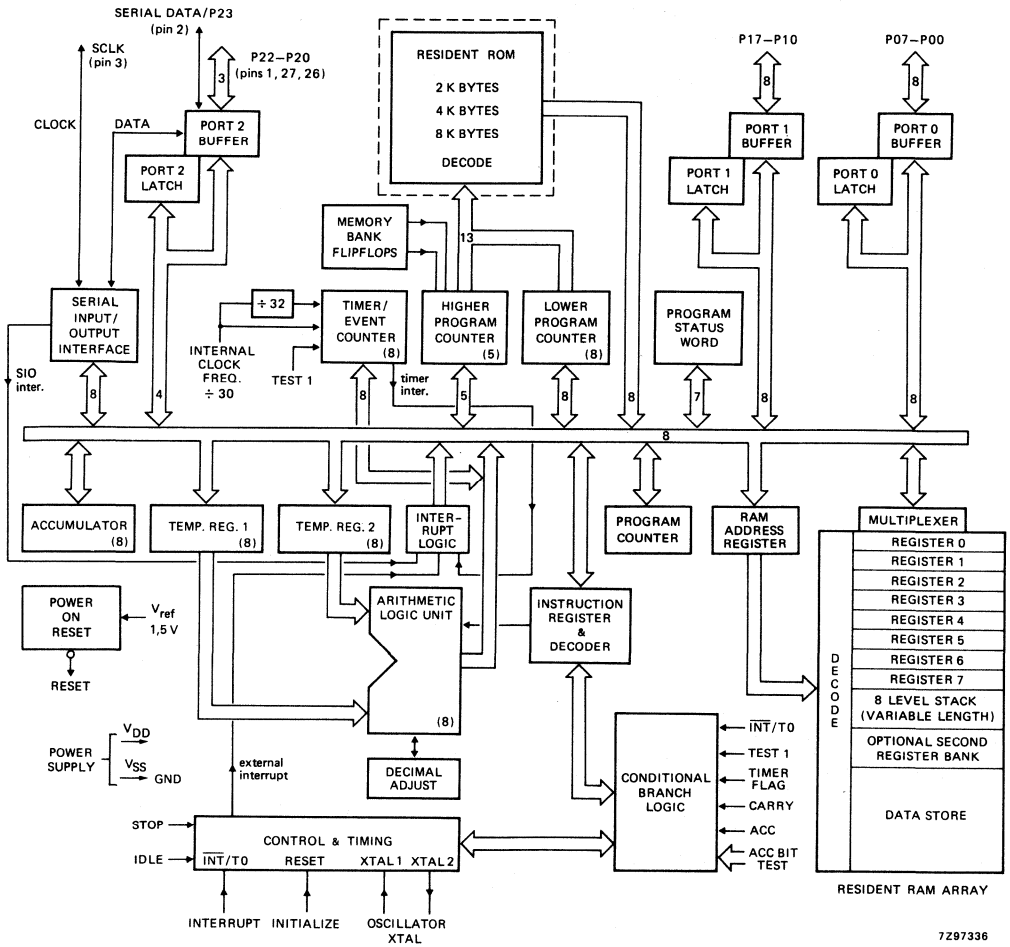
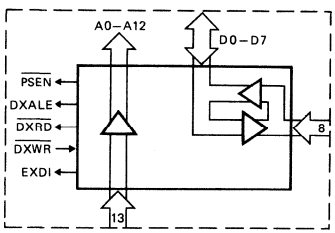
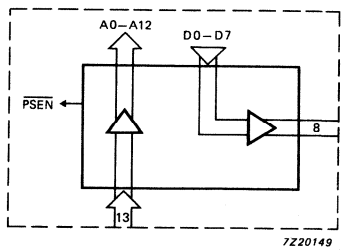


Fig. 1 Block diagram; PCF84CXX.



(a)

Fig. 1a Replacement of dotted part in Fig. 1, for the PCF84C00T ROM-less version.



(b)

Fig. 1b Replacement of dotted part in Fig. 1, for the PCF84C00B 'Piggy-back' version.

DEVELOPMENT DATA

PINNING

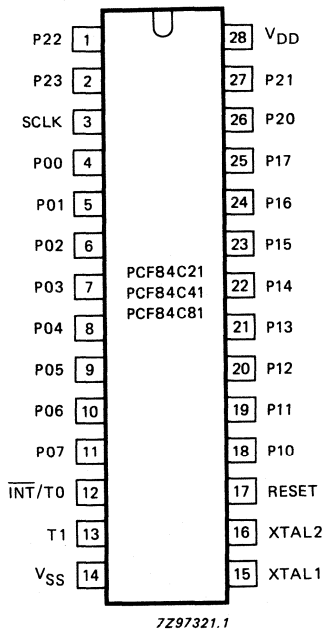


Fig. 2 Pin designation of the PCF84C21/41/81.

PIN DESIGNATION

Pin	symbol	type	function
3	SCLK	I/O	**Clock:** bidirectional clock for serial I/O.
4-11	P00-P07	I/O	**Port 0:** 8-bit quasi-bidirectional I/O port.
12	$\overline{INT}$/T0	I	**Interrupt/Test 0:** external interrupt input (sensitive to negative-going edge)/test input pin; when used as a test input directly tested by conditional branch instructions JT0 and JNT0.
13	T1	I	**Test 1:** test input pin, directly tested by conditional branch instructions JT1 and JNT1. T1 also functions as an input to the 8-bit timer/event countertimer/event counter, using the STRT CNT instruction.
14	V_{SS}	I	**Ground:** circuit earth potential.
15	XTAL 1	I	**Oscillator input:** crystal which determines the internal oscillator frequency or the external clock generator.
16	XTAL 2	I/O	Connection to other side of timing component.
17	RESET	I/O	**Reset input:** used to initialize the processor (active HIGH), or output of power-on-reset circuit.
18-25	P10-P17	I/O	**Port 1:** 8-bit quasi-bidirectional I/O port.
26,27, 1,2	P20-P23	I/O	**Port 2:** 4-bit quasi-bidirectional I/O port. P23 is the serial data input/output in serial I/O mode.
28	V_{DD}	I	**Power supply:** 2,5 V to 5,5 V.

PINNING (continued)

Pin designation of PCF84C00B piggy-back version

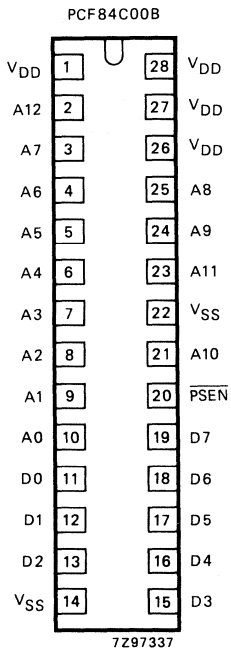


Fig. 3 Pinning diagram: PCF84C00B 'Piggy-back' version top pinning; to access a 2732 or 2764 EPROM.

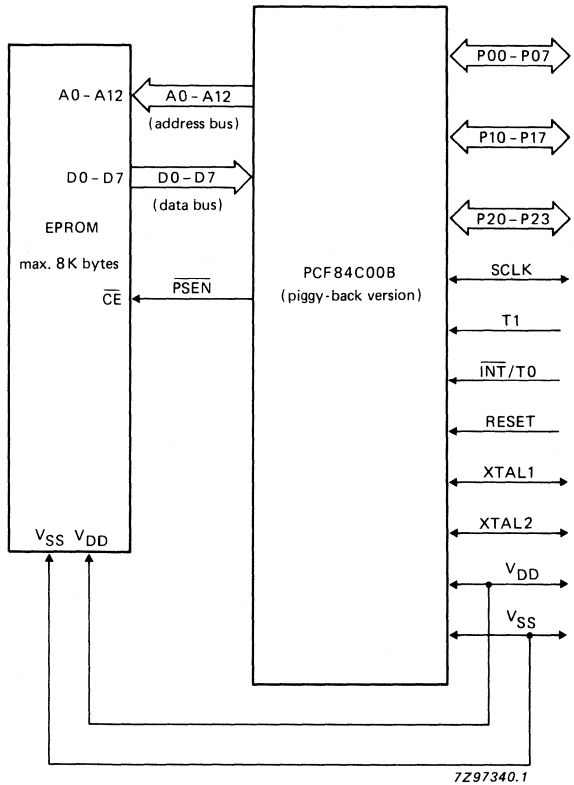


Fig. 3a Connection of EPROM to 'Piggy-back' package PCF84C00B.

The PCF84C00B is chiefly used in prototyping and low volume production. The device is mounted in a 'piggy-back' package, i.e. a 4 K or 8 K byte EPROM may be mounted in the 24/28 pin socket on top of the package.

Notes

1. RAM capacity of PCF84C00B is 256 bytes.
2. Access time for ROMS/EPROMS to be below 7 x $1/f_{XTAL}$.
3. Bottom pinning is identical to that of Fig. 2.

DEVELOPMENT DATA

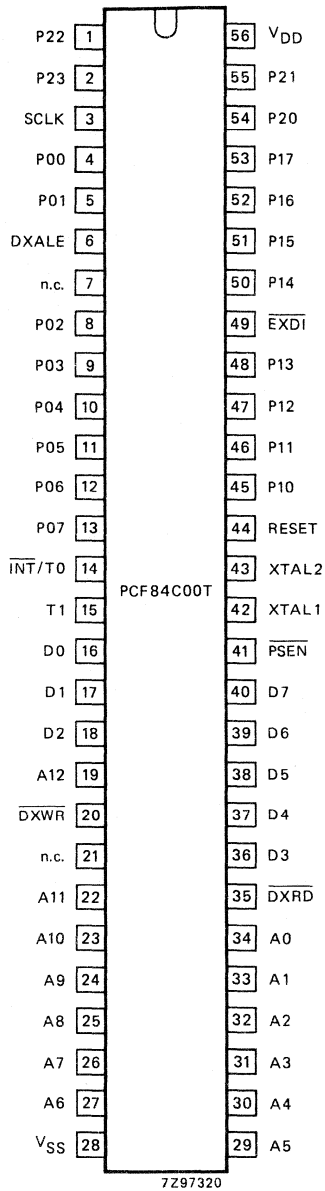


Fig. 4 Pinning diagram; ROM-less version PCF84C00T.

The PCF84C00T is chiefly used for prototyping future derivatives of the PCF84CXX family or low-volume production. The device is packaged in a 56-lead VSO outline. Additional signals are available (see pinning information following) to control external program memory and derivative functions.

PIN DESIGNATION (continued)

pin	symbol	type	function
34-29 27-22, 19	A00-A12	O	**Address** outputs.
16-18 36-40	D0-D7	I/O	**Data lines,** internal pull-ups provided.
41	$\overline{\text{PSEN}}$	O	**Program store enable** (active low). Used for enabling external program memory. Active during TS9 and TS10 of each machine cycle and TS1 of each following cycle. $\overline{\text{PSEN}}$ is halted during STOP mode.
6	DXALE	O	**Address latch enable.** Using the falling edge, the Dx address can be latched in an appropriate external latch. This signal occurs only during execution of the MOV Dx,A, MOV A,Dx, ANL Dx,A and ORL Dx, A instructions, with x = O to FF H.
35	$\overline{\text{DXRD}}$	O	**Read strobe** (active LOW). When this signal is active, external registers emulating Dx registers can be enabled to the data bus of the PCF84C00. This signals occurs only during execution of MOV A, Dx, ANL Dx,A and ORL Dx,A instructions, with x = O to FF H.
20	$\overline{\text{DXWR}}$	O	**Write strobe** (active low). On the rising edge, data on D0-D7 can be latched in appropriate external registers emulating Dx. This signal occurs only during execution of MOV Dx,A, ANL Dx,A and ORL Dx,A instructions, with x = O to FF H.
49	$\overline{\text{EXDI}}$	I	**External derivative interrupt** (active low). EXDI is 'OR-ed' with the internal serial interrupt and can be used to initiate an interrupt from external hardware emulating derivative functions. An internal pull-up is provided. A derivative interrupt is internally polled during time slot TS6*, and is only accepted if an EN SI instruction has been executed and the device is not already in an interrupt routine. Derivative interrupts are not latched in the PCF84C00.

* The interrupt signal must be continually active until the vector address 05 H is present on the address bus.

FUNCTIONAL DESCRIPTION

ROM-less version PCF84C00T

The PCF84C00T is a microcontroller that contains no on-board ROM, but has all address and data lines brought-out to access an external ROM or EPROM. This version has more pins than the PCF84CXX with on-board ROM (see Fig. 1a). The device has an address range of 8 K for ROM or external program memory, and has an internal data RAM of 256 bytes.

'Piggy-back' version PCF84C00B

The PCF84C00B is a special package that has standard pinning to the bottom which facilitates insertion as a mask-programmed device. An EPROM can be mounted on top in an additional socket. The total package height is greater than the standard DIL package. The device has an address range of 8 K for ROM or external RAM, and has an internal data RAM of 256 bytes.

Program memory

The program memory consists of 2, 4 or 8 K bytes, in a read-only memory (ROM). Each location is directly addressable by the program counter. The memory is mask-programmed at the factory. Figure 5 shows the program memory map.

Four program memory locations are of special importance:

- Location 0; contains the first instruction to be executed after the processor is initialized (RESET),
- Location 3; contains the first byte of an external interrupt service subroutine;
- Location 5; contains the first byte of a serial I/O and derivative interrupt service subroutine,
- Location 7; contains the first byte of a timer/event counter interrupt service subroutine.

Program memory is arranged in banks of 2 K bytes, which are selected by SEL MB instructions. The program memory is further divided into location 'pages', each of 256 bytes. This latter division applies only for conditional branches. Memory bank boundaries can be crossed only by using unconditional branch instructions after the appropriate memory bank has been selected. A CALL instruction can transfer control to a subroutine on any 'page'; RET and RETR instructions can transfer control from a subroutine back to the main program.

Data memory

Data memory consists of 64, 128 or 256 bytes, random-access data memory (RAM). Allocations are indirectly addressable using RAM pointer registers; up to 16 designated locations are directly addressable. Memory also includes an 8-level program counter stack addressed by a 3-bit stack pointer. Figure 6 shows the data memory map.

Working registers

Locations 0 to 7 are designated as working registers, directly addressable by the direct register instructions. Ease of addressing, and a minimum requirement of instruction bytes to manipulate their contents, makes these locations suitable for storing frequently addressed intermediate results. This bank of registers can be selected by the SEL RB0 instruction.

Executing the select register bank instruction SEL RB1, designates locations 24 to 31 as working registers, instead of locations 0 to 7, and these are then directly addressable. This second bank of working registers may be used as an extension of the first or reserved for use during interrupt service subroutines saving the first bank for use in the main program. If the second bank is not used, locations 24 to 31 may serve as general purpose RAM.

The first locations of each bank contain the RAM pointer registers R0, R1, R0' and R1', which indirectly address all RAM locations.

All RAM locations make efficient program loop counters when used with the decrement register and test instruction DJNZ.

FUNCTIONAL DESCRIPTION (continued)

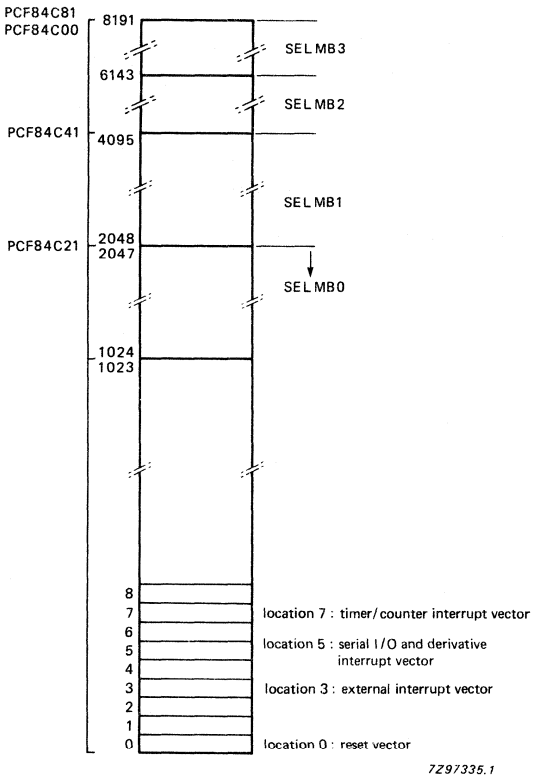


Fig. 5 Program memory map.

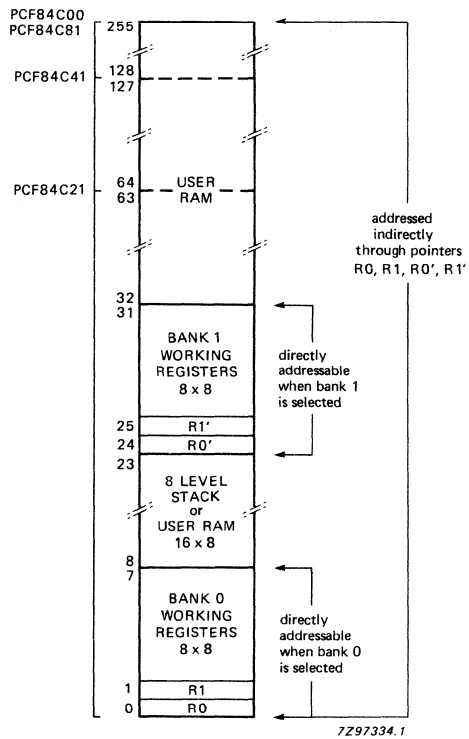


Fig. 6 Data memory map.

Program counter stack

Locations 8 to 23 may be designated as an 8-level program counter stack (2 locations per level), or as general purpose RAM. The program counter stack (Fig. 7) enables the processor to keep track of the return addresses and status generated by interrupts or CALL instructions by storing the contents of the program counter prior to servicing the subroutine. A 3-bit stack pointer determines which of the eight register pairs of the program counter stack will be loaded with next generated return address.

The stack pointer, when initialized to 000 by RESET, points to RAM locations 8 and 9. On the first subroutine CALL or interrupt, the contents of the program counter and bits 4, 6 and 7 of the program status word (PSW) are transferred to locations 8 and 9. The stack pointer increments by one and points to locations 10 and 11 ready for another CALL. Because an address may be up to 13 bits long, two bytes must be used to store each address.

At the end of a subroutine, which is signalled by a return instruction (RET or RETR), the stack pointer decrements by one and the contents of the register pair on top of the stack are transferred to the program counter. The saved PSW bits are transferred to the PSW only by the RETR instruction.

If not all 8 levels of subroutine and interrupt nesting are used, the unused portion of the stack may be used as any other indirectly addressable RAM locations.

Nesting of subroutines within subroutines can continue up to 8 times without overflowing the stack. If overflow does occur the deepest address stored (locations 8 and 9) will be overwritten and lost since the stack pointer overflows from 111 to 000. It also underflows from 000 to 111.

FUNCTIONAL DESCRIPTION (continued)

The value of the saved contents of the program counter is different for an interrupt CALL compared to a normal CALL to subroutine. With an interrupt CALL, the program counter return address is saved; with a subroutine CALL, the saved program counter value is one less than the program counter return address.

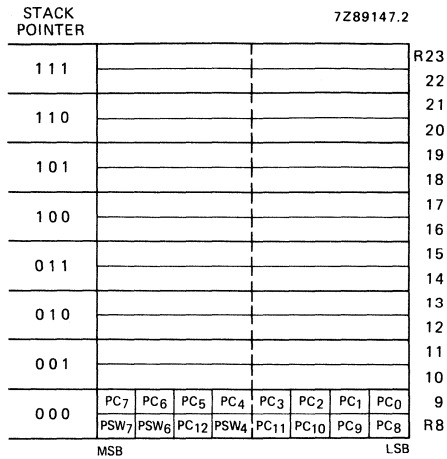


Fig. 7 Program counter stack.

IDLE and STOP modes

IDLE mode

When the microcontroller enters the IDLE mode via the IDLE instruction (01 H) the oscillator, timer/counter and serial I/O are kept running. The microcontroller exits from the IDLE mode by one of three interrupts if they are enabled or by activating a RESET. If the interrupt is not enabled the processor will remain in the IDLE mode. An active signal on the RESET pin restarts the microcontroller and a normal RESET sequence is executed (see Fig. 8).

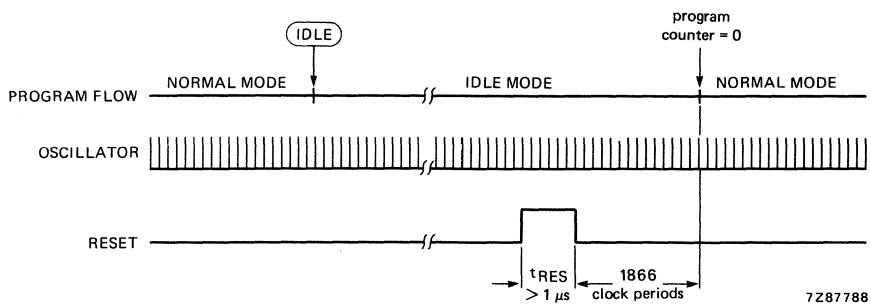


Fig. 8 Exit from IDLE mode via a RESET.

An active signal coming from an enabled interrupt causes the execution of the normal interrupt routine since normal interrupt scanning is still being carried out. A HIGH-to-LOW transition on the external interrupt pin ($\overline{INT}$/T0) reactivates the microcontroller. A LOW level applied to $\overline{INT}$/T0 will reactivate the microcontroller only in the STOP mode. Thus,if $\overline{INT}$/T0 was LOW before the microcontroller entered the IDLE mode, it must go HIGH before the microcontroller can be reactivated (see Fig. 9).

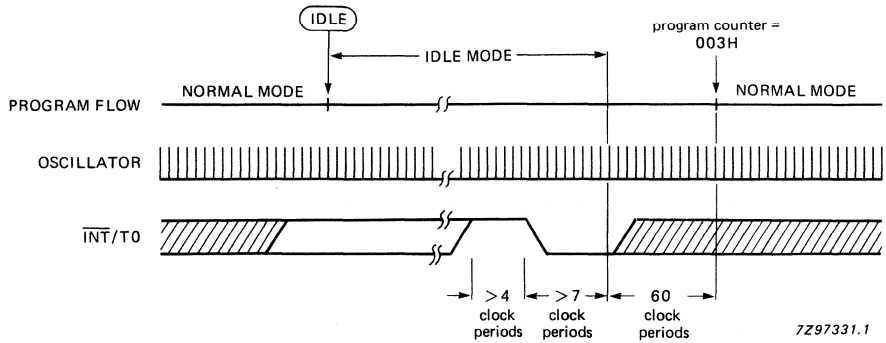


Fig. 9 Exit from IDLE mode via an interrupt.

Wake-up from the IDLE mode is ensured when $\overline{INT}$/T0 is HIGH for at least 4 CP (clock periods) followed by a LOW for 7 CP. After the initial forced CALL 003 H operation (60 CP) the program continues with the external interrupt service routine.

STOP mode

The microcontroller enters the STOP mode by the STOP instruction (22 H). The oscillator is switched off. The internal status of the CPU, RAM contents and the state of I/O ports are not affected. The microcontroller can be brought-out of the STOP mode by an active signal at the external interrupt input or by an external RESET signal. When one of these two signals is applied an internal delay of 1866 CP is provided to ensure that all internal clocks are operating correctly before restart (see Fig. 10). Note; the start-up time of a crystal oscillator is measured in milliseconds, and the 1866 CP count begins after this start-up time.

If the microcontroller exits from the STOP mode by activating RESET, a normal RESET sequence is executed.

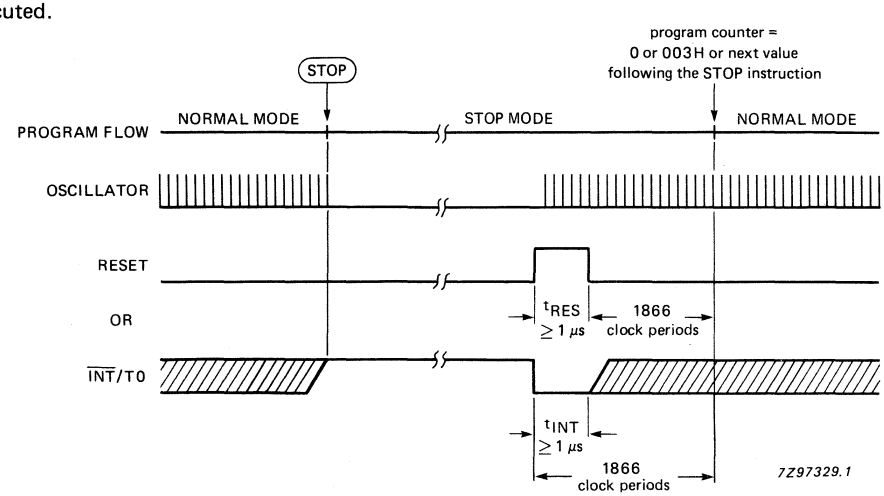


Fig. 10 Entering and exiting the STOP mode.

DEVELOPMENT DATA

FUNCTIONAL DESCRIPTION (continued)

If the microcontroller exits the STOP mode by pulling the external interrupt input pin LOW, an interrupt sequence is executed only if the external interrupt is enabled. In this event the microcontroller resumes the normal program sequence after returning from the interrupt routine, as in the normal mode. If the interrupt is not enabled, it continues the normal program sequence, executing the instruction following the STOP instruction.

The microcontroller is restarted by a LOW level applied at the $\overline{INT}$/T0 pin, and not by a HIGH-to-LOW transition as in a normal interrupt mechanism.
Note: when leaving the STOP mode with an interrupt, a further instruction in the main program series is executed prior to entering the interrupt routine.

When the $\overline{INT}$/T0 level is active during the STOP instruction then no STOP is executed.

A LOW level on the external interrupt input of at least 1 μs will cause the microcontroller to exit the STOP mode.

I/O facilities

The PCF84CXX family has 23 I/O lines arranged as:

- Port 0 parallel port of 8 lines (P00 to P07)
- Port 1 parallel port of 8 lines (P10 to P17)
- Port 2 parallel port of 4 lines (P20 to P23)
- SCLK serial I/O clock line
- $\overline{INT}$/T0 external interrupt and test input. When used as a test input it can be directly tested by conditional branch instructions JT0 and JNT0
- T1 test input which can alter program sequences when tested by conditional jump instructions JT1 and JTN1. T1 also functions as an input to the 8-bit timer/event counter.

Parallel ports

All parallel ports can be used as outputs or inputs, their structure is quasi-bidirectional.
Output data written to a port is latched and remains unchanged until rewritten.
Input data is not latched and so must be present until read by an input instruction.

Input lines are fully CMOS compatible, output lines can drive one TTL or CMOS load.

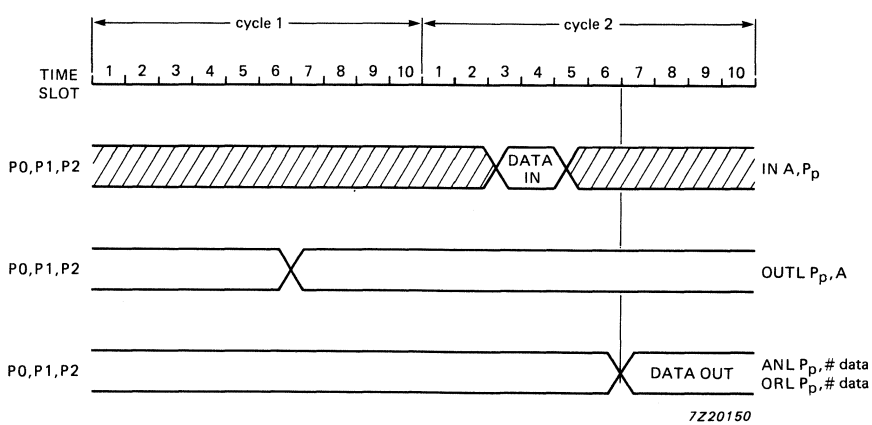


Fig. 11 Shows the timing diagram for all ports using IN, OUTL, ANL and ORL instructions. For the OUTL instruction data changes on time slot 7 of cycle 1. For the MOV, ANL and ORL instructions, the ports change on time slot 7 of cycle 2.

Fig. 12 shows the quasi-bidirectional I/O interface with push-pull output and switched pull-up current source.
Each line is pulled up to V_{DD} via a constant current source (TR4), which is enabled via TR3 whenever one of the two output latches contains a logic 1. This current source is sufficient for a TTL HIGH level, yet can be pulled LOW by an external CMOS device, thus allowing the same pin to be used for both input and output.

When a logic 1 is written to the line for the first time (MQ = 1, SQ = 0), TR2 is switched on for the duration of the internal write pulse (one oscillator period), to provide a fast transition from logic 0 to logic 1. Subsequent writing of a logic 1 to the port lines will not switch TR2 on. This prevents unnecessary current through external components connected to the port lines of the same port which might be in the input mode and also connected to ground.

When a logic 0 is written to the line, TR3 switches off the current source. Current sinking capability is provided by TR1, which is now switched on. When used as an input, a logic 1 must first be written to the line, otherwise TR1 will remain low impedance.

The PCF84CXX family offers the possibility to select individually 19 of the 20 parallel port pins (not P23), by the following mask options: (PCF84C00 has only option 1):

Option 1 – STANDARD PORT; quasi-bidirectional I/O with switched pull-up current source of 100 μA (typ.) and P-channel booster transistor TR2.
TR2 is only active during 1 clock cycle (Fig. 12).

Option 2 – OPEN DRAIN; quasi-bidirectional I/O with only an N-channel open drain output. Application as an output requires connection of an external pull-up resistor (Fig. 13).

Option 3 – PUSH-PULL OUTPUT; drive capability of the output will be 1,6 mA (min.) at V_{DD} = 5 V in both polarities. To avoid a large current flowing through the output transistors during the input mode, these push-pull pins must only be used as outputs (Fig. 14).

DEVELOPMENT DATA

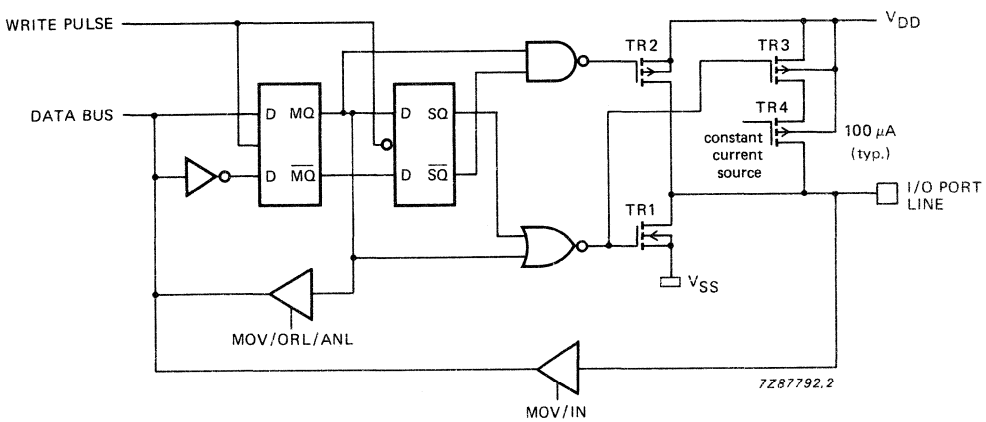


Fig. 12 Standard output with switched pull-up current source.

FUNCTIONAL DESCRIPTION (continued)

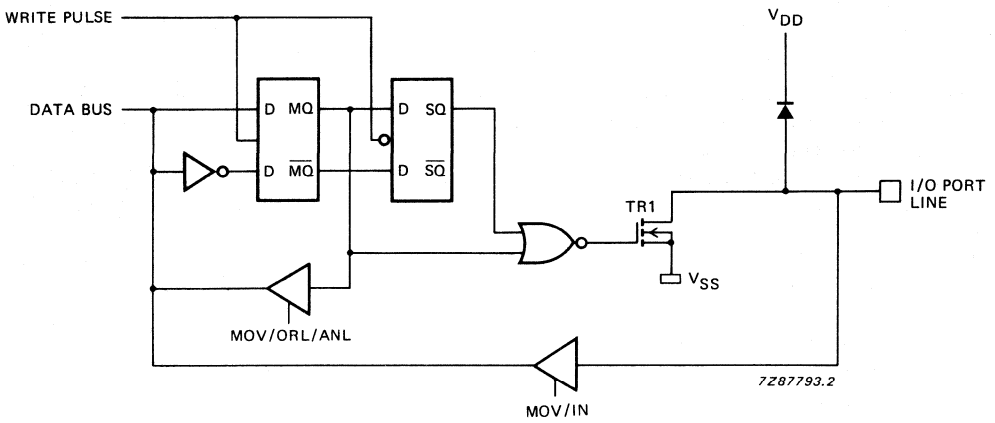


Fig. 13 Open drain output.

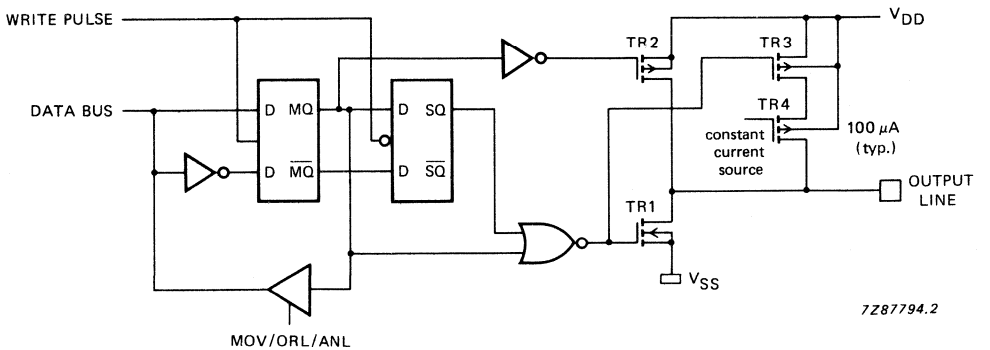


Fig. 14 Push-pull output.

Serial I/O (SIO)

The PCF84CXX has an on-chip serial I/O interface that supports I^2C-bus communication. Whereas a normal microcontroller must regularly monitor the serial data bus for the presence of data, the serial I/O interface detects, receives and converts the serial data stream into parallel format without interrupting the execution of the current program. An interrupt is sent to the PCF84CXX only when a complete byte is received. It then reads the data byte in one instruction.

The design of the PCF84CXX serial I/O system allows any number of devices from PCF8500 family (clips) to be connected via the two-line serial bus. The ability of any devices to communicate, without interrupting the operation of any other devices on the bus, is an outstanding attribute of the system. This is achieved by allocating a specific 7-bit address to each device and providing a system whereby a device reacts only to a message prefixed with its own address or the 'general CALL' address. Address recognition is performed by the interface hardware so that operation of the microcontroller need only be interrupted when a valid address has been received. This saves significant processing time and memory space compared with a conventional microcontroller employing a software serial interface. When the addressing facility is not required, for instance in a system with only two microcontrollers, direct data transfer without addressing can be performed. In multi-master systems, an automatically invoked arbitration procedure prevents two or more devices from continuing simultaneous transmission.

In NORMAL (running) and IDLE mode, the serial I/O logic remains active; its internal system clock will be switched off when there is no activity on the serial bus.

After execution of the STOP instruction, the oscillator of the PCF84CXX is switched off. This means that the serial I/O logic will remain in the state it was at the occurance of the STOP instruction. To avoid "bus block" problems and to assure correct start-up of the bus after exit from the STOP mode, the user should disable the serial logic (ESO = 0) prior to the execution of the STOP instruction. This must be carried out only when the PCF84CXX has finished a serial data transfer.

Serial I/O interface

Figure 15 shows the serial I/O interface. The clock line of the serial bus has exclusive use of pin 3 (SCLK) while the data line shares pin 2 (serial data) with the I/O line P23 of port 2. When the serial I/O is enabled, P23 is disabled as a parallel port line; (P23 and SCLK only open drain).

The microcontroller and interface communicate via the internal microcontroller bus and the Serial Interrupt Request line. Data and information controlling the operation of the interface are stored in four registers:

- Data shift register (S0)
- Serial I/O interface status word (S1)
- Serial clock control word (S2)
- Address register

FUNCTIONAL DESCRIPTION (continued)

Data shift register (S0)

Register S0 converts serial data to parallel format and vice versa. A pending interrupt is generated only after a complete byte has been transmitted, or after a complete data byte, specific address or 'general CALL' address has been received. The most significant bit is transmitted first.

Serial I/O interface status word (S1)

Register S1 provides information concerning the state of the interface and stores information from the microcontroller. Bits 0 to 3 are duplicated: control bits in these positions can only be written by the microcontroller, while interface bits can only be read.

MST and TRX (see Table 1)

These bits determine the operating mode of the serial I/O interface.

Table 1 Operating modes of the serial I/O interface

MST	TRX	operating mode
0	0	slave receiver
1	0	master receiver
0	1	slave transmitter
1	1	master transmitter

BB: Bus Busy.

This is the flag which indicates the status of the bus.

PIN: Pending Interrupt Not

PIN = '0' indicates the presence of a pending interrupt, which will cause a Serial Interrupt Request when the serial interrupt mechanism is enabled.

ESO: Enable Serial output

The ESO flag enables/disables the serial I/O interface: ESO = '1' enables, ESO = '0' disables. ESO can only be written by software.

BC0, BC1 and BC2

Bits BC0, BC1 and BC2 indicate the number of bits received or transmitted in a data stream. These bits can only be written by software.

AL: Arbitration Lost

The arbitration lost flag is set by hardware when the serial I/O interface, as master transmitter, loses a bus arbitration procedure.

AAS: Addressed As Slave

This flag is set by hardware when the interface detects either its own specific address or the 'general CALL' address as the first byte of a transfer and the interface has been programmed to operate in the address recognition mode.

AD0: Address Zero

This flag is set by hardware after detection of the 'general CALL' address when the interface is operating in the address recognition mode.

LRB: Last Received Bit

This contains either the last data bit received or, for a transmitting device in the acknowledgement mode, the acknowledgement signal from the receiving device.

Bits AL, AAS, AD0 and LRB can only be read by software.

Serial clock control word (S2)

Bits 0 to 4 of the clock control register S2 are used to set the frequency of the serial clock signal. When a 6 MHz crystal is used, the frequency of the serial clock can be varied between 154 kHz and 1 kHz (see Table 2). An asymmetrical clock with a HIGH-to-LOW ratio of 3 : 1 can be generated using bit 5. The asymmetrical clock allows a microcontroller more time per clock period for sampling the data line, making the timing of this action less critical. Bit 6 can be used to activate the acknowledge mode of the serial I/O. S2 is a write only register.

Address register

The address register contains the 7-bit address back-up latches and the bit (ALS) used to enable/disable the address recognition mode. The address register can be written using the MOV S0, A and MOV S0, # data instructions, but only when ES0 = '0'.

Serial I/O interrupt logic

An EN SI instruction enables and a DIS SI instruction disables the interrupt logic. When the logic is enabled, a pending interrupt results in a serial I/O interrupt to the processor, causing a CALL to location 5 in the ROM. When disabled, the presence of an interrupt is still indicated by PIN in S1, allowing the interrupt to be serviced. However, vectored interrupt will not occur.

FUNCTIONAL DESCRIPTION (continued)

Table 2 SIO clock pulse frequency control when using a 6 MHz and a 10 MHz crystal

hexadecimal S20-S24 code	divisor	f_{XTAL} 6 MHz) f_{SCLK} (kHz) ▲	f_{XTAL} (10 MHz) f_{SCLK} (kHz) ▲
0	not allowed		
1	39	*154	*256
2	45	*133	*222
3	51	*118	*196
4	63	95	*159
5	75	80	*133
6	87	69	*115
7	99	61	*101
8	123	49	81
9	147	41	68
A	171	35	58
B	195	31	51
C	243	25	41
D	291	21	34
E	339	18	29
F	387	16	26
10	483	12	21
11	579	10	17
12	675	8,9	15
13	771	7,8	13,4
14	963	6,2	10,4
15	1155	5,2	8,7
16	1347	4,5	7,4
17	1539	3,9	6,5
18	1923	3,1	5,2
19	2307	2,6	4,3
1A	2691	2,2	3,7
1B	3075	2,0	3,3
1C	3843	1,6	2,6
1D	4611	1,3	2,2
1E	5379	1,1	1,9
1F	6147	1,0	1,6

* Not permitted for I^2C operation.

▲ The maximum clock frequency in the I^2C systems is 100 kHz.

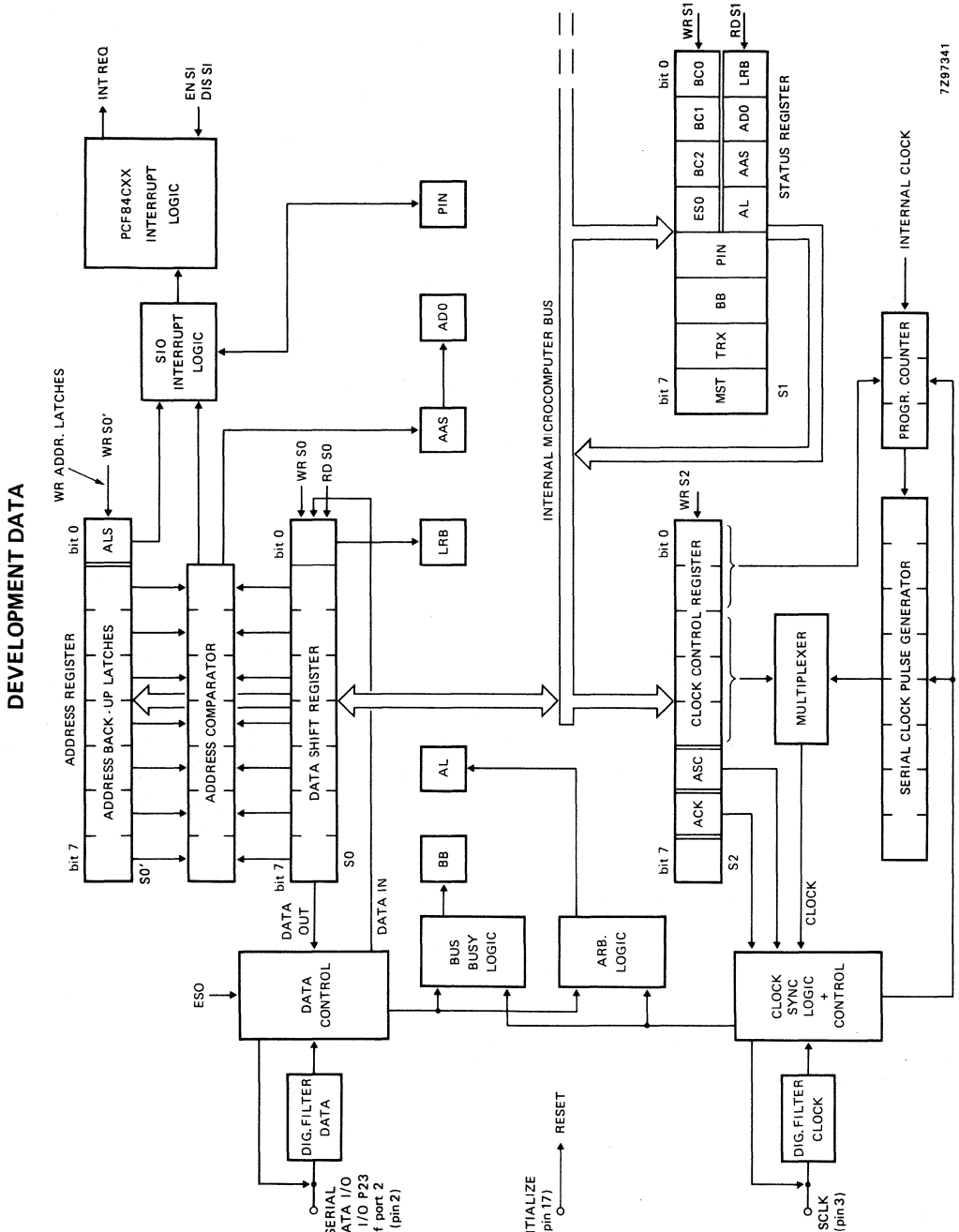


Fig. 15 Serial I/O interface.

FUNCTIONAL DESCRIPTION (continued)

Interrupts (see Fig. 16)

When the external interrupt is enabled, a HIGH-to-LOW transition on the $\overline{INT}$/T0 input initiates an external interrupt subroutine which causes a CALL to program memory location 3 following completion of the current instruction.
The interrupt must remain enabled until the interrupt instruction is completed, otherwise the next instruction of the main program will be executed. Serial I/O interrupt, when enabled, causes a CALL to location 5, and a timer/event counter overflow forces a CALL to location 7 when the timer interrupt is enabled.

When an interrupt subroutine starts, the contents of the program counter and bits 4, 6 and 7 of the PSW have been saved in the program counter stack. Accumulator contents have to be saved by software. Interrupt acknowledgement can be carried out by software via port pins. All interrupt subroutines must reside in memory bank 0.

Since the interrupt system is single level, once an interrupt is detected, all further interrupt requests are latched, but ignored, pending a RETR instruction to re-enable the interrupt input logic. After executing RETR, the program continues in the main part; this is independent of the occurance of a second interrupt during the running of the first routine. If 2 or 3 interrupts occur simultaneously, their priority is:

- (1) external
- (2) serial I/O
- (3) timer/event counter

An external interrupt can be generated by using the timer/counter in the event counter mode. The counter is first preset to (FFH), then EN TCNTI instruction is executed. A LOW-to-HIGH transition of the T1 input will then initiate an interrupt subroutine and cause a CALL to location 7.

On execution of a DIS I instruction, the PCF84CXX always clears the digital filter/latch and the External Interrupt Flag.

The Timer Flag (TF) is reset only when the JTF or JNTF instruction is executed or after RESET.

The Timer Interrupt Flag is set when timer overflow occurs, only if the timer interrupt is enabled.

The microcontroller will exit the IDLE mode when any one of the following three interrupts is enabled:

- External
- Serial I/O
- Timer/event counter

There is no internal pull-up or pull-down device connected to the external interrupt input (pin 12). If required pin 12 must be externally connected to a resistor ($R \leqslant 100\ k\Omega$). When the external interrupt is not used pin 12 must be connected to V_{DD}.

Improvements to interrupt and timer logic with respect to the MAB8400 family

For detailed information see the PCF84CXX section in the user manual "Single-chip 8-bit microcontrollers".

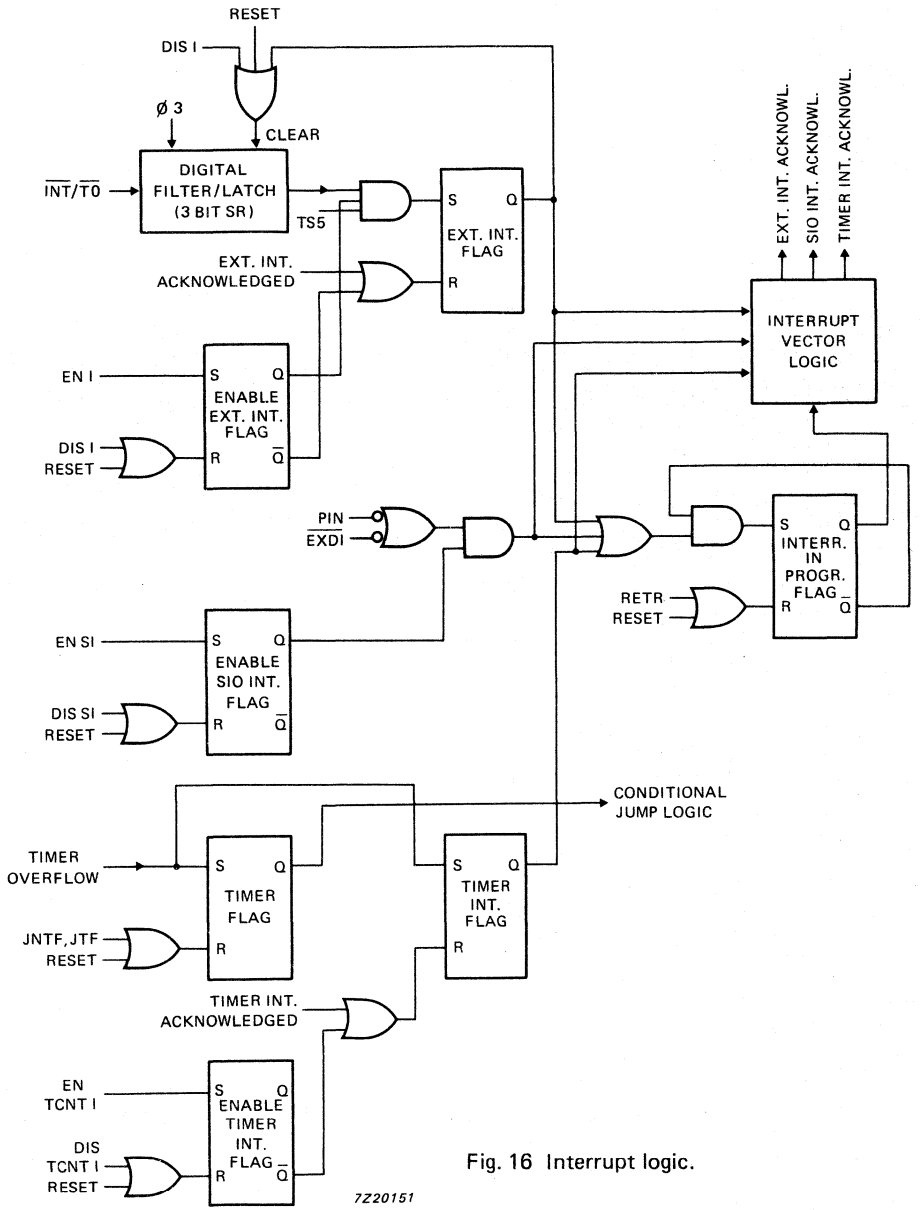


Fig. 16 Interrupt logic.

Notes to figure 16

1. $\overline{INT}$/T0 negative edge is always latched in the digital filter/latch.
2. Correct interrupt timing is ensured when $\overline{INT}$/T0 is HIGH for $>$ 4 CP followed by a LOW for $>$ 7 CP.
3. When the interrupt in progress flag is set, further external and timer interrupts are latched but ignored, until RETR is executed.
4. A DIS I instruction always clears a pending external interrupt.
5. For all flip-flops, RESET overrules SET.
6. EXDI is an input of PCF8400T, in derivative versions it is connected internally to a logic '1'.

FUNCTIONAL DESCRIPTION (continued)

Oscillator (see Fig. 17)

The oscillator can be inhibited by the STOP instruction under software control. It is also inhibited when a low-supply voltage condition is present to prevent discharge of a weak back-up battery. Provided the supply voltage is within the operating range the oscillator will be restarted after a STOP instruction by a LOW level at the $\overline{INT}$/T0 pin or a HIGH level at the RESET pin.

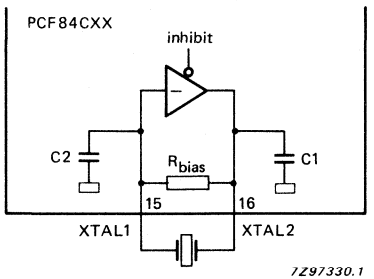


Fig. 17 Oscillator with integrated elements.

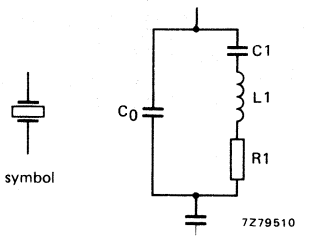


Fig. 18 Crystal unit equivalent circuit.

The values of crystal series resistance R1 and the crystal's total load capacitance C_L (C_0 + wiring + external capacitors) must not be above the curve (Fig. 19) for the corresponding frequency.
Note; if external capacitors are connected to XTAL 1 and XTAL they must be of equal value.

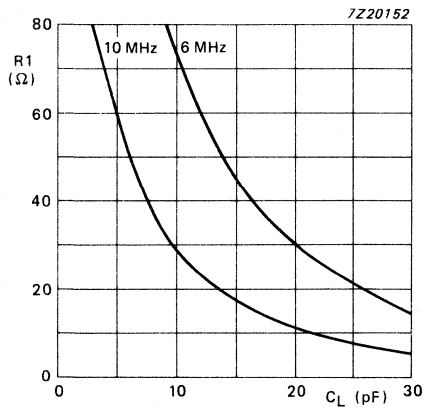


Fig. 19 Crystal circuit criteria.

The oscillator has the output drive capability via pin 16 (XTAL2). An external clock can be applied to pin 15 (XTAL1). A machine cycle consists of 10 time slots, each time slot being 3 oscillator periods.

Timer/event counter (see Fig. 20)

An internal 8-bit up-counter is provided. This can count external events, modulo-32 machine cycles, or machine cycles directly. Table 3 gives the instructions that control the counter and the prescaler, and the functions performed.

When used as a timer, the input to the counter is either the overflow or input of a 5-bit prescaler. When used as an event counter, LOW-to-HIGH transitions on pin 13 (T1) are counted. The maximum rate at which the counter may be incremented is once every machine cycle. When the counter overflows, the timer flag is set. The flag can be tested and reset using the JTF (jump is timer flag = 1) or JNTF instruction. Overflow also generates an interrupt to the processor via setting of the Timer Interrupt Flag when the timer/event counter interrupt is enabled.

Table 3 Timer/event counter control

function	timer mode modulo-1, modulo-32*	counter mode
CLEAR	MOV T,A (A) = 0 or RESET	MOV T,A (A) = 0 or RESET
PRESET	MOV T,A	MOV T,A
START	STRT T	STRT CNT
STOP	STOP TCNT or RESET	STOP TCNT or RESET
TEST	JTF/JNTF	JTF/JNTF
READ**	MOV A,T	MOV A,T

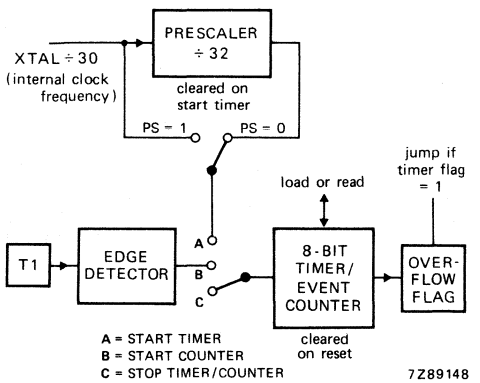


Fig. 20 Timer/event counter.

DEVELOPMENT DATA

Program status word (see Fig. 21)

The program status word (PSW) is an 8-bit word (1 byte) in the CPU which stores information about the current status of the microcontroller.
The PSW bits are:

- Bits 0 to 2 — stack pointer bits (SP_0, SP_1, SP_2)
- Bit 3 — prescaler select (PS);
  0 = modulo-32; 1 = modulo-1 (no prescaling)
- Bit 4 — working register bank select (RBS);
  0 = register bank 0; 1 = register bank 1
- Bit 5 — not used (1)
- Bit 6 — auxiliary carry (AC); half-carry bit generated by an ADD instruction and used by the decimal adjust instruction DA A
- Bit 7 — carry (CY); the carry flag indicates that previous operation has resulted in an overflow of the accumulator.

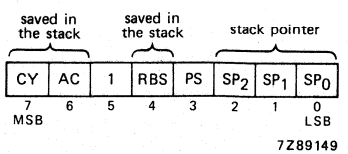


Fig. 21 Program status word.

* With prescaler select, PS = 0, the timer counts modulo-32 machine cycles, with PS = 1 it counts modulo-1 cycles (prescaler not used); prescaler cleared with STRT T, prescaler not readable.

** READ does not disturb the counting process.

FUNCTIONAL DESCRIPTION (continued)

Program status word (continued)

All bits can be read using the MOV A, PSW instruction. Bits 7 and 6 are set and cleared by CPU operation. Bit 4 can be changed by a SEL RB instruction, bit 3 by the MOV PSW, A instruction, and bits 0, 1 and 2 by the CALL, RET or RETR instructions and in the event of an interrupt. Bits 7, 6 and 4 are stored in the program counter stack during subroutine and interrupt calls. These bits are restored in the PSW with a RETR (return and restore) instruction which must be used at the end of an interrupt and can be used at the end of a normal subroutine. The RET instruction has no restore feature and cannot be used at the end of an interrupt.

Program counter (see Fig. 22)

The 13-bit program counter is able to address 8 K bytes of ROM. The arrangement of the bits is shown in figure 22. During an interrupt subroutine PC_{11} and PC_{12} are forced to logic 0. All 13 bits are saved in the stack during CALL and interrupt routines.

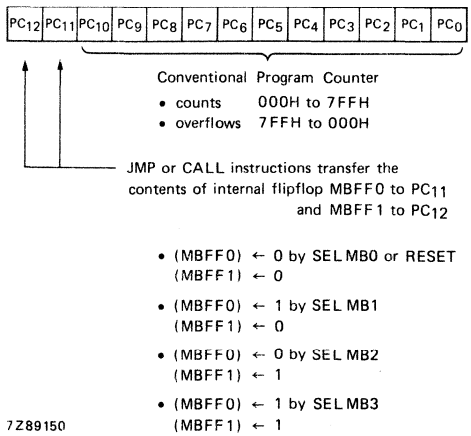


Fig. 22 Program counter.

Central processing unit

The PCF84CXX has arithmetic, logical and branching capabilities. The DA A, SWAP A and XCHD instructions simplify BCD arithmetic and the handling of nibbles. The MOVP A,@A instruction permits efficient table look-up from the current ROM page.

Conditional branch logic

The conditional branch logic within the processor enables several conditions, internal and external to the processor, to be tested by the user's program. Table 4 lists the conditional jump instructions used to change the program sequence. The DJNZ instruction decrements a designated register or data memory location and branches if the contents are not zero. This instruction is useful for looping control. The JMPP@A instruction allows multiway branches to destinations determined by the contents of the accumulator.

Table 4 Conditional branches

test	jump condition	jump instruction
accumulator	all bits zero	JZ
	any bit non-zero	JNZ
accumulator bit test	1	JB0 to JB7
carry flag	1	JC
	0	JNC
timer overflow flag	1	JTF
	0	JNTF
test input T0	1	JT0
	0	JNT0
test input T1	1	JT1
	0	JNT1
register	non-zero	DJNZ

Test input T1 (pin 13)

The T1 input line can be used as:

- A test input for branch instructions JT1 and JNT1
- An external input to the event counter

When used as a test input:

- JT1 instruction tests for logic 1 level
- JNT1 instruction tests for logic 0 level

When used as an input to the event counter, T1 must be LOW for > 4 CP, followed by a HIGH for > 4 CP. The transition can be recognized with a repetition rate of once per 30 oscillator clock periods (1 machine cycle).

There is no internal pull-up or pull-down resistor connected to the T1 input. If required it must be externally connected to a resistor ($R = \leqslant 100\ k\Omega$). When T1 is not used pin 13 must be connected to V_{DD} or V_{SS}.

Reset (pin 17)

A positive-going signal on the RESET input

- Sets the program counter to zero
- Selects location 0 of memory bank 0 and register bank 0
- Sets the stack pointer to zero (000); pointing to RAM address 8
- Disables the interrupts (external, timer and serial I/O)
- Stops the timer/event counter, then sets it to zero
- Sets the timer prescaler to modulo-32
- Resets the timer flag
- Sets all ports to input mode
- Sets the serial I/O to slave receiver mode and disables the serial I/O
- Cancels IDLE and STOP mode

FUNCTIONAL DESCRIPTION (continued)

Power-on-reset

The internal power-on reset circuit monitors the PCF84CXX supply voltage V_{DD}. For as long as the supply voltage remains below the internal reference level V_{ref} (typically 1,5 V) the oscillator is inhibited and RESET (pin 17) has an undefined level. When V_{DD} rises above the internal reference level, the oscillator is released and RESET is pulled high to V_{DD} by TR1 for a period t_D (typically 50 μs).

N.B. Because of the narrow bandwidth of the crystal, the start-up time of the oscillator is typically 10 ms.

Three modes of power-on reset are possible:

1. If V_{DD} can be switched with a fast rise time i.e. V_{DD} reaches its minimum operating value (corresponding to the selected oscillator frequency) before the RESET signal (t_D) has finished, then no extra components are required (see Fig. 23 and 24). Note that the first instruction is executed after the oscillator start-up time plus 1866 clock periods have elapsed.

2. If V_{DD} has a slow rise time then the RESET signal should be stretched by an external RC circuit (see Fig. 25 and 26). In the event of a short drop in the supply voltage, the diode path rapidly discharges the capacitor to ensure a reliable power-on reset. To ensure a correct reset, the RESET signal should reach at least 70% of the final value of V_{DD}. Given that the RESET voltage and V_{DD} rise exponentially, the above requirement is satisfied when the time constant τ of the RESET pulse is >8 times the time constant of V_{DD}. If V_{DD} rises linearly, then a RESET time constant >2 times the rise time of V_{DD} is required.

   When a reset is completed (RESET goes LOW) before the oscillator has started up, program execution begins after the oscillator start-up time plus 1866 clock periods have elapsed (see Fig. 26). If the oscillator is started-up prior to the completion of RESET, then program execution begins 1866 clock periods after RESET goes LOW.

3. Figure 27 shows an external reset to the PCF84CXX during power-on. The external reset signal must remain HIGH until V_{DD} has reached its minimum operating value corresponding to the selected oscillator frequency. When a reset is completed (RESET goes LOW) before the oscillator has started up, program execution begins after the oscillator start-up time plus 1866 clock periods have elapsed (see Fig. 28). If the oscillator is started-up prior to the completion of RESET, then program execution begins 1866 clock periods after RESET goes LOW.

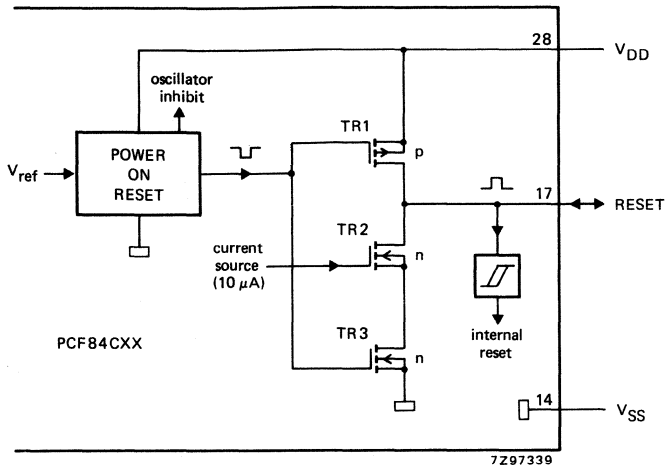


Fig. 23 Power-on-reset configuration.

DEVELOPMENT DATA

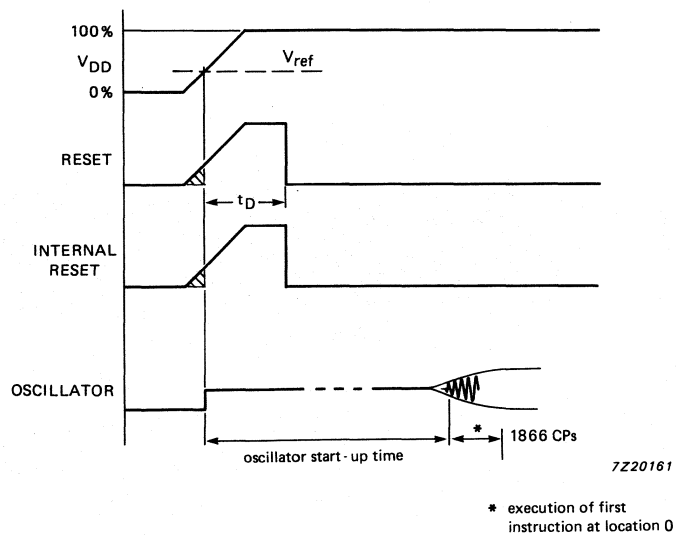


Fig. 24 Timing of power-on-reset with fast rise time.

FUNCTIONAL DESCRIPTION (continued)

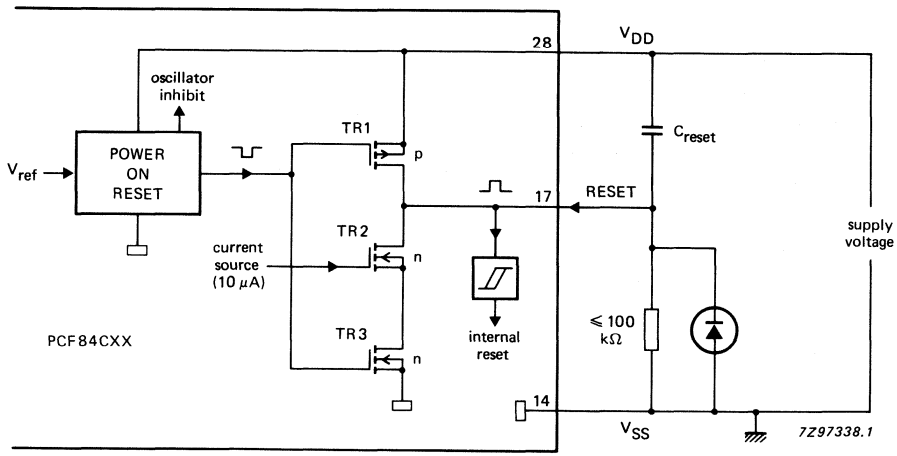


Fig. 25 Stretched power-on-reset with external components.

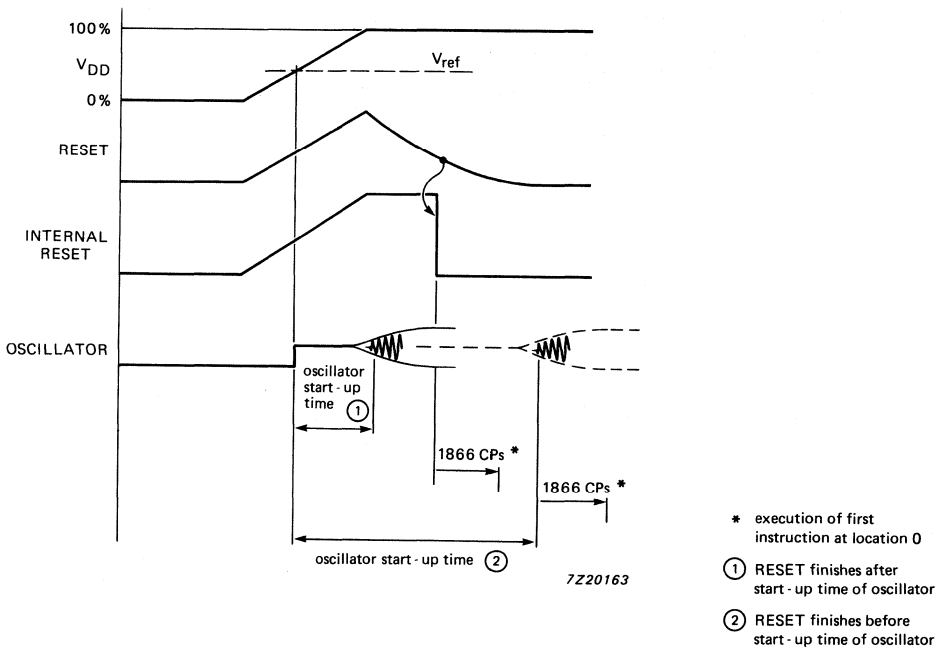


Fig. 26 Timing of power-on-reset with a slowly rising V_{DD} and a stretched RESET pulse.

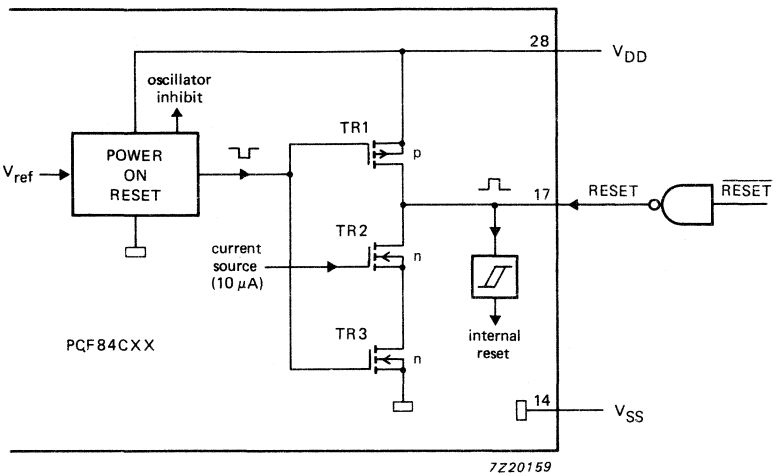


Fig. 27 External power-on-reset configuration.

DEVELOPMENT DATA

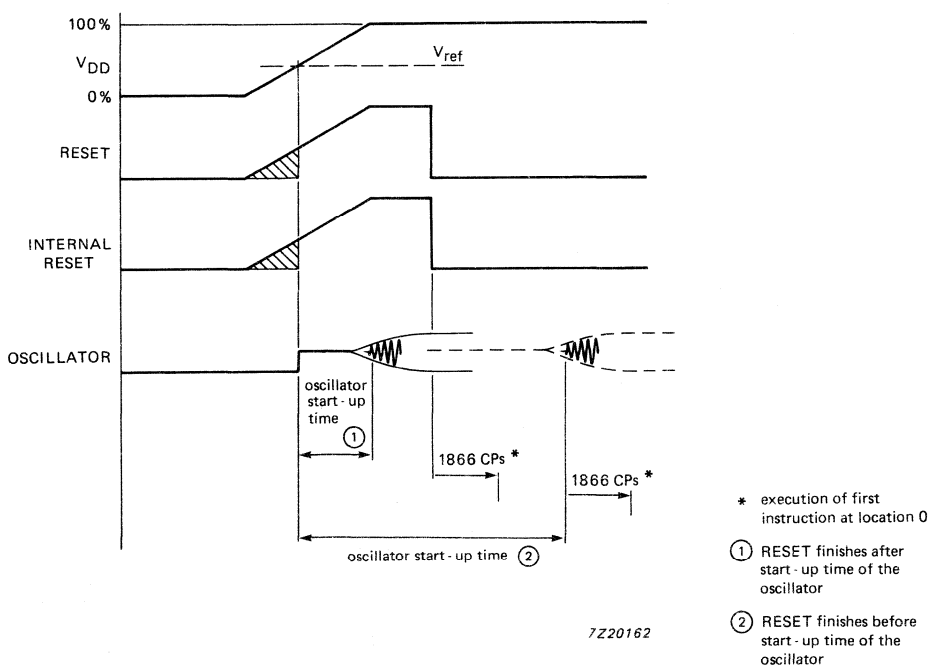


Fig. 28 Timing of external power-on-reset.

INSTRUCTION SET

The PCF84CXX instruction set consists of over 80 one and two byte instructions and is identical to the MAB8400 instruction set. New instructions are added for STOP and IDLE mode. Program code efficiency is high because all RAM locations and all ROM locations on a 256 byte page require only a single byte address.

Table 7 gives the instruction set of the PCF84CXX. Table 6 shows the instruction map and Table 5 details the symbols and definition descriptions that are used.

Table 5 Symbols and definitions used in Table 7

symbol	definition description
A	accumulator
addr	program memory address
Bb	bit designation (b = 0-7)
RBS	register bank select
C	carry bit (bit CY)
CNT	event counter
Dx	mnemonic derivative register (x = 0 ... 255)
data	8-bit number or expression
I	interrupt
MB	memory bank
MBFF	memory bank flip-flop
P	mnemonic for 'in-page' operation
PC	program counter
Pp	port designation (p = 0, 1 or 2)
PSW	program status word
RB	register bank
Rr	register designation (r = 0-7)
Sn	serial I/O register
SP	stack pointer
T	timer
TF	timer flag
T0, T1	test 0 and 1 inputs
#	immediate data prefix
@	indirect address prefix
(X)	contents of X
((X))	contents of location addressed by X
←	is replaced by
↔	is exchanged with

Table 6 PCF84CXX instruction map

first hexadecimal character of opcode ↓ / second hexadecimal character of opcode →	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NOP	IDLE		ADD A,#data	JMP page 0	EN I	JNTF addr	DEC A	IN A,Pp 0 1 2				MOV A,Sn 0 1			
1	INC @Rr 0 1		JBO addr	ADDC A,#data	CALL page 0	DIS I	JTF addr	INC A	INC Rr 0 1 2 3 4 5 6 7							
2	XCH A,@Rr 0 1		STOP	MOV A,#data	JMP page 1	EN TCNTI	JNTO addr	CLR A	XCH A,Rr 0 1 2 3 4 5 6 7							
3	XCHD A,@Rr 0 1		JB1 addr		CALL page 1	DIS TCNTI	JTO addr	CPL A	OUTL Pp,A 0 1 2				MOV Sn,A 0 1 2			
4	ORL A,@Rr 0 1		MOV A,T	ORL A,#data	JMP page 2	STRT CNT	JNT1 addr	SWAP A	ORL A,Rr 0 1 2 3 4 5 6 7							
5	ANL A,@Rr 0 1		JB2 addr	ANL A,#data	CALL page 2	STRT T	JT1 addr	DA A	ANL A,Rr 0 1 2 3 4 5 6 7							
6	ADD A,@Rr 0 1		MOV T,A		JMP page 3	STOP TCNT		RRC A	ADD A,Rr 0 1 2 3 4 5 6 7							
7	ADDC A,@Rr 0 1		JB3 addr		CALL page 3			RR A	ADDC A,Rr 0 1 2 3 4 5 6 7							
8				RET	JMP page 4	EN SI			ORL Pp,#data 0 1 2				MOV A,Dx	MOV Dx,A	ANL Dx,A	ORL Dx,A
9			JB4 addr	RETR	CALL page 4	DIS SI	JNZ addr	CLR C	ANL Pp,#data 0 1 2				MOV Sn,#data 0 1 2			
A	MOV @Rr,A 0 1			MOVP A,@A	JMP page 5	SEL MB2		CPL C	MOV Rr,A 0 1 2 3 4 5 6 7							
B	MOV @Rr,#data 0 1		JB5 addr	JMPP @A	CALL page 5	SEL MB3			MOV Rr,#data 0 1 2 3 4 5 6 7							
C	DEC @Rr 0 1				JMP page 6	SEL RB0	JZ addr	MOV A,PSW	DEC Rr 0 1 2 3 4 5 6 7							
D	XRL A,@Rr 0 1		JB6 addr	XRL A,#data	CALL page 6	SEL RB1		MOV PSW,A	XRL A,Rr 0 1 2 3 4 5 6 7							
E	DJNZ @Rr,addr 0 1				JMP page 7	SEL MB0	JNC addr	RL A	DJNZ Rr, addr 0 1 2 3 4 5 6 7							
F	MOV A,@Rr 0 1		JB7 addr		CALL page 7	SEL MB1	JC addr	RLC A	MOV A,Rr 0 1 2 3 4 5 6 7							

INSTRUCTION SET (continued)

Table 7 Instruction set

	mnemonic	opcode (hex.)	bytes/ cycles	description	function		notes
ACCUMULATOR	ADD A, Rr	6*	1/1	Add register contents to A	(A)←(A) + (Rr)	r = 0–7	1
	ADD A, @Rr	60 61	1/1	Add RAM data, addressed by Rr, to A	(A)←(A) + ((R0)) (A)←(A) + ((R1))		1
	ADD A, #data	03 data	2/2	Add immediate data to A	(A)←(A) + data		1
	ADDC A, Rr	7*	1/1	Add carry and register contents to A	(A)←(A) + (Rr) + (C)	r = 0–7	1
	ADDC A, @Rr	70 71	1/1	Add carry and RAM data, addressed by Rr, to A	(A)←(A) + ((R0)) + (C) (A)←(A) + ((R1)) + (C)		1
	ADDC A, #data	13 data	2/2	Add carry and immediate data to A	(A)←(A) + data + (C)		1
	ANL A, Rr	5*	1/1	'AND' Rr with A	(A)←(A) AND (Rr)	r = 0–7	
	ANL A, @Rr	50 51	1/1	'AND' RAM data, addressed by Rr, with A	(A)←(A) AND ((R0)) (A)←(A) AND ((R1))		
	ANL A, #data	53 data	2/2	'AND' immediate data with A	(A)←(A) AND data		
	ORL A, Rr	4*	1/1	'OR' Rr with A	(A)←(A) OR (Rr)	r = 0–7	
	ORL A, @Rr	40 41	1/1	'OR' RAM data, addressed by Rr, with A	(A)←(A) OR ((R0)) (A)←(A) OR ((R1))		
	ORL A, #data	43 data	2/2	'OR' immediate data with A	(A)←(A) OR data		
	XRL A, Rr	D*	1/1	'XOR' Rr with A	(A)←(A) XOR (Rr)	r = 0–7	
	XRL A, @Rr	D0 D1	1/1	'XOR' RAM, addressed by Rr, with A	(A)←(A) XOR ((R0)) (A)←(A) XOR ((R1))		
	XRL A, #data	D3 data	2/2	'XOR' immediate data with A	(A)←(A) XOR data		
	INC A	17	1/1	increment A by 1	(A)←(A) + 1		
	DEC A	07	1/1	decrement A by 1	(A)←(A) −1		
	CLR A	27	1/1	clear A to zero	(A)←0		
	CPL A	37	1/1	one's complement A	(A)←NOT(A)		
	RL A	E7	1/1	rotate A left	$(A_{n+1})\leftarrow(A_n)$ $(A_0)\leftarrow(A_7)$	n = 0–6	

DEVELOPMENT DATA

ACCUMULATOR (cont.)	RLC A	F7	1/1	rotate A left through carry	$(A_n + 1) \leftarrow A_n$ $(A_0) \leftarrow (C), (C) \leftarrow (A_7)$	n = 0–6	2
	RR A	77	1/1	rotate A right	$(A_n) \leftarrow (A_n + 1)$ $(A_7) \leftarrow (A_0)$	n = 0–6	
	RRC A	67	1/1	rotate A right through carry	$(A_n) \leftarrow (A_n + 1)$ $(A_7) \leftarrow (C), (C) \leftarrow (A_0)$	n = 0–6	2
	DA A	57	1/1	decimal adjust A			2
	SWAP A	47	1/1	swap nibbles of A	$(A_{4-7}) \leftrightarrow (A_{0-3})$		
DATA MOVES	MOV A, Rr	F*	1/1	move register contents to A	(A)←(Rr)	r = 0–7	
	MOV A, @Rr	F0 F1	1/1	move RAM data, addressed by Rr, to A	(A)←((R0)) (A)←((R1))		
	MOV A, #data	23 data	2/2	move immediate data to A	(A)←data		
	MOV Rr, A	A*	1/1	move accumulator contents to register	(Rr)←(A)	r = 0–7	
	MOV @Rr, A	A0 A1	1/1	move accumulator contents to RAM location addressed by Rr	((R0))←(A) ((R1))←(A)		
	MOV Rr, #data	B* data	2/2	move immediate data to Rr	(Rr)←data		
	MOV @Rr, #data	B0 data B1 data	2/2	move immediate data to RAM location addressed by Rr	((R0))←data ((R1))←data		
	XCH A, Rr	2*	1/1	exchange accumulator contents with Rr	(A)↔(Rr)	r = 0–7	
	XCH A, @Rr	20 21	1/1	exchange accumulator contents with RAM data addressed by Rr	(A)↔((R0)) (A)↔((R1))		
	XCHD A, @Rr	30 31	1/1	exchange lower nibbles of A and RAM data addressed by Rr	$(A_{0-3}) \leftrightarrow ((R0_{0-3}))$ $(A_{0-3}) \leftrightarrow ((R1_{0-3}))$		
	MOV A, PSW	C7	1/1	move PSW contents to accumulator	(A)←(PSW)		
	MOV PSW, A	D7	1/1	move accumulator bit 3 to PSW_3	$(PSW_3) \leftarrow (A_3)$		3
	MOVP A, @A	A3	1/2	move indirectly addressed data in current page to A	$(PC_{0-7}) \leftarrow (A), (A) \leftarrow ((PC))$		
FLAGS	CLR C	97	1/1	clear carry bit	(C)←0		2
	CPL C	A7	1/1	complement carry bit	(C)←NOT(C)		2

INSTRUCTION SET (continued)

	mnemonic	opcode (hex.)	bytes/ cycles	description	function	notes
REGISTER	INC Rr	1*	1/1	increment register by 1	(Rr)←(Rr) + 1 r = 0–7	
	INC @Rr	10 11	1/1	increment RAM data, addressed by Rr, by 1	((R0))←((R0)) + 1 ((R1))←((R1)) + 1	
	DEC Rr	C*	1/1	decrement register by 1	(Rr)←(Rr) − 1 r = 0–7	
	DEC @Rr	C0 C1	1/1	decrement RAM data, addressed by Rr, by 1	((R0))←((R0)) − 1 ((R1))←((R1)) − 1	
BRANCH	JMP addr	● 4 address	2/2	unconditional jump within a 2 K bank	(PC_{8-10})←$addr_{8-10}$ (PC_{0-7})←$addr_{0-7}$ (PC_{11-12})←MBFF 0–1	
	JMPP @A	B3	1/2	indirect jump within a page	(PC_{0-7})←((A))	
	DJNZ Rr, addr	E* address	2/2	decrement Rr by 1 and jump if not zero to addr	(Rr)←(Rr) − 1 r = 0–7 if (Rr) not zero (PC_{0-7})←addr	
	DJNZ @Rr, addr	E0	2/2	decrement RAM data, addressed by Rr by 1 and jump if not zero to addr	((R0))←((R0)) − 1 if ((R0)) not zero (PC_{0-7})←addr	
		E1			((R1))←((R1)) − 1 if ((R1)) not zero (PC_{0-7})←addr	
	JBb addr	▲ 2 address	2/2	jump to addr if Acc. bit b = 1	if b = 1 : (PC_{0-7})←addr b = 0–7	
	JC addr	F6 address	2/2	jump to addr if C = 1	if C = 1 : (PC_{0-7})←addr	
	JNC addr	E6 address	2/2	jump to addr if C = 0	if C = 0 : (PC_{0-7})←addr	
	JZ addr	C6 address	2/2	jump to addr if A = 0	if A = 0 : (PC_{0-7})←addr	
	JNZ addr	96 address	2/2	jump to addr if A is NOT zero	if A≠0 : (PC_{0-7})←addr	
	JT0 addr	36 address	2/2	jump to addr if T0 = 1	if T0 = 1: (PC_{0-7})←addr	
	JNT0 addr	26 address	2/2	jump to addr if T0 = 0	if T0 = 0: (PC_{0-7})←addr	
	JT1 addr	56 address	2/2	jump to addr if T1 = 1	if T1 = 1: (PC_{0-7})←addr	
	JNT1 addr	46 address	2/2	jump to addr if T1 = 0	if T1 = 0: (PC_{0-7})←addr	
	JTF addr	16 address	2/2	jump to addr if Timer Flag = 1	if TF = 1: (PC_{0-7})←addr	4
	JNTF addr	06 address	2/2	jump to addr if Timer Flag = 0	if TF = 0: (PC_{0-7})←addr	

TIMER/EVENT COUNTER	MOV A, T	42	1/1	move timer/event counter contents to accumulator	(A)←(T)	
	MOV T, A	62	1/1	move accumulator contents to timer/event counter	(T)←(A)	
	STRT CNT	45	1/1	start event counter		
	STRT T	55	1/1	start timer		
	STOP TCNT	65	1/1	stop timer/event counter		
	EN TCNTI	25	1/1	enable timer/event counter interrupt		
	DIS TCNTI	35	1/1	disable timer/event counter interrupt		
CONTROL	EN I	05	1/1	enable external interrupt		
	DIS I	15	1/1	disable external interrupt		
	SEL RB0	C5	1/1	select register bank 0	(RBS)←0	5
	SEL RB1	D5	1/1	select register bank 1	(RBS)←1	5
	SEL MB0	E5	1/1	select program memory bank 0	(MBFF0)←0, (MBFF1)←0	
	SEL MB1	F5	1/1	select program memory bank 1	(MBFF0)←1, (MBFF1)←0	
	SEL MB2	A5	1/1	select program memory bank 2	(MBFF0)←0, (MBFF1)←1	
	SEL MB3	B5	1/1	select program memory bank 3	(MBFF0)←1, (MBFF1)←1	
	STOP	22	1/1	enter STOP mode		
	IDLE	01	1/1	enter IDLE mode		
SUBROUTINE	CALL addr	▲ 4 address	2/2	jump to subroutine	((SP))←(PC), ($PSW_{4, 6, 7}$) 6 (SP)←(SP) + 1 (PC_{8-10})←$addr_{8-10}$ (PC_{0-7})←$addr_{0-7}$ (PC_{11-12})←MBFF 0–1	6
	RET	83	1/2	return from subroutine	(SP)←(SP) – 1 (PC)←((SP))	6
	RETR	93	1/2	return from interrupt and restore bits 4, 6, 7 of PSW	(SP)←(SP) – 1 ($PSW_{4, 6, 7}$) + (PC)←((SP))	6

INSTRUCTION SET (continued)

	mnemonic	opcode (hex.)	bytes/ cycles	description	function		notes
PARALLEL INPUT/OUTPUT	IN A, Pp	08 09 0A	1/2	input port p data to accumulator	(A)←(P0) (A)←(P1) (A)←(P2)		7
	OUTL Pp, A	38 39 3A	1/2	output accumulator data to port p	(P0)←(A) (P1)←(A) (P2)←(A)		
	ANL Pp, #data	98 data 99 data 9A data	2/2	AND port p data with immediate data	(P0)←(P0) AND data (P1)←(P1) AND data (P2)←(P2) AND data		
	ORL Pp, #data	88 data 89 data 8A data	2/2	OR port p data with immediate data	(P0)←(P0) OR data (P1)←(P1) OR data (P2)←(P2) OR data		
DERIVATIVE INPUT/OUTPUT	MOV A, Dx	8C	2/2	move derivative register contents to accumulator	(A)←(Dx)	x = 0 to 255	8
	MOV Dx, A	8D	2/2	move accumulator contents to derivative register	(Dx)←(A)	x = 0 to 255	8
	ANL Dx, A	8E	2/2	AND derivative register with accumulator	(Dx)←(Dx) AND (A)	x = 0 to 255	8
	ORL Dx, A	8F	2/2	OR derivative register with accumulator	(Dx)←(Dx) OR (A)	x = 0 to 255	8

DEVELOPMENT DATA

	mnemonic	opcode (hex.)	bytes/ cycles	description	function	notes
SERIAL INPUT/OUTPUT	MOV A, S_n	0C 0D	1/2	move serial I/O register contents to accumulator	(A)←(S0) (A)←(S1)	9
	MOV S_n, A	3C 3D 3E	1/2	move accumulator contents to serial I/O register	(S0)←(A) (S1)←(A) (S2)←(A)	
	MOV S_n, #data	9C data 9D data 9E data	2/2	move immediate data to serial I/O register	(S0)←data (S1)←data (S2)←data	
	EN SI	85	1/1	enable serial I/O interrupt		
	DIS SI	95	1/1	disable serial I/O interrupt		
	NOP	00	1/1	no operation		

Notes to Table 8

1. PSW CY, AC affected
2. PSW CY affected
3. PSW PS affected
4. Execution of JTF and JNTF instructions resets the Timer Flag (TF).

* : 8, 9, A, B, C, D, E, F
● : 0, 2, 4, 6, 8, A, C, E
▲ : 1, 3, 5, 7, 9, B, D, F

5. PSW RBS affected
6. PSW SP_0, SP_1, SP_2 affected
7. (A) = 0000P23, P22, P21, P20.
8. Instructions for PCF84C00 only.
9. (S1) has a different meaning for read and write operation, see serial I/O interface.

RATINGS

Limiting values in accordance with the Absolute Maximum System (IEC 134)

Supply voltage (pin 28)	V_{DD}		−0,8 to +8	V
All input voltages	V_I		−0,5 to V_{DD} +0,5	V
D.C. current into any input or output	$\pm I_I, \pm I_O$	max.	10	mA
Total power dissipation (see note)	P_{tot}	max.	500	mW
DIL-28 (SOT-177)	P_{tot}	max.	1	W
SO-28 (SOT-136A)	P_{tot}	max.	0,6	W
Storage temperature range	T_{stg}		−65 to +150	°C
Operating ambient temperature range	T_{amb}		−40 to +85	°C
Operating junction temperature	T_j	max.	125	°C

HANDLING

Inputs and outputs are protected against electrostatic discharge in normal handling. However, to be totally safe, it is desirable to take normal precautions appropriate to handling MOS devices (see 'Handling MOS devices').

Note

Thermal resistance (junction to ambient)

for SOT-117D	$R_{th\ j\text{-}a}$	max.	120	K/W
for SOT-135A	$R_{th\ j\text{-}a}$	max.	60	K/W
for SOT-136A	$R_{th\ j\text{-}a}$	max.	150	K/W

DEVELOPMENT DATA

D.C. CHARACTERISTICS

V_{DD} = 2,5 to 5,5 V; V_{SS} = 0 V; T_{amb} = −40 to +85 °C; all voltages with respect to V_{SS}; unless otherwise specified.

parameter	symbol	min.	typ.	max.	unit
Supply voltage					
operating (see Fig. 33)	V_{DD}	2,5	–	5,5	V
Supply current					
operating (see Fig. 34; not valid for PCF84C00)					
at V_{DD} = 5 V; f_{XTAL} = 10 MHz	I_{DD}	–	1,6	3,2	mA
at V_{DD} = 5 V; f_{XTAL} = 6 MHz	I_{DD}	–	1	2	mA
at V_{DD} = 3 V; f_{XTAL} = 3,58 MHz	I_{DD}	–	0,3	0,6	mA
IDLE mode (see Fig. 35; not valid for PCF84C00)					
at V_{DD} = 5 V; f_{XTAL} = 10 MHz	I_{DD}	–	0,8	1,6	mA
at V_{DD} = 5 V; f_{XTAL} = 6 MHz	I_{DD}	–	0,5	1	mA
at V_{DD} = 3 V; f_{XTAL} = 3,58 MHz	I_{DD}	–	0,15	0,4	mA
STOP mode (see Fig. 41 and note 1)					
at V_{DD} = 2,5 V; T_{amb} = 25 °C	I_{DD}	–	1,2	2,5	μA
at V_{DD} = 2,5 V; T_{amb} = 85 °C	I_{DD}	–	–	10	μA
Inputs					
Input voltage LOW	V_{IL}	0	–	0,3V	V
Input voltage HIGH	V_{IH}	0,7V_{DD}	–	V_{DD}	V
Input leakage current					
at $V_{SS} < V_I < V_{DD}$	$\pm I_{IL}$	–	–	1	μA
Outputs					
Output voltage LOW					
at V_I = V_{SS} or V_{DD}; $\|I_O\| < 1$ μA	V_{OL}	–	–	0,05	V
Output sink current LOW					
at V_{DD} = 5 V ± 10%; V_O = 0,4 V					
except P23/SDA, SCLK (see Fig. 37) and port 1	I_{OL}	1,6	3	–	mA
P23/SDA, SCLK (see Fig. 38)	I_{OL}	3	–	–	mA
P10-P17 (not PCF84C00) at V_{OL} = 1 V	I_{OL}	10	–	–	mA
Pull-up output source current HIGH (see Fig. 39)					
at V_{DD} = 5 V ± 10%; V_O = 0,7V_{DD}	$-I_{OH}$	40	–	–	μA
at V_{DD} = 5 V ± 10%; V_O = V_{SS}	$-I_{OH}$	–	–	400	μA
Push-pull output source current HIGH					
at V_{DD} = 5 V ± 10%; V_O = V_{DD}−0,4 V	$-I_{OH}$	1,6	3	–	mA

A.C. CHARACTERISTICS

V_{DD} = 2,5 to 5,5 V; V_{SS} = 0 V; T_{amb} = −40 to +85 °C. All voltages with respect to V_{SS} unless otherwise specified.

parameter	symbol	min.	typ.	max.	unit
PCF84C00/non-standard pins (note 12)					
Control pulse width (note 1)	t_{CC}	900	–	–	ns
Address to PSEN set-up (note 2)	t_{AS}	150	–	–	ns
Data to PSEN set-up (note 3)	t_{DS}	260	–	–	ns
Data hold time	t_{DR}	0	–	–	ns
Address to data-in (note 4)	t_{AD}	–	–	790	ns
Rise time all outputs (note 5)	t_R	–	30	–	ns
Fall time all outputs (note 5)	t_F	–	30	–	ns
Data out to DXWR set-up	t_{SDO}	tbf	2	–	ns
Data out to DXWR hold (note 6)	t_{HDO}	100	–	–	ns
Time from DXALE to PSEN (note 7)	t_{SLPH}	150	–	–	ns
Data-in to DXRD LOW to HIGH set-up (note 8)	t_{DS1}	250	–	–	ns
Data hold time after DXRD LOW to HIGH	t_{DR1}	0	–	–	ns
HIGH time of DXALE (note 9)	t_{DXALE}	450	–	–	ns
LOW time of DXRD (note 10)	t_{DXRD}	600	–	–	ns
LOW time of DXWR (note 11)	t_{DXWR}	300	–	–	ns

Notes to characteristics

1 CP = 1 clock pulse = 100 ns at f_{XTAL} = 10 MHz; 1 time slot = 3 CP

1. t_{CC} = 9 CP
2. t_{AS} = 1,5 CP
3. t_{OS} = 2 CP + t_{CPS1}
4. t_{AD} = 8,5 CP − t_{CPS1}
5. CL = 50 pF
6. t_{HDO} = 1 CP
7. t_{SLPH} = 1,5 CP
8. t_{DS1} = 2,5 CP
9. t_{DXALE} = 4,5 CP
10. t_{DXRD} = 6 CP
11. t_{DXWR} = 3 CP
12. All values for the PCF84C00 are preliminary.

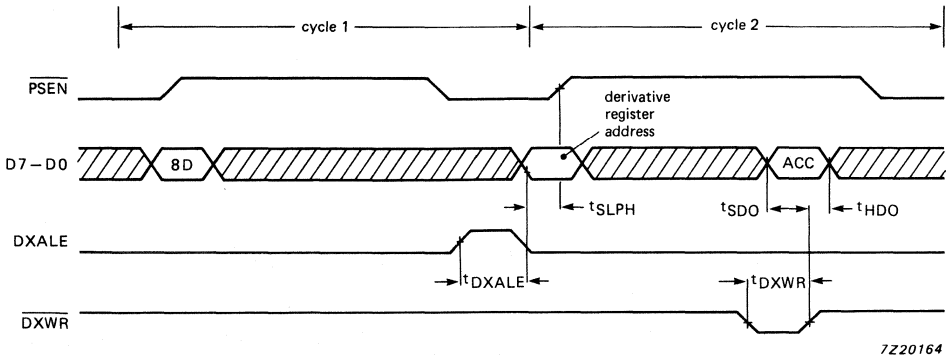


Fig. 29 Instruction MOV Dx,A timing (PCF84C00T only).

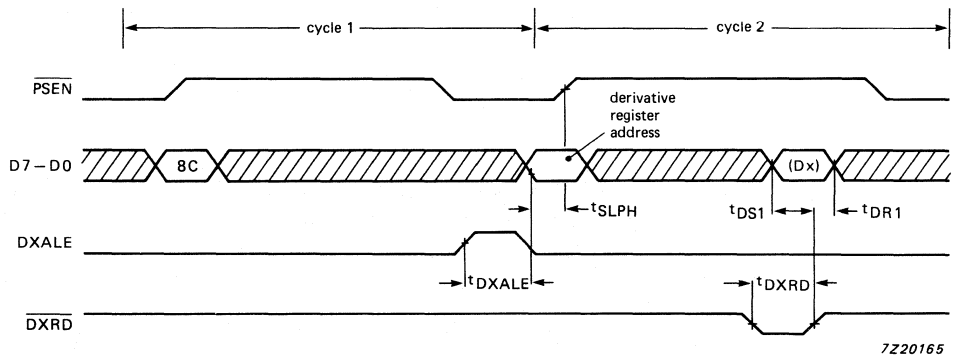


Fig. 30 Instruction MOV A,Dx timing (PCF84C00T only).

A.C. CHARACTERISTICS (continued)

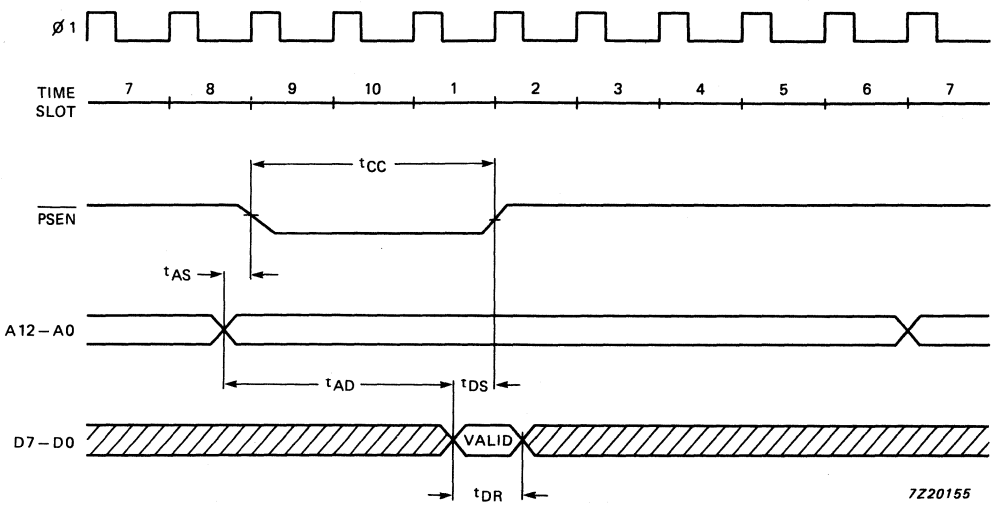


Fig. 31 Timing external memory (PCF84C00T only).

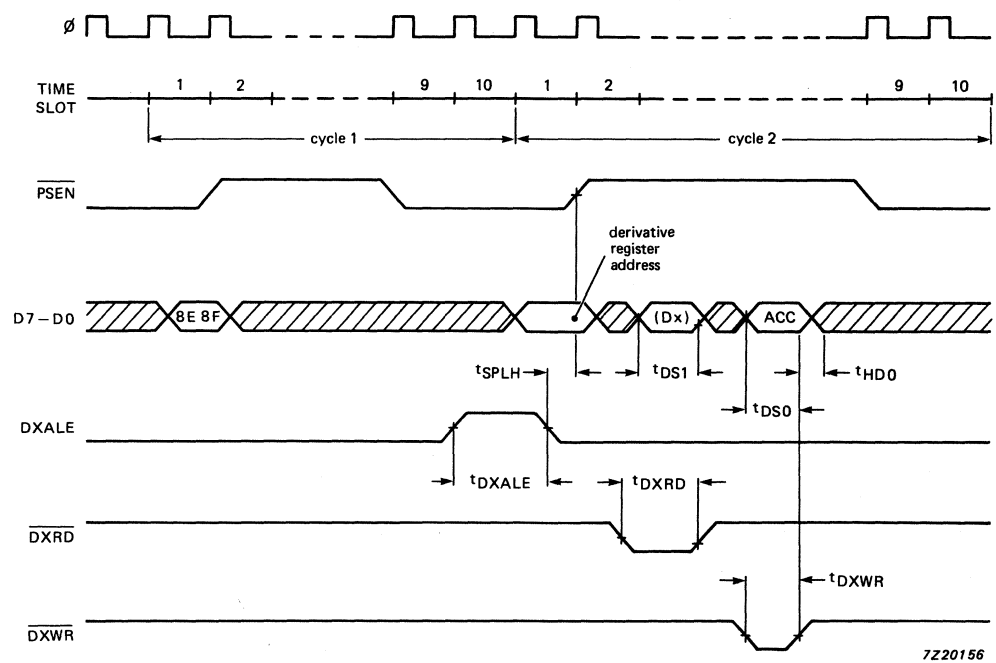
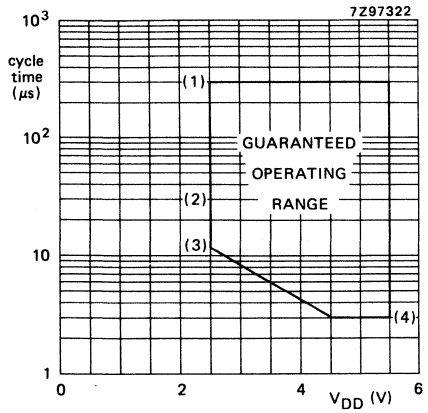
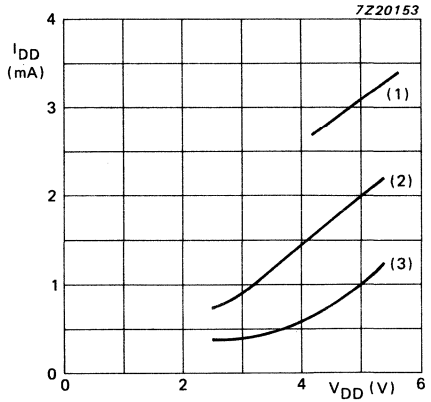


Fig. 32 ANL/ORL derivitive interface timing (PCF84C00T only).



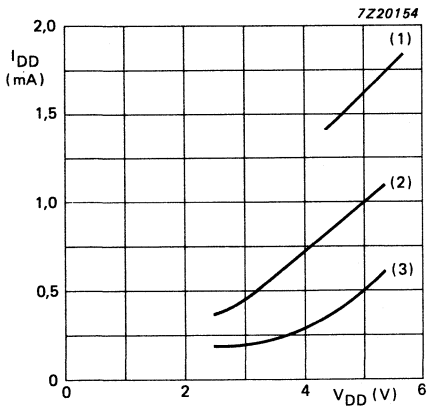
(1) clock frequency = 200 kHz
(2) clock frequency = 1 MHz
(3) clock frequency = 3 MHz
(4) clock frequency = 10 MHz

Fig. 33 Maximum clock frequency (f_{XTAL}) as a function of the supply voltage (V_{DD}).



(1) clock frequency = 10 MHz
(2) clock frequency = 6 MHz
(3) clock frequency = 3,58 MHz

Fig. 34 Maximum supply current (I_{DD}) in operating mode as a function of the supply voltage.



(1) clock frequency = 10 MHz
(2) clock frequency = 6 MHz
(3) clock frequency = 3,58 MHz

Fig. 35 Maximum supply current (I_{DD}) in IDLE mode as a function of the supply voltage (V_{DD}).

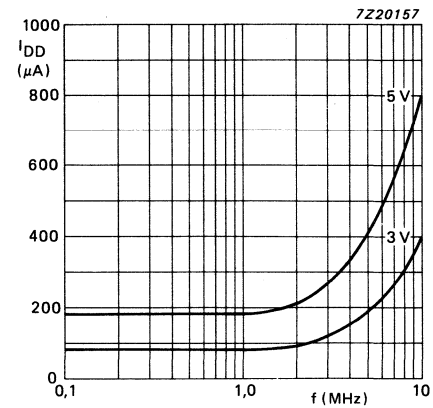


Fig. 36 Typical supply current during IDLE mode as a function of frequency at V_{DD} = 3 V and V_{DD} = 5 V.

DEVELOPMENT DATA

A.C. CHARACTERISTICS (continued)

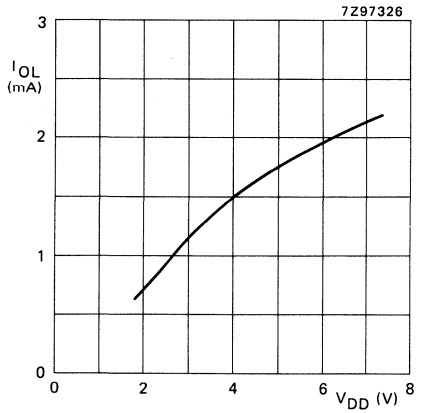


Fig. 37 Output sink current LOW (I_{OL}), except outputs P23/SDA and SCLK, as a function of supply voltage (V_{DD}); V_O = 0,4 V.

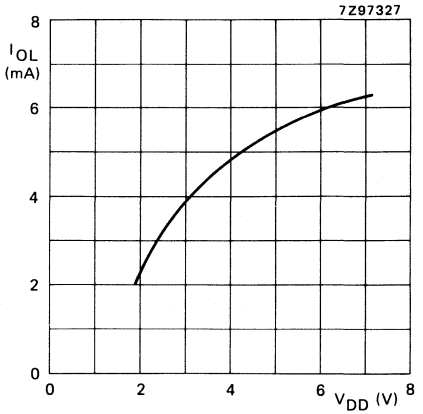
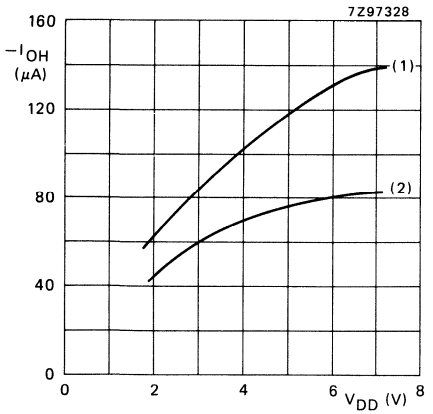


Fig. 38 Output current LOW (I_{OL}), outputs P23/SDA and SCLK, as a function of supply voltage (V_{DD}); V_O = 0,4 V.



(1) $V_O = V_{SS}$
(2) V_O = 0,7 V_{DD}

Fig. 39 Output source current HIGH ($-I_{OH}$) as a function of supply voltage (V_{DD}).

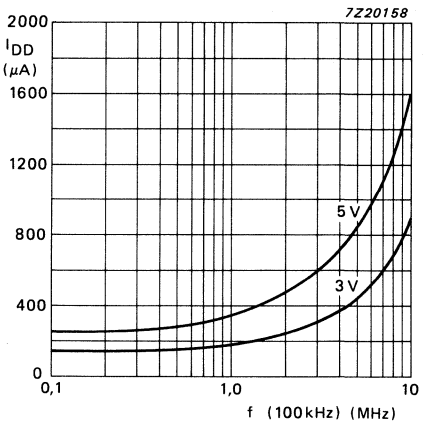
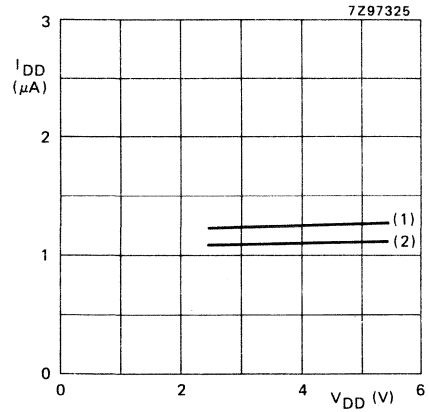


Fig. 40 Typical supply current during operating mode as a function of frequency at V_{DD} = 3 V and V_{DD} = 5 V.



(1) T_{amb} = 85 °C
(2) T_{amb} = 25 °C

Fig. 41 Typical supply current (I_{DD}) in STOP mode as a function of the supply voltage (V_{DD}).

Table 8 Input timing shown in figure 42.

symbol	timing
t_{BUF}	$\geqslant 14t_{XTAL}$
$t_{HD;STA}$	$\geqslant 14t_{XTAL}$
t_{HIGH}	$\geqslant 17t_{XTAL}$
t_{LOW}	$\geqslant 17t_{XTAL}$
$t_{SU;STO}$	$\geqslant 14t_{XTAL}$
$t_{HD;DAT}$	> 0
$t_{SU;DAT}$	$\geqslant 250$ ns
t_{RD}	$\leqslant 1\ \mu s$
t_{RC}	$\leqslant 1\ \mu s$
t_{FD}	$\leqslant 1\ \mu s$
t_{FC}	$\leqslant 0{,}3\ \mu s$

Notes to Table 8

t_{XTAL} = one period of the XTAL input frequency (f_{XTAL})
= 167 ns for f_{XTAL} = 6 MHz.
These figures apply to all modes.

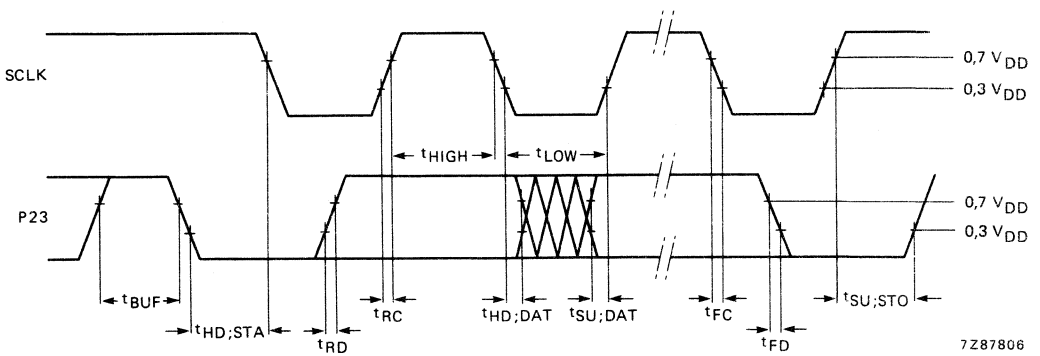


Fig. 42 PCF84CXX timing requirements for the P23 and SCLK *input* signals.

DEVELOPMENT DATA

Purchase of Philips' I²C components conveys a license under the Philips' I²C patent to use the components in the I²C-system provided the system conforms to the I²C specifications defined by Philips.

A.C. CHARACTERISTICS (continued)

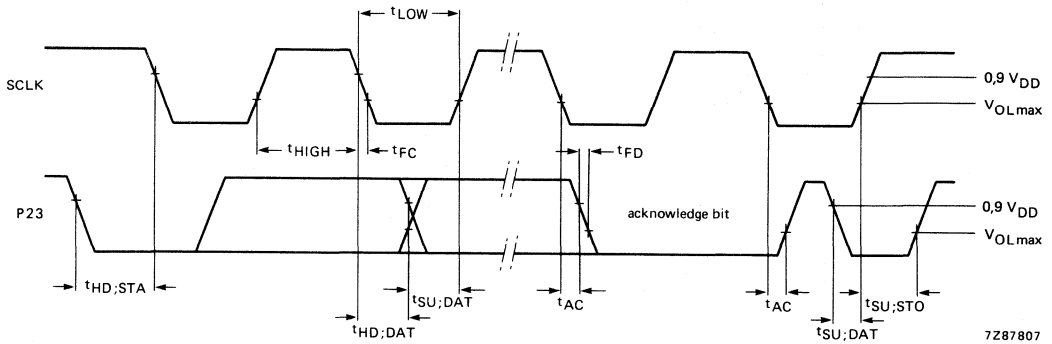


Fig. 43 PCF84CXX timing requirements for the P23 and SCLK *output* signals.

Table 9 Output timing shown in Figure 43

symbol	timing	
	normal mode (ASC in S2 = 0)	low-speed mode (ASC in S2 = 1)
$t_{HD;STA}$	½ (DF + 9) t_{XTAL}	¾ (DF + 9) t_{XTAL}
t_{HIGH}	½ (DF) t_{XTAL}	¾ (DF) t_{XTAL}
t_{LOW}	½ (DF) t_{XTAL}	¼ (DF) t_{XTAL}
$t_{SU;STO}$	½ (DF − 3) t_{XTAL}	¼ (DF − 3) t_{XTAL}
$t_{HD;DAT}$ (slave transmitter)		
any DF	$\geqslant 9t_{XTAL}$ $\leqslant 12t_{XTAL}$	$\geqslant 9t_{XTAL}$ $\leqslant 12t_{XTAL}$
$t_{HD;DAT}$ (master transmitter)		
for DF $\leqslant$ 51	$\geqslant 9t_{XTAL}$ $\leqslant 12t_{XTAL}$	–
for DF $\leqslant$ 99	–	$\geqslant 9t_{XTAL}$ $\leqslant 12t_{XTAL}$
$t_{SU;DAT}$ (master transmitter)		
for DF > 51	$\geqslant 15t_{XTAL}$ $\leqslant 24t_{XTAL}$	–
for DF > 99	–	$\geqslant 15t_{XTAL}$
t_{AC}	$\geqslant 9t_{XTAL}$ $\leqslant 12t_{XTAL}$	$\geqslant 9t_{XTAL}$ $\leqslant 12t_{XTAL}$
t_{FD}, t_{FC}	$\leqslant$ 100 ns at C_b = 400 pF	$\leqslant$ 100 ns at C_b = 400 pF

Notes to Table 9

t_{XTAL} = one period of the XTAL input frequency (f_{XTAL})
= 167 ns for f_{XTAL} = 6 MHz.
DF = divisor (see Table 2 Serial I/O section).
C_b = the maximum bus capacitance for each line.

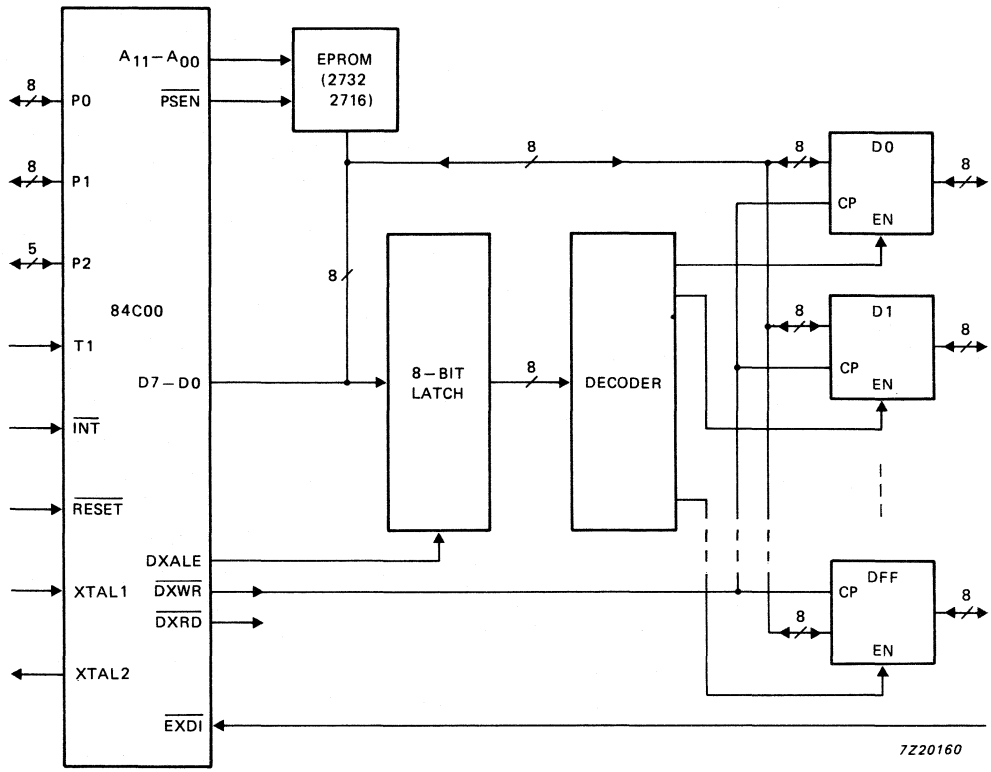


Fig. 44 Block diagram of the external Dx register interface. The Dx interface can only be used with PCF84C00T.

SINGLE-CHIP 8-BIT MICROCONTROLLER

DESCRIPTION

The PCF84C85 microcontroller is manufactured in CMOS, and is designed to be an efficient controller as well as an arithmetic processor. The instruction set is based on that of the MAB8048 and is software compatible with the PCF84CXX family. The PCF84C85 has two additional derivative ports and the microcontroller has bit handling abilities and facilities for both binary and BCD arithmetic.

For detailed information on the PCF84CXX see the "Single-chip 8-bit Microcontrollers" user manual.

Features

- 8-bit CPU, ROM, RAM, I/O in a single 40-lead DIL or mini-pack package
- 8 K ROM
- 256 RAM bytes
- 32 quasi-bidirectional I/O port lines
- Two test inputs: one of which is also the external interrupt input
- Single-level vectored interrupts: external, timer/event counter, serial I/O
- Serial I/O which can be used in single or multi-master systems (serial I/O data via an existing port line and clock via a dedicated line)
- 8-bit programmable timer/event counter
- Clock frequency 100 kHz to 10 MHz
- Over 80 instructions (based on MAB8048) all of 1 or 2 cycles
- Single supply voltage from 2,5 V to 5,5 V
- STOP and IDLE mode
- Power-on-reset circuit
- Operating temperature range: −40 to +85 °C

PACKAGE OUTLINES

PCF84C85P: 40-lead DIL; plastic (SOT-129)
PCF84C85T: 40-lead; mini-pack (SOT-158A)

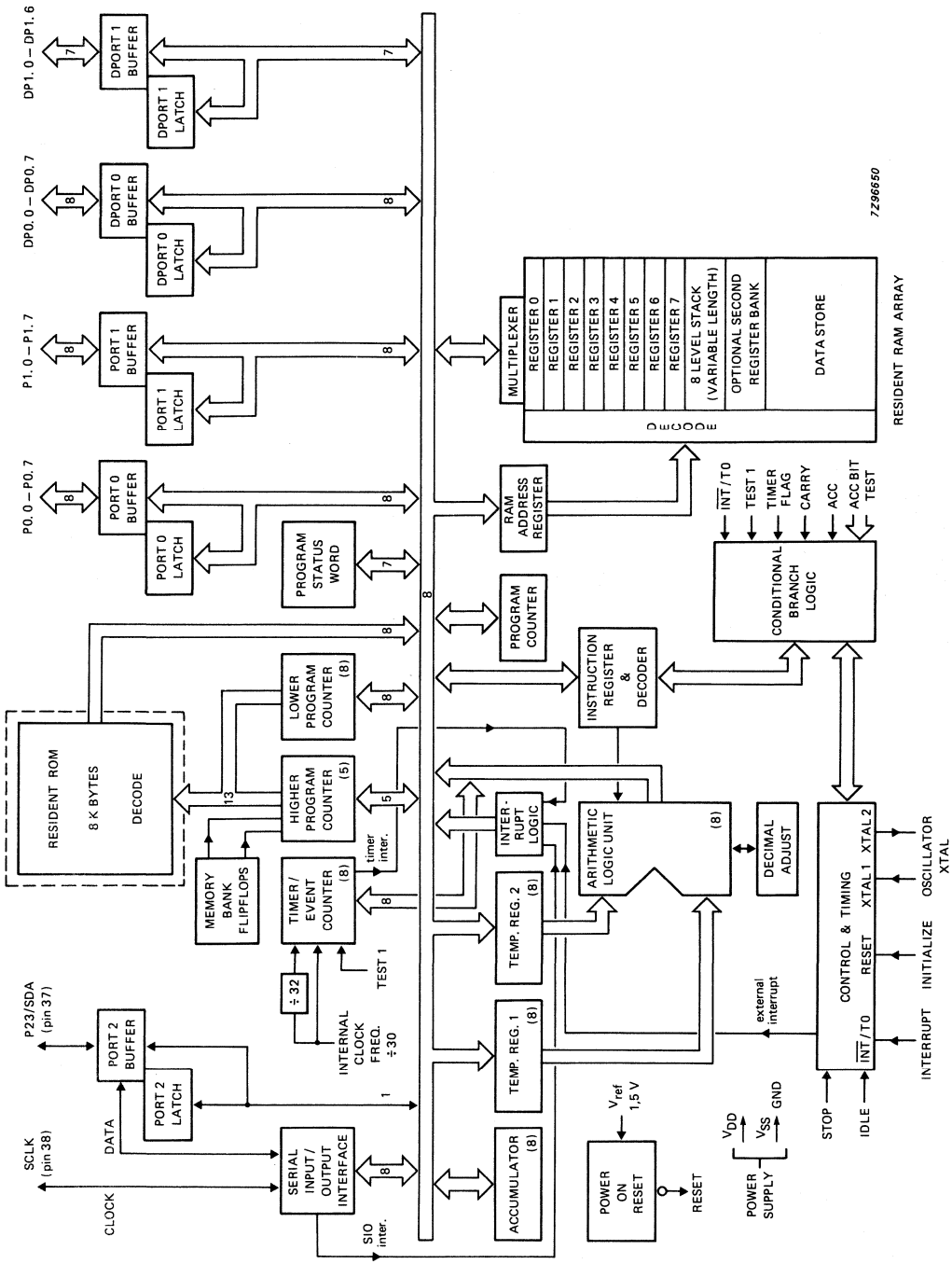


Fig. 1 Block diagram.

DEVELOPMENT DATA

PINNING

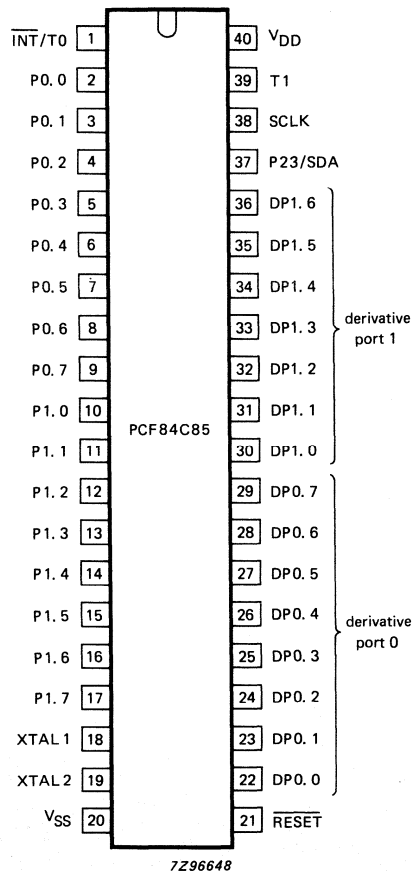


Fig. 2 Pinning diagram.

PIN DESIGNATION

38	SCLK	**Clock:** bidirectional clock for serial I/O.
1	$\overline{\text{INT}}$/T0	**Interrupt/Test 0:** external interrupt input (sensitive to negative-going edge)/test input pin; when used as a test input directly tested by conditional branch instructions JT0 and JNT0.
39	T1	**Test 1:** test input pin, directly tested by conditional branch instructions JT1 and JNT1. T1 also functions as an input to the 8-bit timer/event counter, using the STRT CNT instruction.
18	XTAL 1	**Crystal input:** connection to timing component (crystal) which determines the frequency of the internal oscillator; also the input for an external clock source.
19	XTAL 2	Connection to the other side of the timing component.
21	RESET	**Reset input:** used to initialize the processor (active HIGH), or output of power-on-reset circuit.

PIN DESIGNATION (continued)

2-9	P0.0-P0.7	**Port 0:** 8-bit quasi-bidirectional I/O port.
10-17	P1.0-P1.7	**Port 1:** 8-bit quasi-bidirectional I/O port.
37	P23/SDA	**Port 2:** 1-bit quasi-bidirectional I/O port or serial data input/output in serial I/O mode.
22-29	DP0.0-DP0.7	**Derivitive port 0:** 8-bit quasi-bidirectional I/O port.
30-36	DP1.0-DP1.6	**Derivitive port 1:** 7-bit quasi-bidirectional I/O port.
20	V_{SS}	**Ground:** circuit earth potentional.
40	V_{DD}	**Power supply:** 2,5 V to 5,5 V.

FUNCTIONAL DESCRIPTION

Program memory

The program memory consists of 8 K bytes, in a read-only memory (ROM). Each location is directly addressable by the program counter. The memory is mask-programmed at the factory. Figure 3 shows the program memory map.

Four program memory locations are of special importance:

- Location 0; contains the first instruction to be executed after the processor is initialized (RESET),
- Location 3; contains the first byte of an external interrupt service subroutine;
- Location 5; contains the first byte of a serial I/O interrupt service subroutine.
- Location 7; contains the first byte of a timer/event counter interrupt service subroutine.

Program memory is arranged in banks of 2 K bytes, which are selected by SEL MB instructions. The program memory is further divided into location 'pages', each of 256 bytes. This latter division applies only for conditional branches. Memory bank boundaries can be crossed only by using unconditional branch instructions after the appropriate memory bank has been selected. A CALL instruction can transfer control to a subroutine on any 'page'; RET and RETR instructions can transfer control from a subroutine back to the main program.

Data memory

Data memory consists of 256 bytes, random-access data memory (RAM). All locations are indirectly addressable using RAM pointer registers; up to 16 designated locations are directly addressable. Memory also includes an 8-level program counter stack addressed by a 3-bit stack pointer. Figure 4 shows the data memory map.

Working registers

Locations 0 to 7 are designated as working registers, directly addressable by the direct register instructions. Ease of addressing, and a minimum requirement of instruction bytes to manipulate their contents, makes these locations suitable for storing frequently addressed intermediate results. This bank of registers can be selected by the SEL RB0 instruction.

Executing the select register bank instruction SEL RB1, designates locations 24 to 31 as working registers, instead of locations 0 to 7, and these are then directly addressable. This second bank of working registers may be used as an extension of the first or reserved for use during interrupt service subroutines saving the first bank for use in the main program. If the second bank is not used, locations 24 to 31 may serve as general purpose RAM.

The first locations of each bank contain the RAM pointer registers R0, R1, R0' and R1', which indirectly address all RAM locations.

All RAM locations make efficient program loop counters when used with the decrement register and test instruction DJNZ.

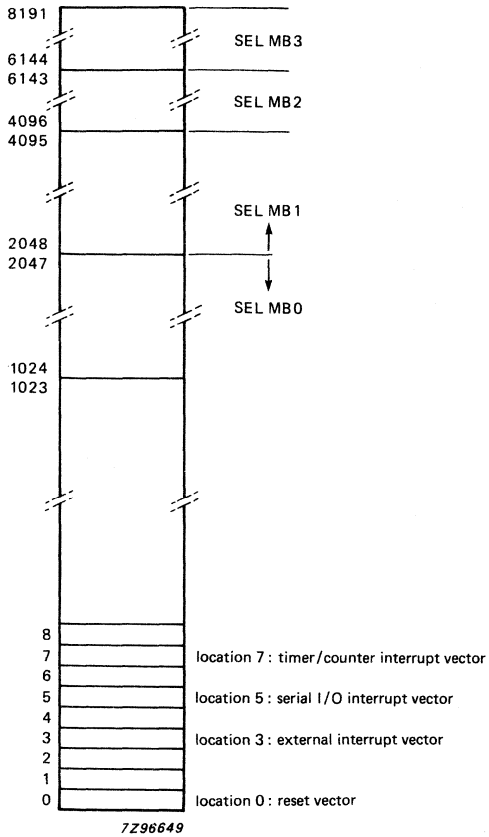


Fig. 3 Program memory map.

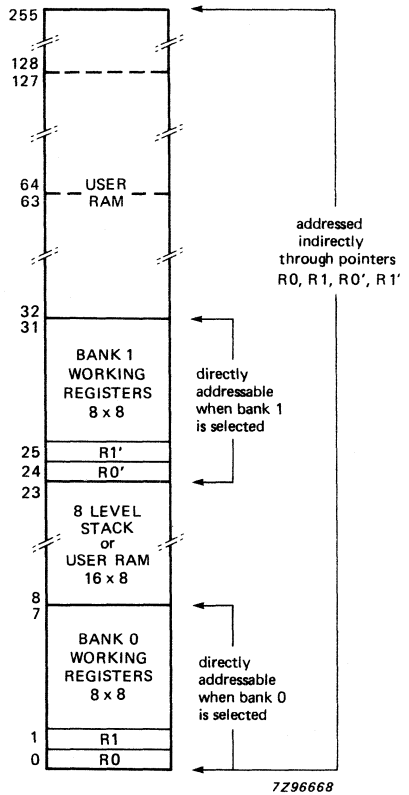


Fig. 4 Data memory map.

Program counter stack

Locations 8 to 23 may be designated as an 8-level program counter stack (2 locations per level), or as general purpose RAM. The program counter stack (Fig.5) enables the processor to keep track of the return addresses and status generated by interrupts or CALL instructions by storing the contents of the program counter prior to servicing the subroutine. A 3-bit stack pointer determines which of the eight register pairs of the program counter stack will be loaded with next generated return address.

The stack pointer, when initialized to 000 by RESET, points to RAM locations 8 and 9. On the first subroutine CALL or interrupt, the contents of the program counter and bits 4, 6 and 7 of the program status word (PSW) are transferred to locations 8 and 9. The stack pointer increments by one and points to locations 10 and 11 ready for another CALL. Because an address may be up to 13 bits long, two bytes must be used to store each address.

At the end of a subroutine, which is signalled by a return instruction (RET or RETR), the stack pointer decrements by one and the contents of the register pair on top of the stack are transferred to the program counter. The saved PSW bits are transferred to the PSW only by the RETR instruction.

If not all 8 levels of subroutine and interrupt nesting are used, the unused portion of the stack may be used as any other indirectly addressable RAM locations.

Nesting of subroutines within subroutines can continue up to 8 times without overflowing the stack. If overflow does occur the deepest address stored (locations 8 and 9) will be overwritten and lost since the stack pointer overflows from 111 to 000. It also underflows from 000 to 111.

FUNCTIONAL DESCRIPTION (continued)

The value of the saved contents of the program counter is different for an interrupt CALL compared to a normal CALL to subroutine. With an interrupt CALL, the program counter return address is saved; with a subroutine CALL, the saved program counter value is one less than the program counter return address.

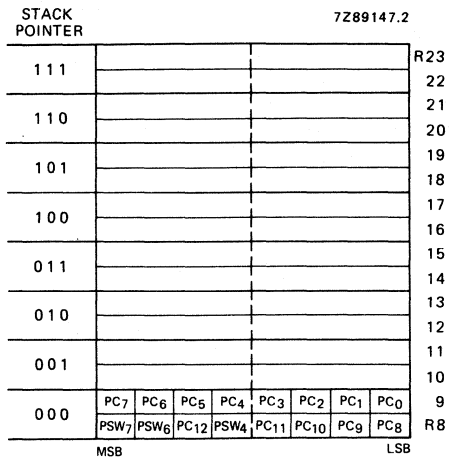


Fig. 5 Program counter stack.

IDLE and STOP modes

IDLE mode

When the microcontroller enters the IDLE mode via the IDLE instruction (01 H) the oscillator, timer/counter and serial I/O are kept running. The microcontroller exits from the IDLE mode by one of three interrupts if they are enabled or by activating a RESET. If the interrupt is not enabled the processor will remain in the IDLE mode. An active signal on the RESET pin restarts the microcontroller and a normal RESET sequence is executed (see Fig. 6).

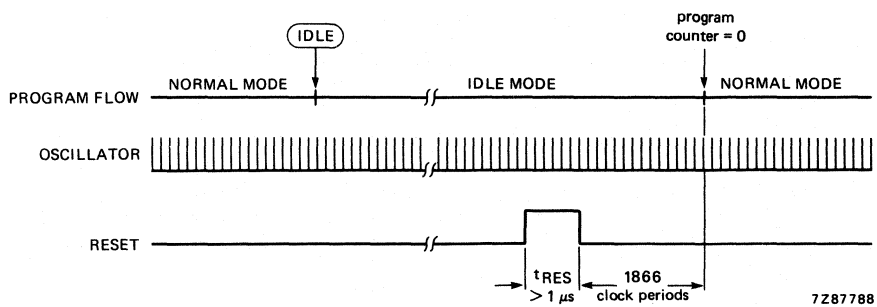


Fig. 6 Exit from IDLE mode via a RESET.

An active signal coming from an enabled interrupt causes the execution of the normal interrupt routine since normal interrupt scanning is still being carried out. A HIGH-to-LOW transition on the external interrupt pin ($\overline{INT}$/T0) reactivates the microcontroller. A LOW level applied to $\overline{INT}$/T0 will reactivate the microcontroller only in the STOP mode. Thus,if $\overline{INT}$/T0 was LOW before the microcontroller entered the IDLE mode, it must go HIGH before the microcontroller can be reactivated (see Fig. 7).

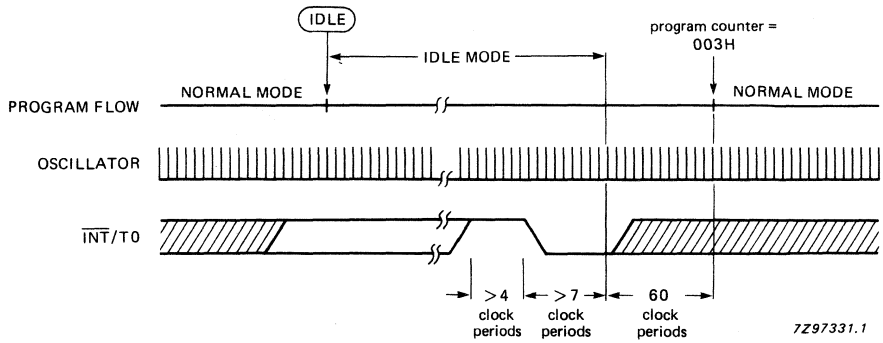


Fig. 7 Exit from IDLE mode via an interrupt.

Wake-up from the IDLE mode is ensured when $\overline{INT}$/T0 is HIGH for at least 4 CP (clock periods) followed by a LOW for 7 CP. After the initial forced CALL(03 H) operation (60 CP) the program continues with the external interrupt service routine.

STOP mode

The microcontroller enters the STOP mode by the STOP instruction (22 H). The oscillator is switched off. The internal status of the CPU, RAM contents and the state of I/O ports are not affected. The microcontroller can be brought-out of the STOP mode by an active signal at the external interrupt input or by an external RESET signal. When one of these two signals is applied an internal delay of 1866 CP is provided to ensure that all internal clocks are operating correctly before restart (see Fig. 8). Note; the start-up time of a crystal oscillator is measured in milliseconds, and the 1866 CP count begins after this start-up time.

If the microcontroller exits from the STOP mode by activating RESET, a normal RESET sequence is executed.

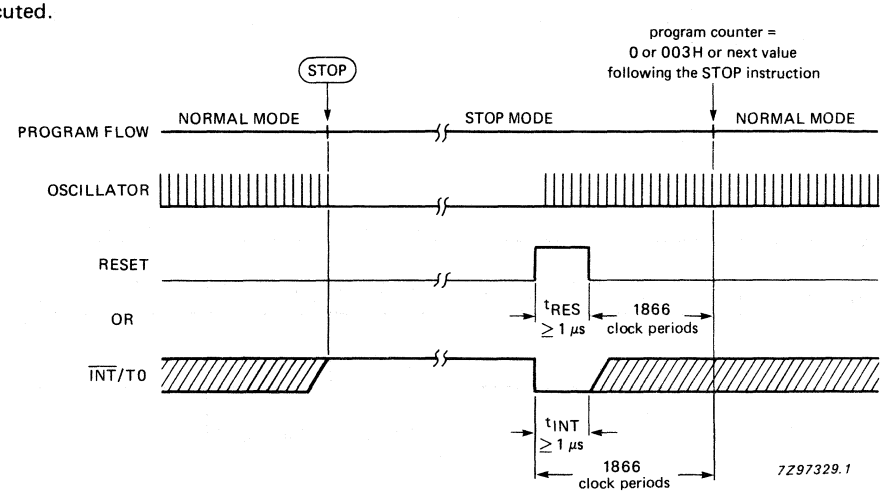


Fig. 8 Entering and exiting the STOP mode.

FUNCTIONAL DESCRIPTION (continued)

If the microcontroller exits the STOP mode by pulling the external interrupt input pin LOW, an interrupt sequence is executed only if the external interrupt is enabled. In this event the microcontroller resumes the normal program sequence after returning from the interrupt routine, as in the normal mode. If the interrupt is not enabled, it continues the normal program sequence, executing the instruction following the STOP instruction.

The microcontroller is restarted by a LOW level applied at the $\overline{INT}$/T0 pin, and not by a HIGH-to-LOW transition as in a normal interrupt mechanism.
Note: when leaving the STOP mode with an interrupt, a further instruction in the main program series is executed prior to entering the interrupt routine.

When the $\overline{INT}$/T0 level is active during the STOP instruction then no STOP is executed.

A LOW level on the external interrupt input of at least 1 μs will cause the microcontroller to exit the STOP mode.

I/O facilities

The PCF84C85 family has 32 I/O lines arranged as:

- Port 0 parallel port of 8 lines (P0.0 to P0.7)
- Port 1 parallel port of 8 lines (P1.0 to P1.7)
- Port 2 parallel port of 1 line (P2.0)
- D port 0 parallel port of 8 lines (DP0.0-DP0.7)
- D port 1 parallel port of 7 lines (DP1.0-DP1.6)
- SCLK I^2C-bus serial clock line shared with parallel port line P2.3
- $\overline{INT}$/T0 external interrupt and test input. When used as a test input can be directly tested by conditional branch instructions JT0 and JNT0
- T1 test input which can alter program sequences when tested by conditional jump instructions JT1 and JTN1. T1 also functions as an input to the 8-bit timer/event counter.

Parallel ports

All parallel ports can be used as outputs or inputs, their structure is quasi-bidirectional.
Output data written to a port is latched and remains unchanged until rewritten.
Input data is not latched and so must be present until read by an input instruction.

Input lines are fully CMOS compatible, output lines can drive one TTL or CMOS load.

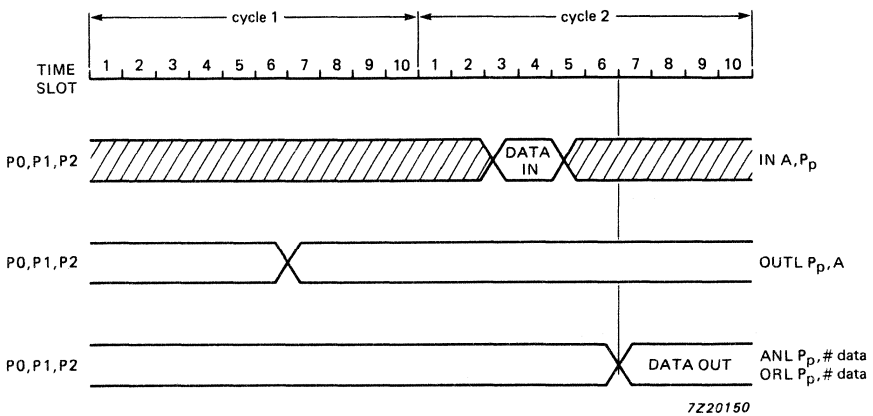


Fig. 9 Shows the timing diagram for all ports using IN, OUTL, ANL and ORL instructions. For the OUTL instruction data changes on time slot 7 of cycle 1. For the MOV, ANL and ORL instructions, the ports change on time slot 7 of cycle 2.

Fig. 10 shows the quasi-bidirectional I/O interface with push-pull output and switched pull-up current source. Each line is pulled up to V_{DD} via a constant current source (TR4), which is enabled via TR3 whenever one of the two output latches contains a logic 1. This current source is sufficient for a TTL HIGH level, yet can be pulled LOW by an external CMOS device, thus allowing the same pin to be used for both input and output.

When a logic 1 is written to the line for the first time (MQ = 1, SQ = 0), TR2 is switched on for the duration of the internal write pulse (one oscillator period), to provide a fast transition from logic 0 to logic 1. Subsequent writing of a logic 1 to the port lines will not switch TR2 on. This prevents unnecessary current through external components connected to the port lines of the same port which might be in the input mode and also connected to ground.

When a logic 0 is written to the line, TR3 switches off the current source. Current sinking capability is provided by TR1, which is now switched on. When used as an input, a logic 1 must first be written to the line, otherwise TR1 will remain low impedance.

The PCF84C85 family offers the possibility to select individually 31 of the 32 parallel port pins (not P23), by the following mask options:

Option 1 – STANDARD PORT; quasi-bidirectional I/O with switched pull-up current source of 100 μA (typ.) and P-channel booster transistor TR2.
TR2 is only active during 1 clock cycle (Fig. 10).

Option 2 – OPEN DRAIN; quasi-bidirectional I/O with only an N-channel open drain output. Application as an output requires connection of an external pull-up resistor (Fig. 11).

Option 3 – PUSH-PULL OUTPUT; drive capability of the output will be 1,6 mA (min.) at V_{DD} = 5 V in both polarities. To avoid a large current flowing through the output transistors during the input mode, these push-pull pins must only be used as outputs (Fig. 12).

DEVELOPMENT DATA

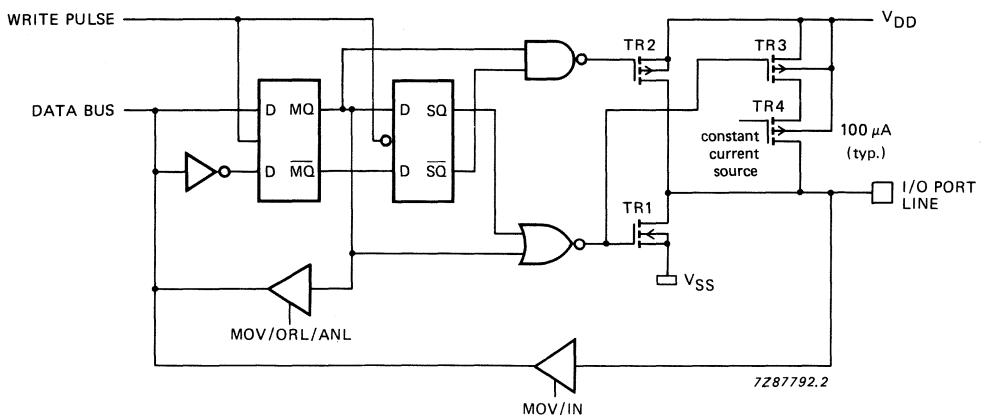


Fig. 10 Standard output with switched pull-up current source.

FUNCTIONAL DESCRIPTION (continued)

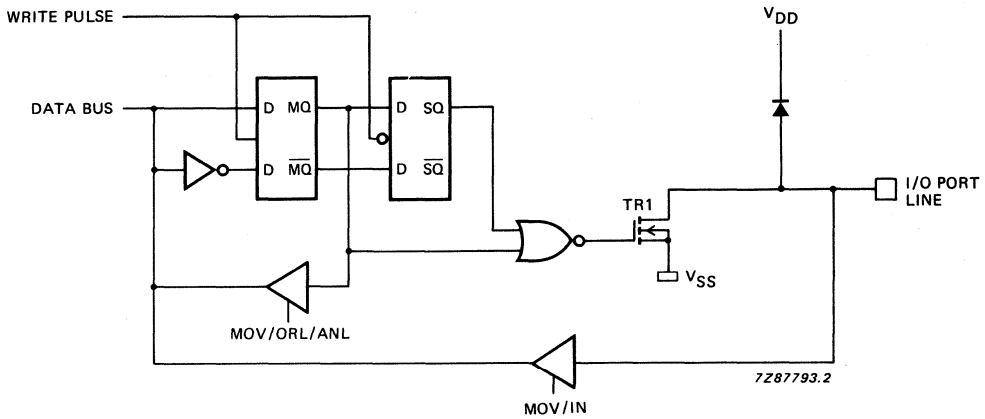


Fig. 11 Open drain output.

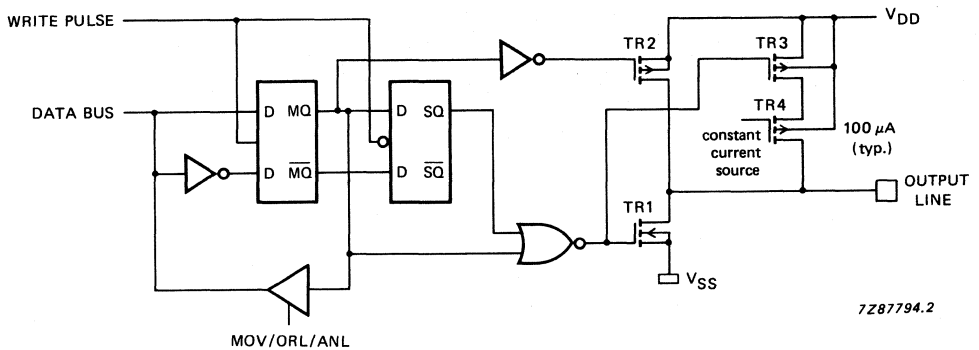


Fig. 12 Push-pull output.

Serial I/O (SIO)

The PCF84C85 has an on-chip serial I/O interface.

Whereas a normal microcontroller must regularly monitor the serial data bus for the presence of data, the serial I/O interface detects, receives and converts the serial data stream into parallel format without interrupting the execution of the current program. An interrupt is sent to the PCF84C85 only when a complete byte is received. It then reads the data byte in one instruction. Likewise during transmission the serial I/O interface performs parallel to serial conversion and subsequent serial output of the data. The microcontroller is only interrupted in the execution of its programmed tasks when a complete byte has been transmitted.

The design of the PCF84C85 serial I/O system allows any number of devices from PCF8500 family (clips) to be connected via the two-line serial bus. The ability of any devices to communicate, without interrupting the operation of any other devices on the bus, is an outstanding attribute of the system. This is achieved by allocating a specific 7-bit address to each device and providing a system whereby a device reacts only to a message prefixed with its own address or the 'general CALL' address. Address recognition is performed by the interface hardware so that operation of the microcontroller need only be interrupted when a valid address has been received. This saves significant processing time and memory space compared with a conventional microcontroller employing a software serial interface. When the addressing facility is not required, for instance in a system with only two microcontrollers, direct data transfer without addressing can be performed. In multi-master systems, an automatically invoked arbitration procedure prevents two or more devices from continuing simultaneous transmission.

In NORMAL (running) and IDLE mode, the serial I/O logic remains active; its internal system clock will be switched off when there is no activity on the serial bus.

After execution of the STOP instruction, the oscillator of the PCF84C85 is switched off. This means that the serial I/O logic will remain in the state it was at the occurance of the STOP instruction. To avoid "bus block" problems and to assure correct start-up of the bus after exit from the STOP mode, the user should disable the serial logic (ESO = 0) prior to the execution of the STOP instruction. This must be carried out only when the PCF84C85 has finished a serial data transfer.

Serial I/O interface

Figure 13 shows the serial I/O interface. The clock line of the serial bus has exclusive use of pin 38 (SCLK) while the data line shares pin 37 (serial data) with the I/O line P23 of port 2. When the serial I/O is enabled, P23 is disabled as a parallel port line; (P23 and SCLK only open drain).

The microcontroller and interface communicate via the internal microcontroller bus and the Serial Interrupt Request line. Data and information controlling the operation of the interface are stored in four registers:

- Data shift register (S0)
- Serial I/O interface status word (S1)
- Serial clock control word (S2)
- Address register

FUNCTIONAL DESCRIPTION (continued)

Data shift register (S0)

Register S0 converts serial data to parallel format and vice versa. A pending interrupt is generated only after a complete byte has been transmitted, or after a complete data byte, specific address or 'general CALL' address has been received. The most significant bit is transmitted first.

Serial I/O interface status word (S1)

Register S1 provides information concerning the state of the interface and stores information from the microcontroller. Bits 0 to 3 are duplicated: control bits in these positions can only be written by the microcontroller, while interface bits can only be read.

MST and TRX (see Table 1)

These bits determine the operating mode of the serial I/O interface.

Table 1 Operating modes of the serial I/O interface

MST	TRX	operating mode
0	0	slave receiver
1	0	master receiver
0	1	slave transmitter
1	1	master transmitter

BB: Bus Busy.

This is the flag which indicates the status of the bus.

PIN: Pending Interrupt Not

PIN = '0' indicates the presence of a pending interrupt, which will cause a Serial Interrupt Request when the serial interrupt mechanism is enabled.

ESO: Enable Serial output

The ESO flag enables/disables the serial I/O interface: ESO = '1' enables, ESO = '0' disables. ESO can only be written by software.

BC0, BC1 and BC2

Bits BC0, BC1 and BC2 indicate the number of bits received or transmitted in a data stream. These bits can only be written by software.

AL: Arbitration Lost

The arbitration lost flag is set by hardware when the serial I/O interface, as master transmitter, loses a bus arbitration procedure.

AAS: Addressed As Slave

This flag is set by hardware when the interface detects either its own specific address or the 'general CALL' address as the first byte of a transfer and the interface has been programmed to operate in the address recognition mode.

AD0: Address Zero

This flag is set by hardware after detection of the 'general CALL' address when the interface is operating in the address recognition mode.

LRB: Last Received Bit

This contains either the last data bit received or, for a transmitting device in the acknowledgement mode, the acknowledgement signal from the receiving device.

Bits AL, AAS, AD0 and LRB can only be read by software.

Serial clock control word (S2)

Bits 0 to 4 of the clock control register S2 are used to set the frequency of the serial clock signal. When a 6 MHz crystal is used, the frequency of the serial clock can be varied between 154 kHz and 1 kHz (see Table 2). An asymmetrical clock with a HIGH-to-LOW ratio of 3 : 1 can be generated using bit 5. The asymmetrical clock allows a microcontroller more time per clock period for sampling the data line, making the timing of this action less critical. Bit 6 can be used to activate the acknowledge mode of the serial I/O. S2 is a write only register.

Address register

The address register contains the 7-bit address back-up latches and the bit (ALS) used to enable/disable the address recognition mode. The address register can be written using the MOV S0, A and MOV S0, # data instructions, but only when ES0 = '0'.

Serial I/O interrupt logic

An EN SI instruction enables and a DIS SI instruction disables the interrupt logic. When the logic is enabled, a pending interrupt results in a serial I/O interrupt to the processor, causing a CALL to location 5 in the ROM. When disabled, the presence of an interrupt is still indicated by PIN in S1, allowing the interrupt to be serviced. However, vectored interrupt will not occur.

FUNCTIONAL DESCRIPTION (continued)

Table 2 SIO clock pulse frequency control when using 6 MHz crystal

hexadecimal S20-S24 code	divisor	f_{XTAL}(6 MHz) f_{SCLK} (kHz)	f_{XTAL} (10 MHz) f_{SCLK} (kHz)
0	not allowed		
1	39	*154	*256
2	45	*133	*222
3	51	*118	*196
4	63	95	*159
5	75	80	*133
6	87	69	*115
7	99	61	*101
8	123	49	81
9	147	41	68
A	171	35	58
B	195	31	51
C	243	25	41
D	291	21	34
E	339	18	29
F	387	16	26
10	483	12	21
11	579	10	17
12	675	8,9	15
13	771	7,8	13,4
14	963	6,2	10,4
15	1155	5,2	8,7
16	1347	4,5	7,4
17	1539	3,9	6,5
18	1923	3,1	5,2
19	2307	2,6	4,3
1A	2691	2,2	3,7
1B	3075	2,0	3,3
1C	3843	1,6	2,6
1D	4611	1,3	2,2
1E	5379	1,1	1,9
1F	6147	1,0	1,6

* Not permitted for I^2C operation; the maximum clock frequency in the I^2C systems is 100 kHz.

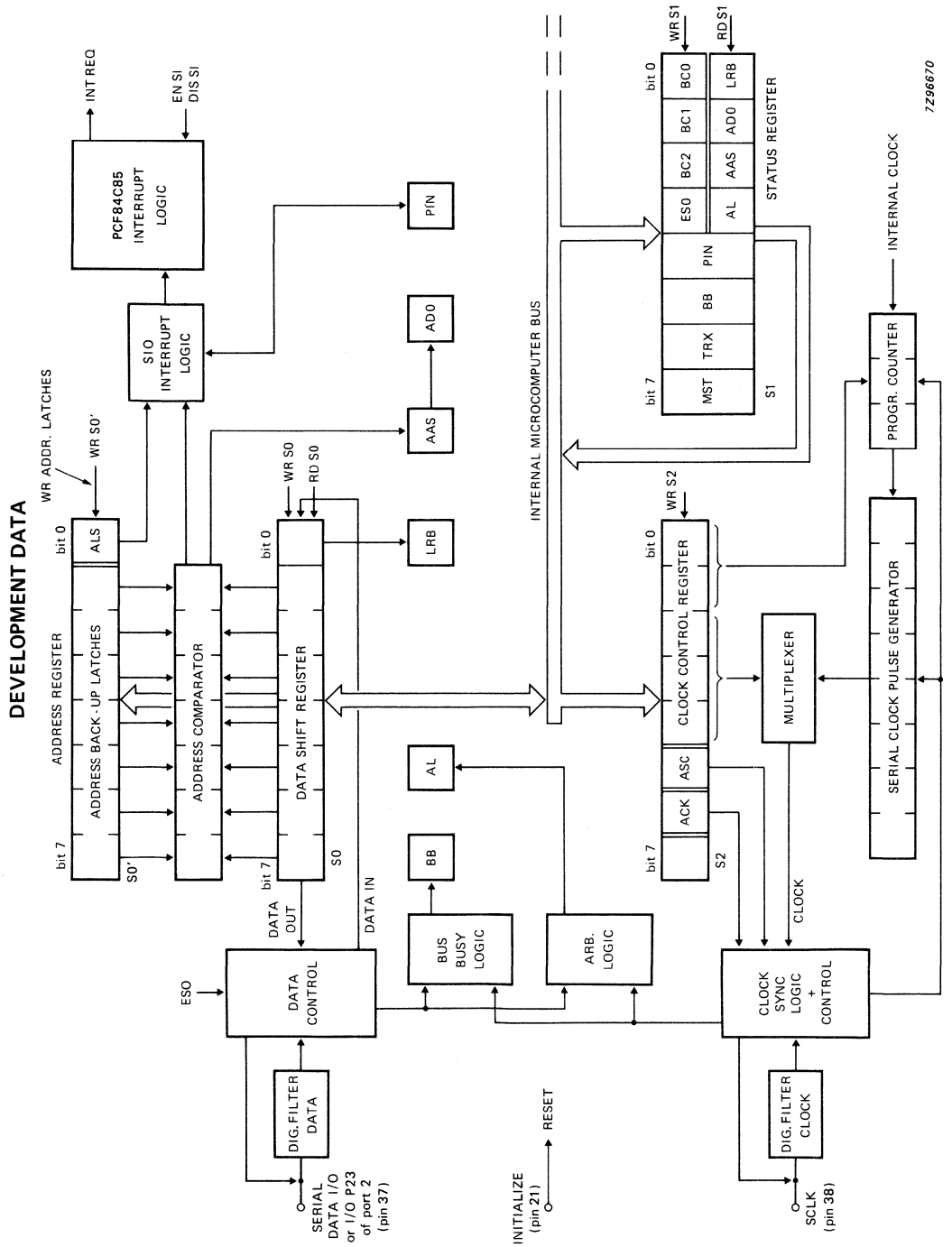


Fig. 13 Serial I/O interface.

FUNCTIONAL DESCRIPTION (continued)

Interrupts (see Fig. 14)

When the external interrupt is enabled, a HIGH-to-LOW transition on the $\overline{INT}$/T0 input initiates an external interrupt subroutine which causes a CALL to program memory location 3 following completion of the current instruction.

Serial I/O interrupt, when enabled, causes a CALL to location 5, and a timer/event counter overflow forces a CALL to location 7 when the timer interrupt is enabled.

When an interrupt subroutine starts, the contents of the program counter and bits 4, 6 and 7 of the PSW have been saved in the program counter stack. Accumulator contents have to be saved by software. Interrupt acknowledgement can be carried out by software via port pins. All interrupt subroutines must reside in memory bank 0.

Since the interrupt system is single level, once an interrupt is detected, all further interrupt requests are latched, but ignored, pending a RETR instruction to re-enable the interrupt input logic. After executing RETR, the program continues in the main part; this is independent of the occurance of a second interrupt during the running of the first routine. If 2 or 3 interrupts occur simultaneously, their priority is:

- (1) external
- (2) serial I/O
- (3) timer/event counter

An external interrupt can be generated by using the timer/counter in the event counter mode. The counter is first preset to (FFH), then EN TCNTI instruction is executed. A LOW-to-HIGH transition of the T1 input will then initiate an interrupt subroutine and cause a CALL to location 7.

On execution of a DIS I instruction, the PCF84C85 always clears the digital filter/latch and the External Interrupt Flag.

The Timer Flag (TF) is reset only when the JTF or JNTF instruction is executed or after RESET.

The Timer Interrupt Flag is set when timer overflow occurs, only if the timer interrupt is enabled.

The microcontroller will exit the IDLE mode when any one of the following three interrupts is enabled:

- External
- Serial I/O
- Timer/event counter

There is no internal pull-up or pull-down device connected to the external interrupt input (pin 1). If required pin 1 must be externally connected to a resistor ($R \leqslant 100\ k\Omega$). When the external interrupt is not used pin 1 must be connected to V_{DD}.

Improvements to interrupt and timer logic with respect to the MAB8400 family

For detailed information see the user manual "Single-chip 8-bit Microcontrollers".

DEVELOPMENT DATA

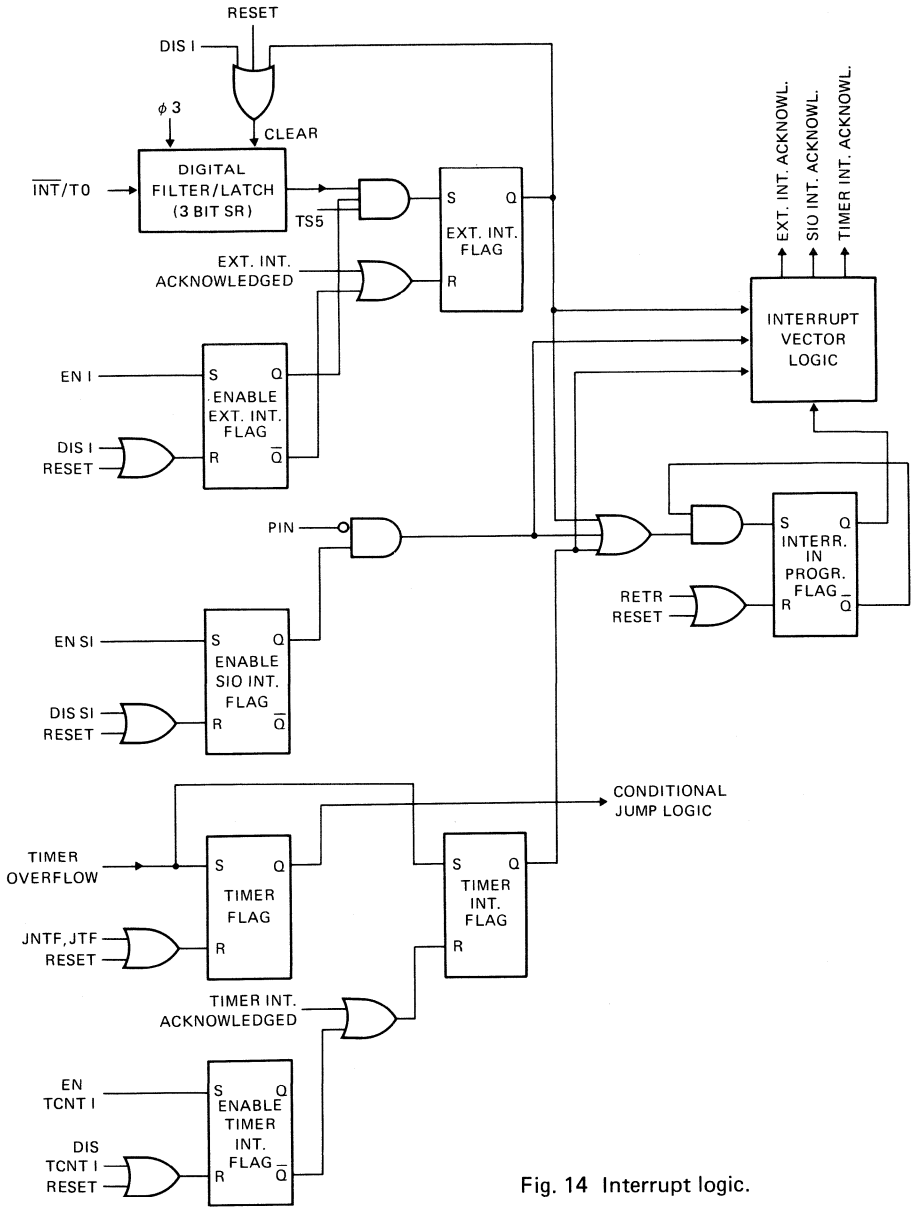


Fig. 14 Interrupt logic.

Notes to figure 14

1. $\overline{\text{INT}}$/T0 negative edge is always latched in the digital filter/latch.
2. Correct interrupt timing is ensured when INT/T0 is HIGH for $>$ 4 CP followed by a LOW for $>$ 7 CP.
3. When the interrupt in progress flag is set, further interrupts are latched but ignored, until RETR is executed.
4. A DIS I instruction always clears a pending external interrupt.
5. For all flip-flops, RESET overrules SET.

FUNCTIONAL DESCRIPTION (continued)

Oscillator (see Fig. 15)

The oscillator can be inhibited by the STOP instruction under software control. It is also inhibited when a low-supply voltage condition is present to prevent discharge of a weak back-up battery. Provided the supply voltage is within the operating range the oscillator will be restarted after a STOP instruction by a LOW level at the $\overline{INT}$/T0 pin or a HIGH level at the RESET pin.

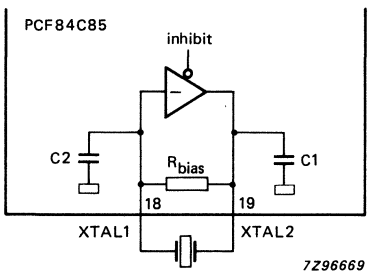


Fig. 15 Oscillator with integrated elements.

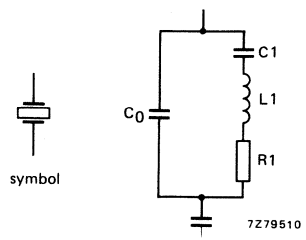


Fig. 16 Crystal unit equivalent circuit.

The values of crystal series resistance R1 and the crystal's total load capacitance C_L (C_0 + wiring + external capacitors) must not be above the curve (Fig. 17) for the corresponding frequency.
Note; if external capacitors are connected to XTAL 1 and XTAL they must be of equal value.

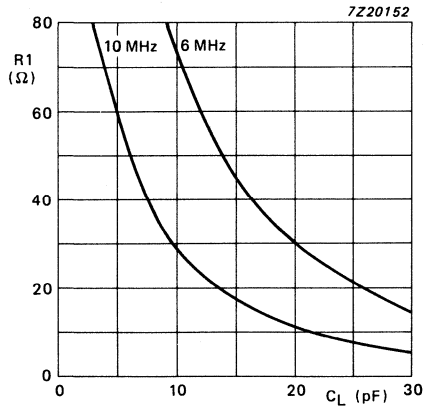


Fig. 17 Crystal circuit criteria.

The oscillator has the output drive capability via pin 19 (XTAL2). An external clock can be applied to pin 18 (XTAL1). A machine cycle consists of 10 time slots, each time slot being 3 oscillator periods.

Timer/event counter (see Fig. 18)

An internal 8-bit up-counter is provided. This can count external events, modulo-32 machine cycles, or machine cycles directly. Table 3 gives the instructions that control the counter and the prescaler, and the functions performed.

When used as a timer, the input to the counter is either the overflow or input of a 5-bit prescaler. When used as an event counter, LOW-to-HIGH transitions on pin 39 (T1) are counted. The maximum rate at which the counter may be incremented is once every machine cycle. When the counter overflows, the timer flag is set. The flag can be tested and reset using the JTF (jump is timer flag = 1) or JNTF instruction. Overflow also generates an interrupt to the processor via setting of the Timer Interrupt Flag when the timer/event counter interrupt is enabled.

Table 3 Timer/event counter control

function	timer mode modulo-1, modulo-32*	counter mode
CLEAR	MOV T,A (A) = 0 or RESET	MOV T,A (A) = 0 or RESET
PRESET	MOV T,A	MOV T,A
START	STRT T	STRT CNT
STOP	STOP TCNT or RESET	STOP TCNT or RESET
TEST	JTF/JNTF	JTF/JNTF
READ**	MOV A,T	MOV A,T

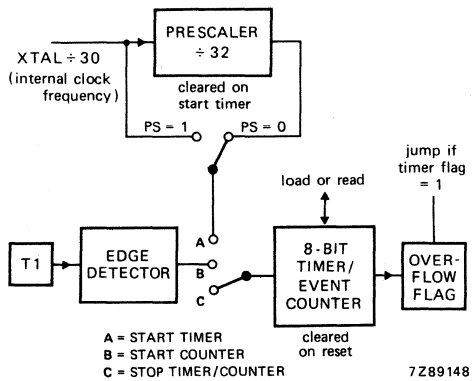


Fig. 18 Timer/event counter.

DEVELOPMENT DATA

Program status word (see Fig. 19)

The program status word (PSW) is an 8-bit word (1 byte) in the CPU which stores information about the current status of the microcontroller.
The PSW bits are:

- Bits 0 to 2 — stack pointer bits (SP_0, SP_1, SP_2)
- Bit 3 — prescaler select (PS);
  0 = modulo-32; 1 = modulo-1 (no prescaling)
- Bit 4 — working register bank select (RBS);
  0 = register bank 0; 1 = register bank 1
- Bit 5 — not used (1)
- Bit 6 — auxiliary carry (AC); half-carry bit generated by an ADD instruction and used by the decimal adjust instruction DA A
- Bit 7 — carry (CY); the carry flag indicates that previous operation has resulted in an overflow of the accumulator.

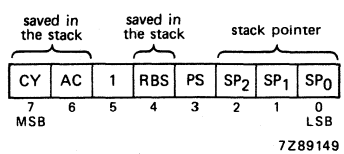


Fig. 19 Program status word.

* With prescaler select, PS = 0, the timer counts modulo-32 machine cycles, with PS = 1 it counts modulo-1 cycles (prescaler not used); prescaler cleared with STRT T, prescaler not readable.

** READ does not disturb the counting process.

FUNCTIONAL DESCRIPTION (continued)

Program status word (continued)

All bits can be read using the MOV A, PSW instruction. Bits 7 and 6 are set and cleared by CPU operation. Bit 4 can be changed by a SEL RB instruction, bit 3 by the MOV PSW, A instruction, and bits 0, 1 and 2 by the CALL, RET or RETR instructions and in the event of an interrupt. Bits 7, 6 and 4 are stored in the program counter stack during subroutine and interrupt calls. These bits are restored in the PSW with a RETR (return and restore) instruction which must be used at the end of an interrupt and can be used at the end of a normal subroutine. The RET instruction has no restore feature and cannot be used at the end of an interrupt.

Program counter (see Fig. 20)

The 13-bit program counter is able to address 8 K bytes of ROM. The arrangement of the bits is shown in figure 20. During an interrupt subroutine PC_{11} and PC_{12} are forced to logic 0. All 13 bits are saved in the stack during CALL and interrupt routines.

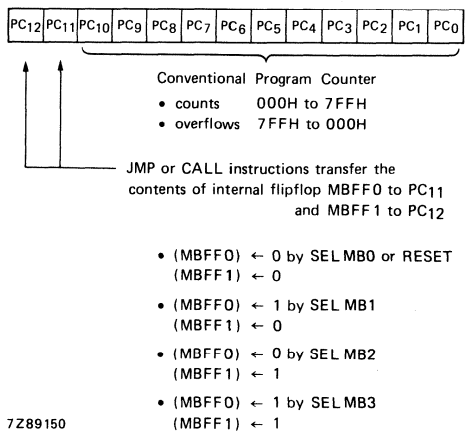


Fig. 20 Program counter.

Central processing unit

The PCF84C85 has arithmetic, logical and branching capabilities. The DA A, SWAP A and XCHD instructions simplify BCD arithmetic and the handling of nibbles. The MOVP A,@A instruction permits efficient table look-up from the current ROM page.

Conditional branch logic

The conditional branch logic within the processor enables several conditions, internal and external to the processor, to be tested by the user's program. Table 4 lists the conditional jump instructions used to change the program sequence. The DJNZ instruction decrements a designated register or data memory location and branches if the contents are not zero. This instruction is useful for looping control. The JMPP@A instruction allows multiway branches to destinations determined by the contents of the accumulator.

Table 4 Conditional branches

test	jump condition	jump instruction
accumulator	all bits zero	JZ
	any bit non-zero	JNZ
accumulator bit test	1	JB0 to JB7
carry flag	1	JC
	0	JNC
timer overflow flag	1	JTF
	0	JNTF
test input T0	1	JT0
	0	JNT0
test input T1	1	JT1
	0	JNT1
register	non-zero	DJNZ

Test input T1 (pin 39)

The T1 input line can be used as:

- A test input for branch instructions JT1 and JNT1
- An external input to the event counter

When used as a test input:

- JT1 instruction tests for logic 1 level
- JNT1 instruction tests for logic 0 level

When used as an input to the event counter, T1 must be LOW for > 4 CP, followed by a HIGH for > 4 CP. The transition can be recognized with a repetition rate of once per 30 oscillator clock periods (1 machine cycle).

There is no internal pull-up or pull-down resistor connected to the T1 input. If required it must be externally connected to a resistor ($R = \leqslant 100\ k\Omega$). When T1 is not used pin 39 must be connected to V_{DD} or V_{SS}.

Reset (pin 21)

A positive-going signal on the RESET input

- Sets the program counter to zero
- Selects location 0 of memory bank 0 and register bank 0
- Sets the stack pointer to zero (000); pointing to RAM address 8
- Disables the interrupts (external, timer and serial I/O)
- Stops the timer/event counter, then sets it to zero
- Sets the timer prescaler to modulo-32
- Resets the timer flag
- Sets all ports to input mode
- Sets the serial I/O to slave receiver mode and disables the serial I/O
- Cancels IDLE and STOP mode

DEVELOPMENT DATA

FUNCTIONAL DESCRIPTION (continued)

Power-on-reset

The internal power-on reset circuit monitors the PCF84C85 supply voltage V_{DD}. For as long as the supply voltage remains below the internal reference level V_{ref} (typically 1,5 V) the oscillator is inhibited and RESET (pin 17) has an undefined level. When V_{DD} rises above the internal reference level, the oscillator is released and RESET is pulled high to V_{DD} by TR1 for a period t_D (typically 50 μs).

N.B. Because of the narrow bandwidth of the crystal, the start-up time of the oscillator is typically 10 ms.

Three modes of power-on reset are possible:

1. If V_{DD} can be switched on with fast rise time i.e. V_{DD} reaches its minimum operating value (corresponding to the selected oscillator frequency) before the RESET signal (t_D) has finished, then no extra components are required (see Fig. 21 and 22). Note that the first instruction is executed after the oscillator start-up time plus 1866 clock periods have elapsed.

2. If V_{DD} has a slow rise time then the RESET signal should be stretched by an external RC circuit (see Fig. 23 and 24). In the event of a short drop in the supply voltage, the diode path rapidly discharges the capacitor to ensure a reliable power-on reset. To ensure a correct reset, the RESET signal should reach at least 70% of the final value of V_{DD}. Given that the RESET voltage and V_{DD} rise exponentially, the above requirement is satisfied when the time constant τ of the RESET pulse is >8 times the time constant of V_{DD}. If V_{DD} rises linearly, then a RESET time constant >2 times the rise time of V_{DD} is required.

   When a reset is completed (RESET goes LOW) before the oscillator has started up, program execution begins after the oscillator start-up time plus 1866 clock periods have elapsed (see Fig. 24). If the oscillator is started-up prior to the completion of RESET, then program execution begins 1866 clock periods after RESET goes LOW.

3. Figure 25 shows an external reset to the PCF84C85 during power-on. The external reset signal must remain HIGH until V_{DD} has reached its minimum operating value corresponding to the selected oscillator frequency. When a reset is completed (RESET goes LOW) before the oscillator has started up, program execution begins after the oscillator start-up time plus 1866 clock periods have elapsed (see Fig. 26). If the oscillator is started-up prior to the completion of RESET, then program execution begins 1866 clock periods after RESET goes LOW.

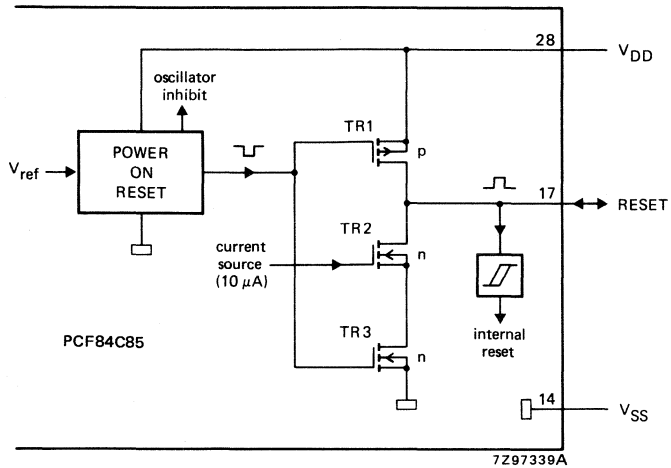


Fig. 21 Power-on-reset configuration.

DEVELOPMENT DATA

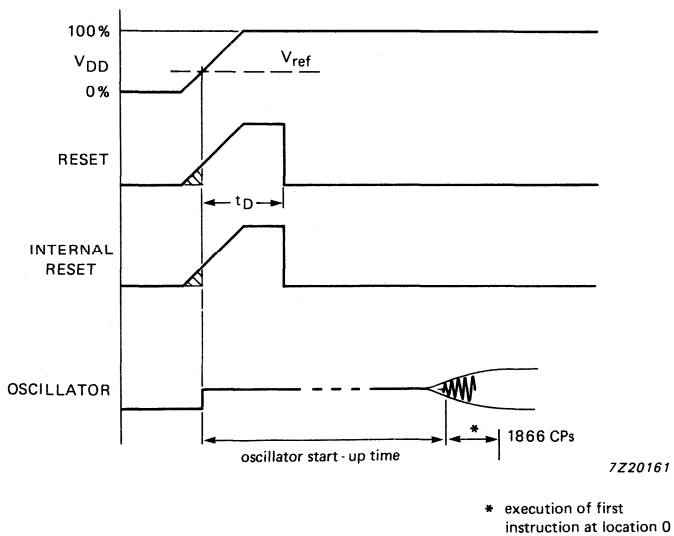


Fig. 22 Timing of power-on-reset with fast rise time.

FUNCTIONAL DESCRIPTION (continued)

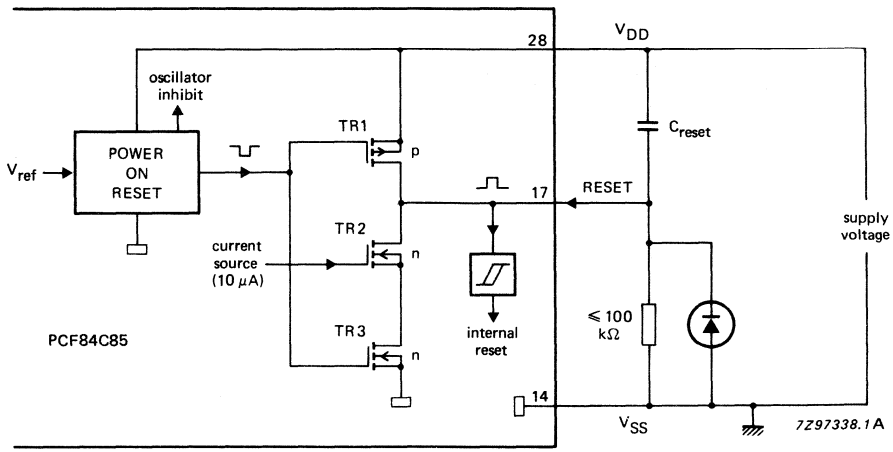


Fig. 23 Stretched power-on-reset with external components.

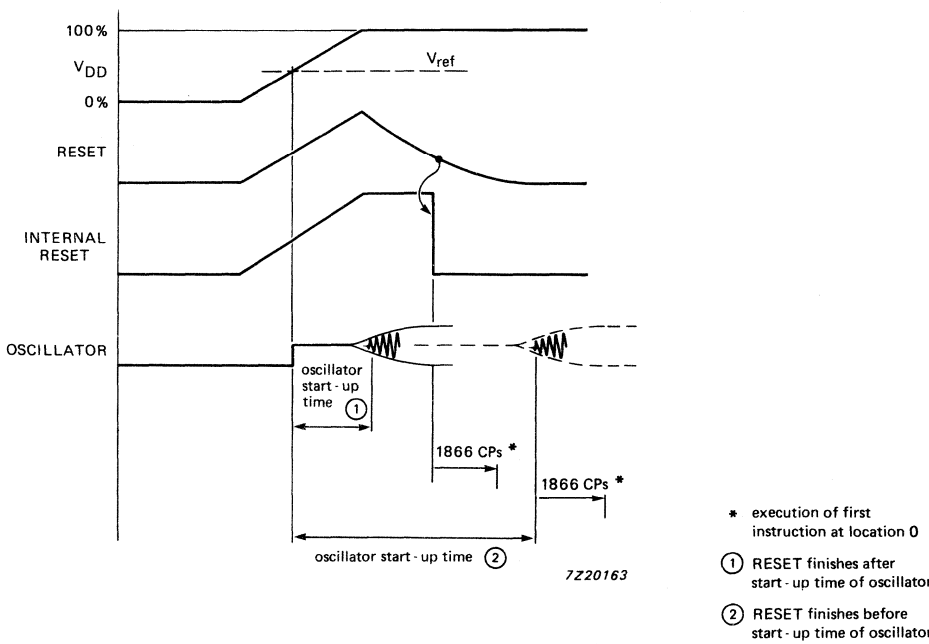


Fig. 24 Timing of power-on-reset with a slowly rising V_{DD} and a stretched RESET pulse.

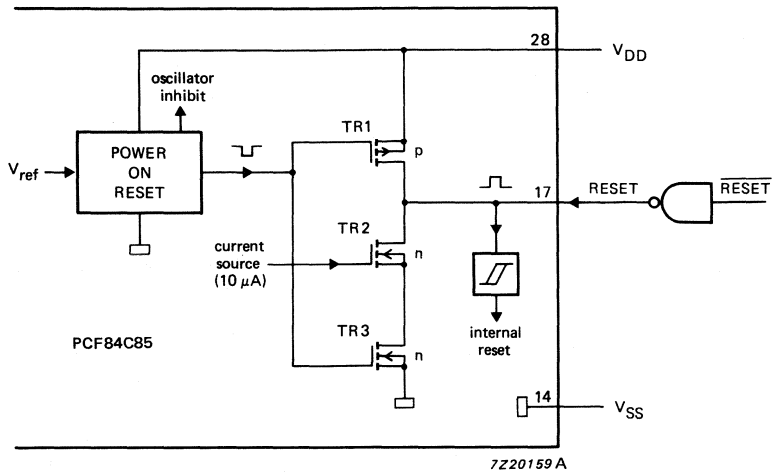


Fig. 25 External power-on-reset configuration.

DEVELOPMENT DATA

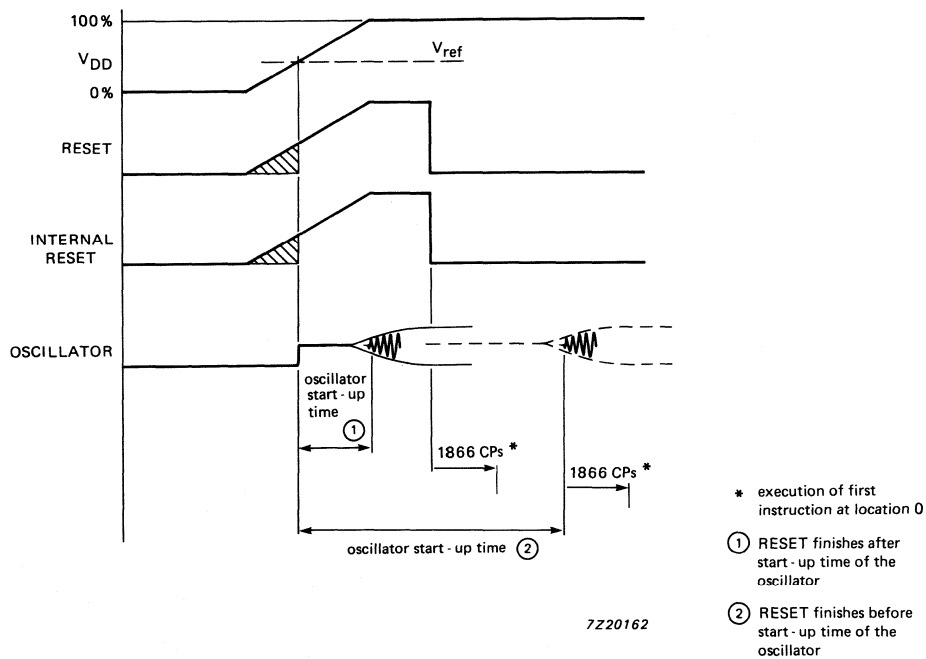


Fig. 26 Timing of external power-on-reset.

INSTRUCTION SET

The PCF84C85 instruction set consists of over 80 one and two byte instructions and is identical to the MAB8400 instruction set. New instructions are added for STOP and IDLE mode. Program code efficiency is high because all RAM locations and all ROM locations on a 256 byte page require only a single byte address.

Table 7 gives the instruction set of the PCF84C85. Table 6 shows the instruction map and Table 5 details the symbols and definition descriptions that are used.

Table 5 Symbols and definitions used in Table 7

symbol	definition description
A	accumulator
addr	program memory address
Bb	bit designation (b = 0-7)
RBS	register bank select
C	carry bit (bit CY)
CNT	event counter
Dx	Derivative register designation (x = 0,1,2 or 3)
data	8-bit number or expression
I	interrupt
MB	memory bank
MBFF	memory bank flip-flop
P	mnemonic for 'in-page' operation
PC	program counter
Pp	port designation (p = 0, 1 or 2)
PSW	program status word
RB	register bank
Rr	register designation (r = 0-7)
Sn	serial I/O register
SP	stack pointer
T	timer
TF	timer flag
T0, T1	test 0 and 1 inputs
#	immediate data prefix
@	indirect address prefix
(X)	contents of X
((X))	contents of location addressed by X
←	is replaced by
↔	is exchanged with

Table 6 PCF84C85 instruction map

first hexadecimal character of opcode ↓ / second hexadecimal character of opcode →	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NOP	IDLE		ADD A,#data	JMP page 0	EN I	JNTF addr	DEC A	IN A,Pp				MOV A,Sn			
									0	1	2		0	1		
1	INC @Rr		JBO addr	ADDC A,#data	CALL page 0	DIS I	JTF addr	INC A	INC Rr							
	0	1							0	1	2	3	4	5	6	7
2	XCH A,@Rr		STOP	MOV A,#data	JMP page 1	EN TCNTI	JNTO addr	CLR A	XCH A,Rr							
	0	1							0	1	2	3	4	5	6	7
3	XCHD A,@Rr		JB1 addr		CALL page 1	DIS TCNTI	JTO addr	CPL A	OUTL Pp,A				MOV Sn,A			
	0	1							0	1	2		0	1	2	
4	ORL A,@Rr		MOV A,T	ORL A,#data	JMP page 2	STRT CNT	JNT1 addr	SWAP A	ORL A,Rr							
	0	1							0	1	2	3	4	5	6	7
5	ANL A,@Rr		JB2 addr	ANL A,#data	CALL page 2	STRT T	JT1 addr	DA A	ANL A,Rr							
	0	1							0	1	2	3	4	5	6	7
6	ADD A,@Rr		MOV T,A		JMP page 3	STOP TCNT		RRC A	ADD A,Rr							
	0	1							0	1	2	3	4	5	6	7
7	ADDC A,@Rr		JB3 addr		CALL page 3			RR A	ADDC A,Rr							
	0	1							0	1	2	3	4	5	6	7
8				RET	JMP page 4	EN SI			ORL Pp,#data				MOV A,Dx	MOV Dx,A	ANL Dx,A	ORL Dx,A
									0	1	2					
9			JB4 addr	RETR	CALL page 4	DIS SI	JNZ addr	CLR C	ANL Pp,#data				MOV Sn,#data			
									0	1	2		0	1	2	
A	MOV @Rr,A			MOVP A,@A	JMP page 5	SEL MB2		CPL C	MOV Rr,A							
	0	1							0	1	2	3	4	5	6	7
B	MOV @Rr,#data		JB5 addr	JMPP @A	CALL page 5	SEL MB3			MOV Rr,#data							
	0	1							0	1	2	3	4	5	6	7
C	DEC @Rr				JMP page 6	SEL RB0	JZ addr	MOV A,PSW	DEC Rr							
	0	1							0	1	2	3	4	5	6	7
D	XRL A,@Rr		JB6 addr	XRL A,#data	CALL page 6	SEL RB1		MOV PSW,A	XRL A,Rr							
	0	1							0	1	2	3	4	5	6	7
E	DJNZ @Rr,addr				JMP page 7	SEL MB0	JNC addr	RL A	DJNZ Rr, addr							
	0	1							0	1	2	3	4	5	6	7
F	MOV A,@Rr		JB7 addr		CALL page 7	SEL MB1	JC addr	RLC A	MOV A,Rr							
	0	1							0	1	2	3	4	5	6	7

INSTRUCTION SET (continued)

Table 7 Instruction set

	mnemonic	opcode (hex.)	bytes/ cycles	description	function		notes
ACCUMULATOR	ADD A, Rr	6*	1/1	Add register contents to A	(A)←(A) + (Rr)	r = 0–7	1
	ADD A, @Rr	60 61	1/1	Add RAM data, addressed by Rr, to A	(A)←(A) + ((R0)) (A)←(A) + ((R1))		1
	ADD A, #data	03 data	2/2	Add immediate data to A	(A)←(A) + data		1
	ADDC A, Rr	7*	1/1	Add carry and register contents to A	(A)←(A) + (Rr) + (C)	r = 0–7	1
	ADDC A, @Rr	70 71	1/1	Add carry and RAM data, addressed by Rr, to A	(A)←(A) + ((R0)) + (C) (A)←(A) + ((R1)) + (C)		1
	ADDC A, #data	13 data	2/2	Add carry and immediate data to A	(A)←(A) + data + (C)		1
	ANL A, Rr	5*	1/1	'AND' Rr with A	(A)←(A) AND (Rr)	r = 0–7	
	ANL A, @Rr	50 51	1/1	'AND' RAM data, addressed by Rr, with A	(A)←(A) AND ((R0)) (A)←(A) AND ((R1))		
	ANL A, #data	53 data	2/2	'AND' immediate data with A	(A)←(A) AND data		
	ORL A, Rr	4*	1/1	'OR' Rr with A	(A)←(A) OR (Rr)	r = 0–7	
	ORL A, @Rr	40 41	1/1	'OR' RAM data, addressed by Rr, with A	(A)←(A) OR ((R0)) (A)←(A) OR ((R1))		
	ORL A, #data	43 data	2/2	'OR' immediate data with A	(A)←(A) OR data		
	XRL A, Rr	D*	1/1	'XOR' Rr with A	(A)←(A) XOR (Rr)	r = 0–7	
	XRL A, @Rr	D0 D1	1/1	'XOR' RAM, addressed by Rr, with A	(A)←(A) XOR ((R0)) (A)←(A) XOR ((R1))		
	XRL A, #data	D3 data	2/2	'XOR' immediate data with A	(A)←(A) XOR data		
	INC A	17	1/1	increment A by 1	(A)←(A) + 1		
	DEC A	07	1/1	decrement A by 1	(A)←(A) −1		
	CLR A	27	1/1	clear A to zero	(A)←0		
	CPL A	37	1/1	one's complement A	(A)←NOT(A)		
	RL A	E7	1/1	rotate A left	$(A_{n+1})\leftarrow(A_n)$ $(A_0)\leftarrow(A_7)$	n = 0–6	

DEVELOPMENT DATA

ACCUMULATOR (cont.)	RLC A	F7	1/1	rotate A left through carry	$(A_{n+1}) \leftarrow A_n$ $(A_0) \leftarrow (C), (C) \leftarrow (A_7)$	n = 0–6	2
	RR A	77	1/1	rotate A right	$(A_n) \leftarrow (A_{n+1})$ $(A_7) \leftarrow (A_0)$	n = 0–6	
	RRC A	67	1/1	rotate A right through carry	$(A_n) \leftarrow (A_{n+1})$ $(A_7) \leftarrow (C), (C) \leftarrow (A_0)$	n = 0–6	2
	DA A	57	1/1	decimal adjust A			2
	SWAP A	47	1/1	swap nibbles of A	$(A_{4-7}) \leftrightarrow (A_{0-3})$		
DATA MOVES	MOV A, Rr	F*	1/1	move register contents to A	(A)←(Rr)	r = 0–7	
	MOV A, @Rr	F0 F1	1/1	move RAM data, addressed by Rr, to A	(A)←((R0)) (A)←((R1))		
	MOV A, #data	23 data	2/2	move immediate data to A	(A)←data		
	MOV Rr, A	A*	1/1	move accumulator contents to register	(Rr)←(A)	r = 0–7	
	MOV @Rr, A	A0 A1	1/1	move accumulator contents to RAM location addressed by Rr	((R0))←(A) ((R1))←(A)		
	MOV Rr, #data	B* data	2/2	move immediate data to Rr	(Rr)←data		
	MOV @Rr, #data	B0 data B1 data	2/2	move immediate data to RAM location addressed by Rr	((R0))←data ((R1))←data		
	XCH A, Rr	2*	1/1	exchange accumulator contents with Rr	(A)↔(Rr)	r = 0–7	
	XCH A, @Rr	20 21	1/1	exchange accumulator contents with RAM data addressed by Rr	(A)↔((R0)) (A)↔((R1))		
	XCHD A, @Rr	30 31	1/1	exchange lower nibbles of A and RAM data addressed by Rr	$(A_{0-3}) \leftrightarrow ((R0_{0-3}))$ $(A_{0-3}) \leftrightarrow ((R1_{0-3}))$		
	MOV A, PSW	C7	1/1	move PSW contents to accumulator	(A)←(PSW)		
	MOV PSW, A	D7	1/1	move accumulator bit 3 to PSW_3	$(PSW_3) \leftarrow (A_3)$		3
	MOVP A, @A	A3	1/2	move indirectly addressed data in current page to A	$(PC_{0-7}) \leftarrow (A), (A) \leftarrow ((PC))$		
FLAGS	CLR C	97	1/1	clear carry bit	(C)←0		2
	CPL C	A7	1/1	complement carry bit	(C)←NOT(C)		2

INSTRUCTION SET (continued)

	mnemonic	opcode (hex.)	bytes/ cycles	description	function	notes
REGISTER	INC Rr	1*	1/1	increment register by 1	(Rr)←(Rr) + 1 r = 0–7	
	INC @Rr	10 11	1/1	increment RAM data, addressed by Rr, by 1	((R0))←((R0)) + 1 ((R1))←((R1)) + 1	
	DEC Rr	C*	1/1	decrement register by 1	(Rr)←(Rr) – 1 r = 0–7	
	DEC @Rr	C0 C1	1/1	decrement RAM data, addressed by Rr, by 1	((R0))←((R0)) – 1 ((R1))←((R1)) – 1	
BRANCH	JMP addr	● 4 address	2/2	unconditional jump within a 2 K bank	(PC_{8-10})←$addr_{8-10}$ (PC_{0-7})←$addr_{0-7}$ (PC_{11-12})←MBFF 0–1	
	JMPP @A	B3	1/2	indirect jump within a page	(PC_{0-7})←((A))	
	DJNZ Rr, addr	E* address	2/2	decrement Rr by 1 and jump if not zero to addr	(Rr)←(Rr) – 1 r = 0–7 if (Rr) not zero (PC_{0-7})←addr	
	DJNZ @Rr, addr	E0 address	2/2	decrement RAM data, addressed by Rr by 1 and jump if not zero to addr	((R0))←((R0)) – 1 if ((R0)) not zero (PC_{0-7})←addr	
		E1 address			((R1))←((R1)) – 1 if ((R1)) not zero (PC_{0-7})←addr	
	JBb addr	▲ 2 address	2/2	jump to addr if Acc. bit b = 1	if b = 1 : (PC_{0-7})←addr b = 0–7	
	JC addr	F6 address	2/2	jump to addr if C = 1	if C = 1 : (PC_{0-7})←addr	
	JNC addr	E6 address	2/2	jump to addr if C = 0	if C = 0 : (PC_{0-7})←addr	
	JZ addr	C6 address	2/2	jump to addr if A = 0	if A = 0 : (PC_{0-7})←addr	
	JNZ addr	96 address	2/2	jump to addr if A is NOT zero	if A ≠ 0 : (PC_{0-7})←addr	
	JT0 addr	36 address	2/2	jump to addr if T0 = 1	if T0 = 1: (PC_{0-7})←addr	
	JNT0 addr	26 address	2/2	jump to addr if T0 = 0	if T0 = 0: (PC_{0-7})←addr	
	JT1 addr	56 address	2/2	jump to addr if T1 = 1	if T1 = 1: (PC_{0-7})←addr	
	JNT1 addr	46 address	2/2	jump to addr if T1 = 0	if T1 = 0: (PC_{0-7})←addr	
	JTF addr	16 address	2/2	jump to addr if Timer Flag = 1	if TF = 1: (PC_{0-7})←addr	4
	JNTF addr	06 address	2/2	jump to addr if Timer Flag = 0	if TF = 0: (PC_{0-7})←addr	

TIMER/EVENT COUNTER	MOV A, T	42	1/1	move timer/event counter contents to accumulator	(A)←(T)	
	MOV T, A	62	1/1	move accumulator contents to timer/event counter	(T)←(A)	
	STRT CNT	45	1/1	start event counter		
	STRT T	55	1/1	start timer		
	STOP TCNT	65	1/1	stop timer/event counter		
	EN TCNTI	25	1/1	enable timer/event counter interrupt		
	DIS TCNTI	35	1/1	disable timer/event counter interrupt		
CONTROL	EN I	05	1/1	enable external interrupt		
	DIS I	15	1/1	disable external interrupt		
	SEL RB0	C5	1/1	select register bank 0	(RBS)←0	5
	SEL RB1	D5	1/1	select register bank 1	(RBS)←1	5
	SEL MB0	E5	1/1	select program memory bank 0	(MBFF0)←0, (MBFF1)←0	
	SEL MB1	F5	1/1	select program memory bank 1	(MBFF0)←1, (MBFF1)←0	
	SEL MB2	A5	1/1	select program memory bank 2	(MBFF0)←0, (MBFF1)←1	
	SEL MB3	B5	1/1	select program memory bank 3	(MBFF0)←1, (MBFF1)←1	
	STOP	22	1/1	enter STOP mode		
	IDLE	01	1/1	enter IDLE mode		
SUBROUTINE	CALL addr	▲ 4 address	2/2	jump to subroutine	$((SP))\leftarrow(PC), (PSW_{4,6,7})$ 6 $(SP)\leftarrow(SP)+1$ $(PC_{8-10})\leftarrow addr_{8-10}$ $(PC_{0-7})\leftarrow addr_{0-7}$ $(PC_{11-12})\leftarrow MBFF\ 0-1$	6
	RET	83	1/2	return from subroutine	$(SP)\leftarrow(SP)-1$ $(PC)\leftarrow((SP))$	6
	RETR	93	1/2	return from interrupt and restore bits 4, 6, 7 of PSW	$(SP)\leftarrow(SP)-1$ $(PSW_{4,6,7})+(PC)\leftarrow((SP))$	6

INSTRUCTION SET (continued)

	mnemonic	opcode (hex.)	bytes/ cycles	description	function	notes
PARALLEL INPUT/OUTPUT	IN A, Pp	08 09 0A	1/2	input port p data to accumulator	(A)←(P0) (A)←(P1) (A)←(P2)	7
	OUTL Pp, A	38 39 3A	1/2	output accumulator data to port p	(P0)←(A) (P1)←(A) (P2)←(A)	
	ANL Pp, #data	98 data 99 data 9A data	2/2	AND port p data with immediate data	(P0)←(P0) AND data (P1)←(P1) AND data (P2)←(P2) AND data	
	ORL Pp, #data	88 data 89 data 8A data	2/2	OR port p data with immediate data	(P0)←(P0) OR data (P1)←(P1) OR data (P2)←(P2) OR data	
DERIVATIVE INPUT/OUTPUT	MOV A, Dx	8C 00 8C 01	2/2 2/2	input pin data of port DP0, DP1 to accumulator	(A)←(D0) (A)←(D2)	8
	MOV Dx, A	8D 02 8D 03	2/2 2/2	move contents of accumulator to latch of port DP0, DP1	(D2)←(A) (D3)←(A)	
	ANL Dx, A	8E 02 8E 03	2/2 2/2	AND contents of DP0, DP1 latch with accumulator	(D2)←(D2) AND (A) (D3)←(D2) AND (A)	
	ORL Dx, A	8F 02 8F 03	2/2 2/2	OR contents of DP0, DP1 latch with accumulator	(D2)←(D2) OR (A) (D3)←(D3) OR (A)	
	MOV A, Dx	8C 02 8C 03	2/2 2/2	move contents of DP0, DP1 latch to accumulator	(A)←(D2) (A)←(D3)	8

	mnemonic	opcode (hex.)	bytes/ cycles	description	function	notes
SERIAL INPUT/OUTPUT	MOV A, S_n	0C 0D	1/2	move serial I/O register contents to accumulator	(A)←(S0) (A)←(S1)	9
	MOV S_n, A	3C 3D 3E	1/2	move accumulator contents to serial I/O register	(S0)←(A) (S1)←(A) (S2)←(A)	
	MOV S_n, #data	9C data 9D data 9E data	2/2	move immediate data to serial I/O register	(S0)←data (S1)←data (S2)←data	
	EN SI	85	1/1	enable serial I/O interrupt		
	DIS SI	95	1/1	disable serial I/O interrupt		
	NOP	00	1/1	no operation		

Notes to Table 8

1. PSW CY, AC affected
2. PSW CY affected
3 PSW PS affected
4. Execution of JTF and JNTF instructions resets the Timer Flag (TF).

* : 8,9,A,B,C,D,E,F
● : 9,2,4,6,8,A,C,E
▲ : 1,3,5,7,9,B,D,F

5. PSW RBS affected
6. PSW SP_0, SP_1, SP_2 affected
7. (A) = 0000 P23 111
8. The MSB of A becomes a logic 1
9. (S1) has a different function in read and write operations, see serial I/O interface.

RATINGS

Limiting values in accordance with the Absolute Maximum System (IEC 134)

Supply voltage (pin 40)	V_{DD}		−0,8 to +8	V
All input voltages	V_I		−0,8 to V_{DD} +0,8	V
D.C. current into any input or output	$\pm I_I$, $\pm I_O$	max.	10	mA
Total power dissipation (see note)				
Power dissipation per output				
except P23, SCLK	P_O	max.	50	mW
P23, SCLK	P_O	max.	180	mW
Storage temperature range	T_{stg}		−65 to +150	°C
Operating ambient temperature range	T_{amb}		−40 to +85	°C
Operating junction temperature	T_j	max.	125	°C

HANDLING

Inputs and outputs are protected against electrostatic discharge in normal handling. However, to be totally safe, it is desirable to take normal precautions appropriate to handling MOS devices (see 'Handling MOS devices').

DEVELOPMENT DATA

D.C. CHARACTERISTICS

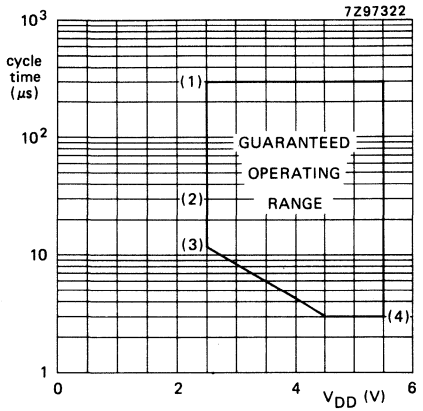
V_{DD} = 2,5 to 5,5 V; V_{SS} = 0 V; T_{amb} = −40 to + 85 °C; all voltages with respect to V_{SS}; unless otherwise specified.

parameter	symbol	min.	typ.	max.	unit
Supply voltage					
operating (see Fig. 27)	V_{DD}	2,5	–	5,5	V
Supply current					
operating (see Fig. 28)					
at V_{DD} = 5 V; f_{XTAL} = 6 MHz	I_{DD}	–	1	2	mA
at V_{DD} = 3 V; f_{XTAL} = 3,58 MHz	I_{DD}	–	0,3	0,6	mA
IDLE mode (see Fig. 30)					
at V_{DD} = 5 V; f_{XTAL} = 6 MHz	I_{DD}	–	0,5	1	mA
at V_{DD} = 3 V; f_{XTAL} = 3,58 MHz	I_{DD}	–	0,15	0,4	mA
STOP mode (see Fig. 35 and note 1)					
at V_{DD} = 2,5 V; T_{amb} = 25 °C	I_{DD}	–	1,2	2,5	μA
at V_{DD} = 2,5 V; T_{amb} = 85 °C	I_{DD}	–	–	10	μA
RESET I/O					
Switching level	V_{RESET}	–	1,5	2	V
Sink current					
at $V_{DD} > V_{RESET}$	I_{OL}	–	7	–	μA
Inputs					
Input voltage LOW	V_{IL}	0	–	$0,3V_{DD}$	V
Input voltage HIGH	V_{IH}	$0,7V_{DD}$	–	V_{DD}	V
Input leakage current					
as $V_{SS} < V_I < V_{DD}$	$\pm I_{IL}$	–	–	1	μA
Outputs					
Output voltage LOW					
at V_I = V_{SS} or V_{DD}; $\|I_O\| < 1$ μA	V_{OL}	–	–	0,05	V
Output sink current LOW					
at V_{DD} = 5 V ± 10%; V_O = 0,4 V					
except P23/SDA, SCLK (see Fig. 31)	I_{OL}	1,6	3	–	mA
P23/SDA, SCLK (see Fig. 32)	I_{OL}	3	–	–	mA
Pull-up output source current HIGH (see Fig. 33)					
at V_{DD} = 5 V ± 10%; $V_O = 0,7V_{DD}$	$-I_{OH}$	40	–	–	μA
at V_{DD} = 5 V ± 10%; $V_O = V_{SS}$	$-I_{OH}$	–	–	400	μA
Push-pull output source current HIGH					
at V_{DD} = 5 V ± 10%; $V_O = V_{DD}$−0,4 V	$-I_{OH}$	1,6	3	–	mA

Note 1

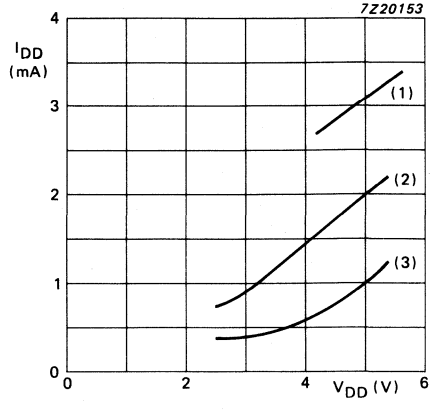
Crystal connected between XTAL1 and XTAL2; SCL and SDA pulled to V_{DD} via 5,6 kΩ resistor; T1 at V_{SS}, $\overline{INT}$ at V_{DD}.

A.C. CHARACTERISTICS (continued)



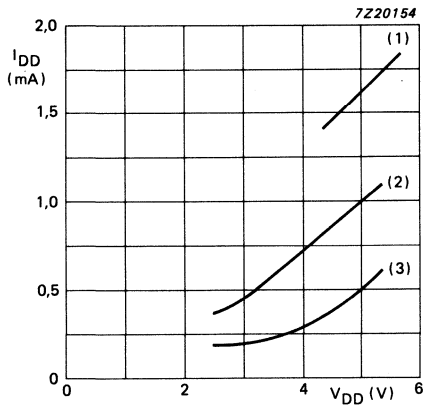
(1) clock frequency = 100 kHz
(2) clock frequency = 1 MHz
(3) clock frequency = 3 MHz
(4) clock frequency = 10 MHz

Fig. 27 Maximum clock frequency (f_{XTAL}) as a function of the supply voltage (V_{DD}).



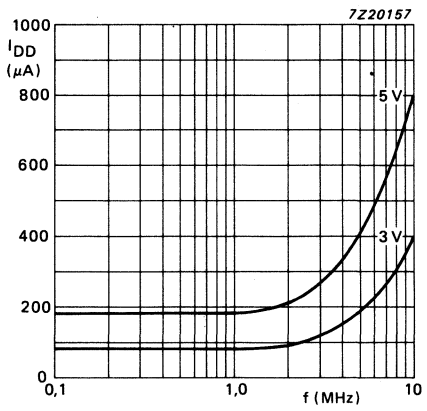
(1) clock frequency = 10 MHz
(2) clock frequency = 6 MHz
(3) clock frequency = 3,58 MHz

Fig. 28 Maximum supply current (I_{DD}) in operating mode as a function of the supply voltage.



(1) clock frequency = 10 MHz
(2) clock frequency = 6 MHz
(3) clock frequency = 3,58 MHz

Fig. 29 Maximum supply current (I_{DD}) in IDLE mode as a function of the supply voltage (V_{DD}).



(1) V_{DD} = 3 V
(2) V_{DD} = 5 V

Fig. 30 Typical supply current during IDLE mode as a function of frequency.

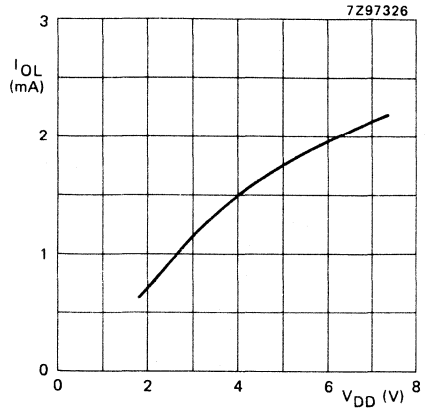


Fig. 31 Output sink current LOW (I_{OL}), except outputs P23/SDA and SCLK, as a function of supply voltage (V_{DD}); V_O = 0,4 V.

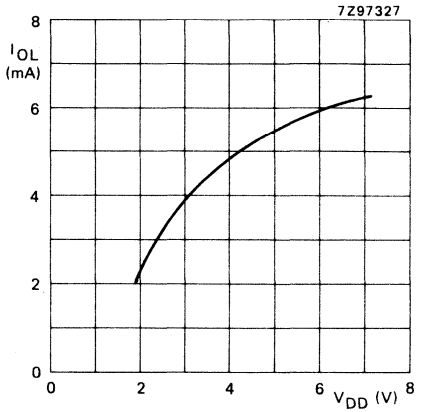
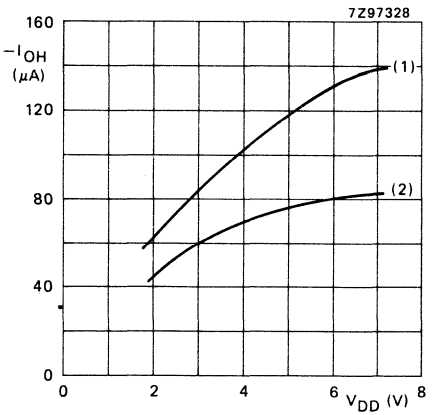


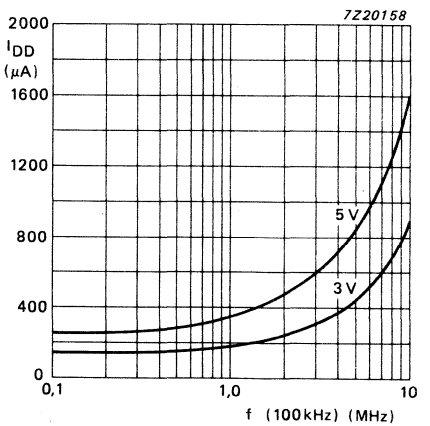
Fig. 32 Output current LOW (I_{OL}), outputs P23/SDA and SCLK, as a function of supply voltage (V_{DD}); V_O = 0,4 V.

DEVELOPMENT DATA



(1) $V_O = V_{SS}$
(2) $V_O = 0,7\ V_{DD}$

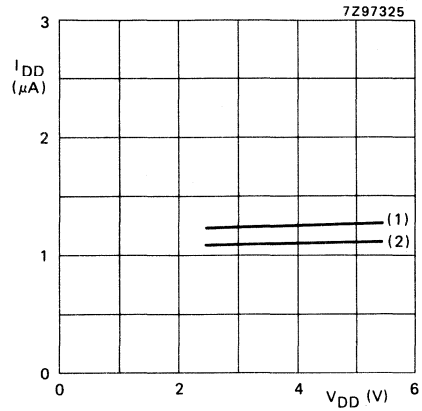
Fig. 33 Output source current HIGH ($-I_{OH}$) as a function of supply voltage (V_{DD}).



(1) V_{DD} = 3 V
(2) V_{DD} = 5 V

Fig. 34 Typical supply current during operating mode as a function of frequency.

A.C. CHARACTERISTICS (continued)



(1) T_{amb} = 85 °C
(2) T_{amb} = 25 °C

Fig. 35 Typical supply current (I_{DD}) in STOP mode as a function of the supply voltage (V_{DD}).

Table 8 Input timing shown in figure 36.

symbol	timing
t_{BUF}	$\geqslant 14t_{XTAL}$
$t_{HD;STA}$	$\geqslant 14t_{XTAL}$
t_{HIGH}	$\geqslant 17t_{XTAL}$
t_{LOW}	$\geqslant 17t_{XTAL}$
$t_{SU;STO}$	$\geqslant 14t_{XTAL}$
$t_{HD;DAT}$	> 0
$t_{SU;DAT}$	$\geqslant 250$ ns
t_{RD}	$\leqslant 1\ \mu s$
t_{RC}	$\leqslant 1\ \mu s$
t_{FD}	$\leqslant 1\ \mu s$
t_{FC}	$\leqslant 0{,}3\ \mu s$

Notes to Table 8

t_{XTAL} = one period of the XTAL input frequency (f_{XTAL})
= 167 ns for f_{XTAL} = 6 MHz.
These figures apply to all modes.

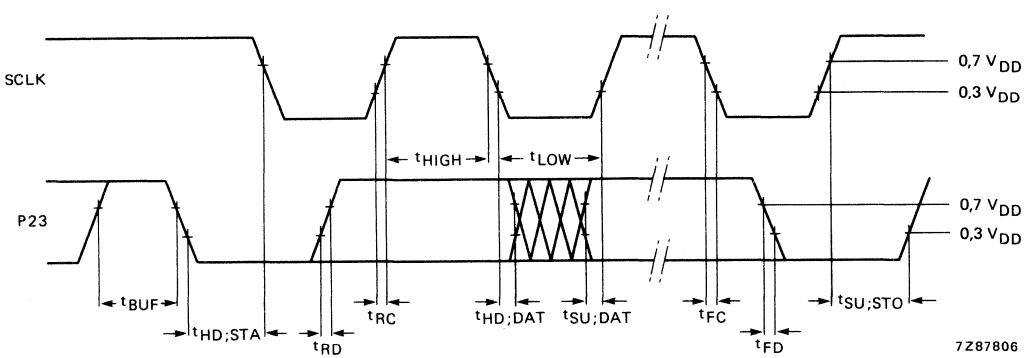


Fig. 36 PCF84C85 timing requirements for the P23 and SCLK *input* signals.

Purchase of Philips' I²C components conveys a license under the Philips' I²C patent to use the components in the I²C-system provided the system conforms to the I²C specifications defined by Philips.

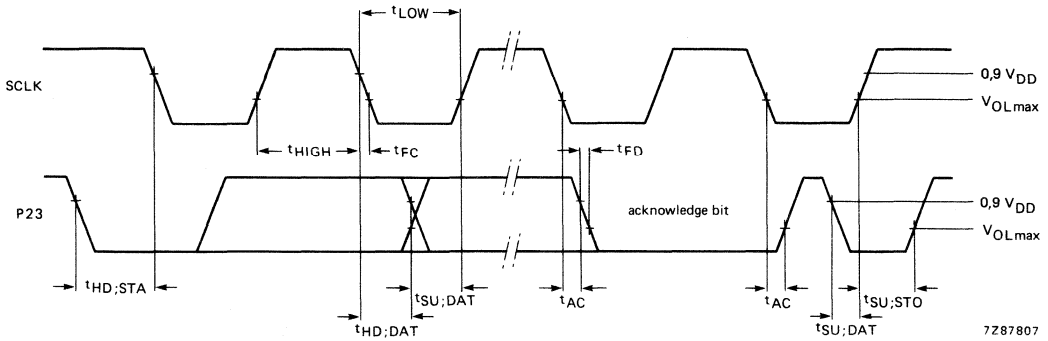


Fig. 37 PCF84C85 timing requirements for the P23 and SCLK *output* signals.

Table 9 Output timing shown in Figure 37

symbol	timing	
	normal mode (ASC in S2 = 0)	low-speed mode (ASC in S2 = 1)
$t_{HD;STA}$	½ (DF + 9) t_{XTAL}	¾ (DF + 9) t_{XTAL}
t_{HIGH}	½ (DF) t_{XTAL}	¾ (DF) t_{XTAL}
t_{LOW}	½ (DF) t_{XTAL}	¼ (DF) t_{XTAL}
$t_{SU;STO}$	½ (DF − 3) t_{XTAL}	¼ (DF − 3) t_{XTAL}
$t_{HD;DAT}$ (slave transmitter)		
any DF	$\geqslant 9t_{XTAL}$ $\leqslant 12t_{XTAL}$	$\geqslant 9t_{XTAL}$ $\leqslant 12t_{XTAL}$
$t_{HD;DAT}$ (master transmitter)		
for DF $\leqslant$ 51	$\geqslant 9t_{XTAL}$ $\leqslant 12t_{XTAL}$	– –
for DF $\leqslant$ 99	– –	$\geqslant 9t_{XTAL}$ $\leqslant 12t_{XTAL}$
$t_{SU;DAT}$ (master transmitter)		
for DF > 51	$\geqslant 15t_{XTAL}$ $\leqslant 24t_{XTAL}$	– –
for DF > 99	–	$\geqslant 15t_{XTAL}$
t_{AC}	$\geqslant 9t_{XTAL}$ $\leqslant 12t_{XTAL}$	$\geqslant 9t_{XTAL}$ $\leqslant 12t_{XTAL}$
t_{FD}, t_{FC}	$\leqslant$ 100 ns at C_b = 400 pF	$\leqslant$ 100 ns at C_b = 400 pF

Notes to Table 9

t_{XTAL} = one period of the XTAL input frequency (f_{XTAL})
= 167 ns for f_{XTAL} = 6 MHz.
DF = divisor (see Table 2 Serial I/O section).
C_b = the maximum bus capacitance for each line.

DEVELOPMENT DATA

SINGLE-CHIP 8-BIT MICROCONTROLLERS (bipolar)

STANDARD PRODUCTS

MICROCONTROLLER

Originally published by Signetics January 1984

FEATURES

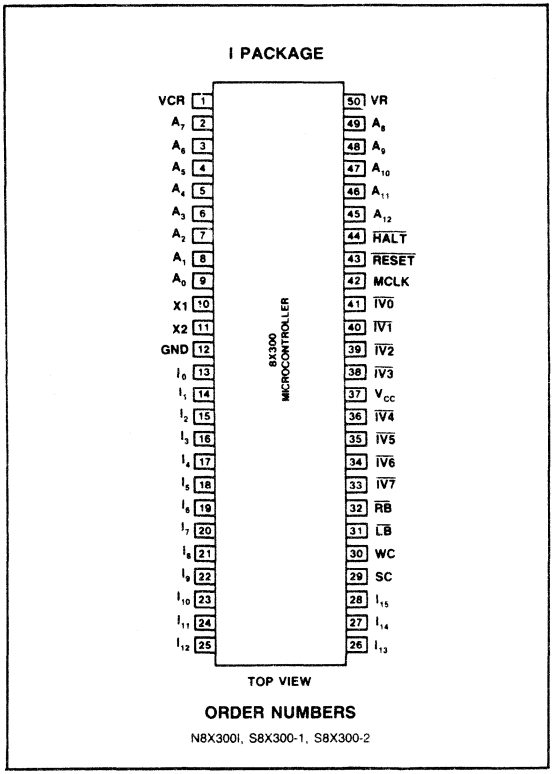
- **Fetch, Decode, and Execute a 16-bit instruction in a minimum of 250-nanoseconds (one machine cycle)**
- **Bit-oriented instruction set (addressable single-or-multiple bit subfields)**
- **Separate address, instruction, and I/O buses**
- **Source/destination architecture**
- **On-Chip oscillator and timing generation**
- **Eight 8-bit working registers**
- **TTL inputs and outputs**
- **BiPolar Low-Power Schottky technology**
- **3-State I/O bus**
- **Single +5V supply**

ARCHITECTURAL OVERVIEW

The Signetics 8X300 Microcontroller (Figure 1) is a high-speed bipolar microprocessor implemented with low-power Schottky technology. The 8X300 brings together all the qualities needed—SPEED, FLEXIBILITY, and ECONOMY—for systems design in the many areas that require reliable bit stream management. Consider!—5V operation, TTL bus compatibility, and an on-chip clock—the result, a system with fewer parts. Consider!—the inherent power of LSI logic (programmable Rotate, Mask, Shift, and Merge functions in the data-processing path) and the ability to Fetch, Decode, and Execute a 16-bit instruction in a minimum of 250-nanoseconds—the result, a system with superior bit handling capabilities. Consider!—the 250ns cycle time in conjunction with extended microcode—the result, the flexibility of bit-slice devices with the programming ease of MOS microprocessors. Now, consider the results!—a device tailored to bit-stream management in the areas of Industrial Control, Input/Output Control, and Data Communications.

The 8X300 uses three separate buses—one for 13-bit instruction addresses, one for 16-bit instructions, and a bidirectional 8-bit input/output data bus; except for the I/O bus, there are no time multiplexing of functions.

PIN CONFIGURATION



ORDER NUMBERS

N8X300I, S8X300-1, S8X300-2

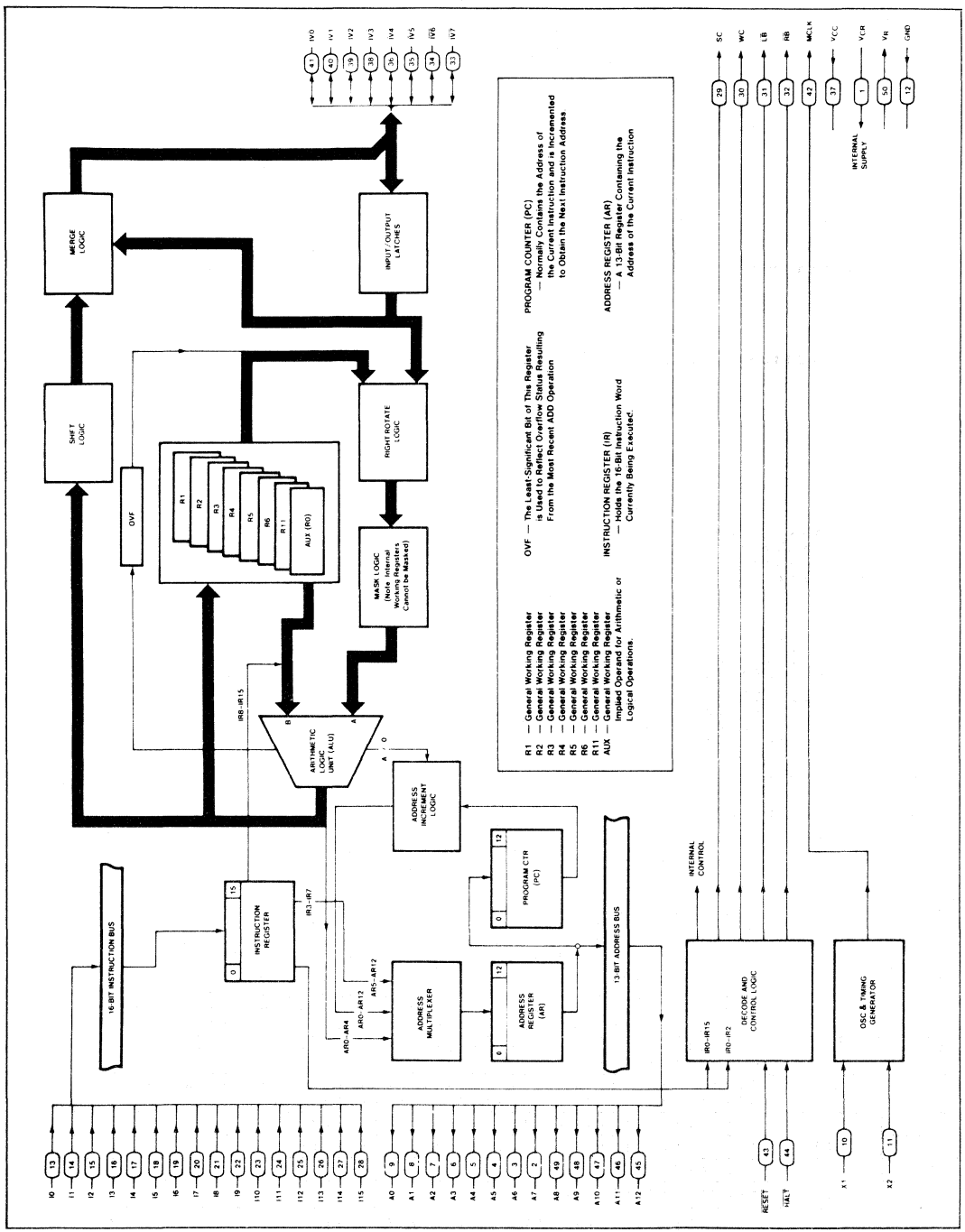
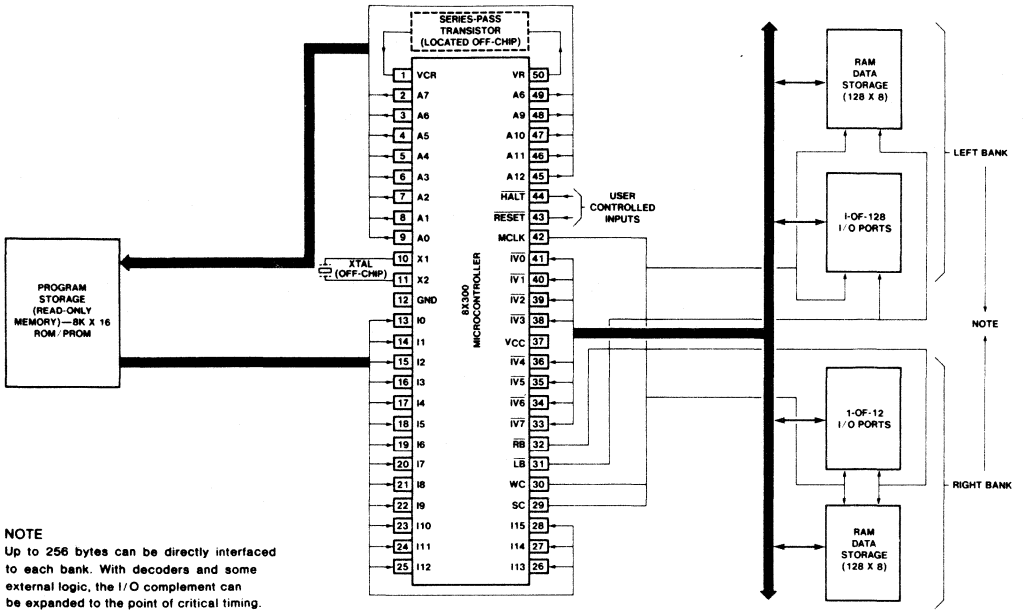


Figure 1. CPU Architecture and PIN Designations For 8X300 Microcontroller



NOTE
Up to 256 bytes can be directly interfaced to each bank. With decoders and some external logic, the I/O complement can be expanded to the point of critical timing.

PIN NO.	IDENTIFIER	NAME AND FUNCTION	ACTIVE STATE
2-9/45-49	A0-A12	**Program Address Lines:** These outputs permit direct addressing of up to 8192 words of program storage. A high voltage level equals a binary "1"; A12 is Least Significant Bit.	High
13-28	I0-I15	**Instruction Lines:** These input lines receive 16-bit instructions from program storage. A high voltage level equals a binary "1"; I15 is Least Significant Bit.	High
33-36 38-41	$\overline{IV0}$-$\overline{IV7}$	**Input/Output Bus:** These bidirectional three-state lines communicate with up to 512 I/O devices (256 per bank). A low voltage level equals a binary "1"; $\overline{IV7}$ is Least Significant Bit.	Low
10 & 11	X1 & X2	**Connections** for a capacitor, a series-resonant crystal, or an external clock source with complementary outputs. For precise frequency control, a crystal or external source is required.	—
42	MCLK	**Master Clock:** This output is used for clocking I/O devices and/or synchronization of external logic.	High
30	WC	**Write Command:** When signal is high (binary 1), data is being output on pins $\overline{IV0}$-$\overline{IV7}$ of I/O bus.	High
29	SC	**Select Command:** When signal is high (binary 1), an address is being output on pins $\overline{IV0}$-$\overline{IV7}$ of I/O bus.	High
31	$\overline{LB}$	When the $\overline{LB}$ signal is low (binary 0), any one of up-to-256 I/O devices (or memory locations) in the left bank can be accessed. When the address of a particular device (or memory location) matches the address on the IV bus, that particular device (or memory location) is enabled and selected for input/output operations. All addresses on the left bank that do not match are deselected.	Low
32	$\overline{RB}$	When the $\overline{RB}$ signal is low (binary 0), any one of up-to-256 I/O devices (or memory locations) in the right bank can be accessed. When the address of a particular device (or memory location) matches the address on the I/O bus, that particular device (or memory location) is enabled and selected for input/output operations. All addresses on the right bank that do not match are deselected.	Low
43	$\overline{RESET}$	When reset input is low (binary 0), the microcontroller is initialized—sets Program Counter/Address to zero and inhibits MCLK output.	Low
44	$\overline{HALT}$	When halt input is low (binary 0), internal operation of microcontroller stops at the start of next instruction. The stop function does not inhibit MCLK or affect any internal registers.	Low
50	VR	Internally-generated reference output voltage for external series-pass transistor.	—
1	VCR	Regulated voltage input from series-pass transistor (2N5320 or equivalent).	—
12	GND	Circuit ground.	—
37	V_{CC}	Input connection for +5V power.	—

Figure 2. Typical 8X300 System with Pin Definitions

TYPICAL 8X300 SYSTEM HOOKUP

Although the system hookup shown in Figure 2 is of the simplest form, it provides a fundamental look at the 8X300 microcontroller and peripheral relationships. As indicated, program storage can be either ROM or PROM and, by using various addressing-methods/decoding-schemes, memory paging techniques can be easily implemented. Also, by proper bit assignment, some external interface logic and, under software control, the program memory can be used as a storage device for interrupt-service subroutines. The user interface ($\overline{IV0}$ through $\overline{IV7}$) is capable of addressing 256 Input/Output ports and, with the additional bank-select bit ($\overline{LB}$ and $\overline{RB}$), the number of addressable I/O ports is 512—the left bank and right bank each consisting of 256 ports. The I/O ports of each bank can be used in a variety of ways; one of these ways is shown in Figure 2. When $\overline{LB}$ is active low, the left bank can be enabled and, providing there is an address match, anyone of 128 I/O ports or anyone of 128 locations within the RAM memory can be accessed for input/output operations. When $\overline{RB}$ is active low, the same set of conditions are applicable to the right bank. With some sacrifice in speed, any given I/O port can be interfaced to a memory peripheral or other I/O device of the user.

PROGRAM STORAGE INTERFACE

As shown in Figure 2, program storage is connected to output address lines A0 through A12 (A12 = LSB) and input instruction lines I0 through I15. An address output on A0/A12 identifies one 16-bit instruction word in program storage. The instruction word is subsequently input on I0/I15 and defines the microcontroller operations which are to follow.

The Signetics 82S115 PROM or any TTL-compatible memory can be used for program storage. (Note. The worst-case access time depends upon the instruction cycle time, and also, the overall system configuration.)

I/O INTERFACE AND CONTROL

An 8-bit I/O data bus is used by the microcontroller to communicate with two fields of I/O devices. The complementary $\overline{LB}$ and $\overline{RB}$ signals identify which field of the I/O devices is enabled.

Both data and address information are output on the I/O bus. The SC (Select Command) and WC (Write Command) signals distinguish between data and address information as follows:

SC	WC	FUNCTION
High	Low	I/O address is being output on the I/O (IV) bus
Low	High	I/O data is being output on the I/O (IV) bus
Low	Low	Input data expected from selected I/O device
High	High	Invalid (not generated by 8X300)

DATA PROCESSING

From a data processing point of view, the 8X300 microcontroller chip (Figure 1) contains eight 8-bit working registers (R1 through R6, R11, AUXiliary), an arithmetic logic unit (ALU), an overflow register (OVF), rotate/shift/mask/merge logic, and a bidirectional 8-bit I/O bus. Internal 8-bit data paths connect the registers and I/O bus to the ALU inputs, and the ALU output to the registers and I/O bus. Inputs to the ALU are preceded by the data-rotate and data-mask logic and the ALU output is followed by the shift and merge logic. Any one or all of the logic functions can operate on 8-bits of data in a single instruction cycle. Data from the source register can be right-rotated (end around) before processing by the ALU; external data (I/O bus) can also be masked to isolate a portion of the 8-bit field. Since the ALU always processes 8-bits of data, bit positions not specified by the mask operation are filled with zeroes.

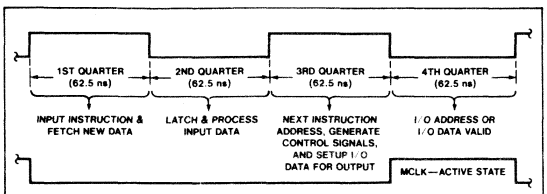
When less than 8-bits of data are specified as output to the I/O bus from the ALU, the data field (shifted and masked, as required) is merged with prior contents of the I/O latches to form the output data. Bit positions of the I/O data not affected by the logic operations are not modified. Depending upon whether an I/O peripheral or an internal register is specified in the instruction as the source of data, the I/O latches contain, respectively, I/O-bus source data or destination data. For instance, when an internal register is specified as a source of data and an I/O peripheral as the destination, data from the peripheral is read into the I/O latches at the start of the instruction cycle; processed data is then merged with contents of the I/O latches to form the I/O output data at the end of the instruction cycle. When an I/O peripheral is specified as both data source and destination, data from the source is used both as the input to the I/O latches and as data to be processed; the processed data is then merged with data from the I/O latches to form the previously-described I/O bus output. If the data source and destination are on opposite banks of the 8X300 bus, the destination data is written with a full 8-bits, since the prior contents were not stored in the I/O latches.

INSTRUCTION CYCLE

Each microcontroller operation is executed in a single instruction cycle. The instruction cycle is divided into quarters with each quarter cycle being as short as 62.5-nanoseconds. Figure 3 shows the general functions that occur during each quarter cycle; specifics regarding minimum/maximum timing and other critical values are described under "Design Parameters" in this data sheet. During the first quarter cycle, a new instruction from program storage is input on signal lines I0 through I15; simultaneously, new data is fetched via the input/output bus ($\overline{IV0}$ through $\overline{IV7}$). At the end of the first quarter cycle, the new instruction is latched in the instruction register and the new I/O data is present at the input of the chip but is not, as yet, latched by the IV latches.

In the second quarter cycle, the I/O data stabilizes and preliminary processing is completed; at the end of this quarter, the IV latches are closed and final processing can be accomplished. During the third quarter cycle, the address for the next instruction is output to the I/O (IV) bus, control signals are generated, and I/O data is setup for the output

phase. During the fourth quarter cycle, a master clock signal (MCLK) generated by the 8X300 is used to latch valid address or data into peripheral devices connected to the IV bus; MCLK is also used to synchronize any external logic with timing circuits of the 8X300. To summarize the action, the first half of the instruction cycle deals primarily with input functions and the second half is mostly concerned with output functions.



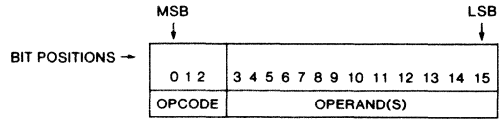
NOTES
1. New instruction must be accepted and latched at end of first quarter cycle.
2. The I/O data latches are open for the first two quarter cycles, that is, for 125-nanoseconds.
3. The address is output during third quarter cycle.
4. I/O bus drivers are active (turned on) during third and fourth quarter cycles.

Figure 3. Instruction Cycle and MCLK with: Crystal = 8MHz and Cycle Time = 250 ns

INSTRUCTION SET

General Format and Basic Operations

The 16-bit instruction word (I0 through 15) from program storage is input to the instruction register (Figure 1) and is subsequently decoded to implement the events to occur during the current instruction cycle. The instruction word is formatted as follows:



Rather than discrete instructions, the three operation code (OP CODE) bits specify eight instruction classes. Each instruction class is subject to a number of powerful variations; these variations are specified by the thirteen operand bits. General areas of control for the eight instruction classes are:

- Arithmetic and Logic Operations (ADD, AND, AND XOR)
- Movement of Data and Constants (MOVE and XMIT)
- Branch or Test (JMP, NZT, and XEC)

Basic operations for each of the eight instruction classes are as follows; a summary of the instruction set is provided in Table 1.

MOVE—data in source register or I/O-bus input is moved to destination register or I/O-bus output. Data can be shifted any number of places and/or masked to any length.

ADD—data in source register or I/O-but input is added to content of AUX (R0) register and the result is placed in the destination register or I/O-bus output. Data can be shifted and/or masked, as required.

Table 1. SUMMARY OF 8X300 INSTRUCTION SET

INSTRUC CLASS	OPCODE	FORMATS	DESCRIPTION	I/O CONT SIG	STATE OF CONTROL SIGNAL DURING INSTRUCTION CYCLE: INPUT PHASE (INSTRUCTION INPUT & DATA PROCESSING)	STATE OF CONTROL SIGNAL DURING INSTRUCTION CYCLE: OUTPUT PHASE (ADDRESS & I/O BUS)
MOVE	0	**F1: Register to Register** 0 1 2 | 3 4 5 6 7 | 8 9 10 | 11 12 13 14 15 OPCODE | S | R | D Invalid values of "S": 07_8, 17_8, 20_8-37_8 Invalid values of "D": 10_8, 20_8-37_8	(S) → D Move content of internal register specified by S-field to internal register specified by D-field. Prior to the "MOVE" operation, right-rotate contents of internal source register by octal value (0 through 7) defined by the R-field.	SC = WC = $\overline{LB}$ = $\overline{LB}$ =	L L X X	H if "D" = 07_8, 17_8 L H if "D" = 17_8 L if "D" = 07_8
		F2: I/O Bus to Register 0 1 2 | 3 4 5 6 7 | 8 9 10 | 11 12 13 14 15 OPCODE | S | L | D Valid values of "S": 20_8-37_8 Invalid values of "D": 10_8, 20_8-37_8	Move right-rotated I/O bus (source) data specified by the S-field to internal register specified by the D-field. The L-field specifies the length of source data starting from the LSB-position and, if less than 8-bits, the remaining bits are filled with zeroes.	SC = WC = $\overline{LB}$ = $\overline{LB}$ =	L L L if "S" = 20_8-27_8 H if "S" = 30_8-37_8	H if "D" = 07_8, 17_8 L H if "D" = 17_8 L if "D" = 07_8
		F2: Register to I/O Bus 0 1 2 3 | 4 5 6 7 | 8 9 10 | 11 12 13 14 15 OPCODE | S | L | D Invalid values of "S": 07_8, 17_8, 20_8, 37_8 Valid values of "D": 20_8-37_8	Move contents of internal register specified by the S-field to the I/O latches. Before outputting on I/O bus, data is shifted as specified by the least significant octal digit of the D-field and the bits specified by the L-field are merged with the latched I/O data.	SC = WC = $\overline{LB}$ = $\overline{LB}$ =	L L L if "D" = 20_8-27_8 H if "D" = 30_8-37_8	L H L if "D" = 20_8-27_8 H if "D" = 30_8-37_8
		F2: I/O Bus to I/O Bus 0 1 2 | 3 4 5 6 7 | 8 9 10 | 11 12 13 14 15 OPCODE | S | L | D Valid values of "S": 20_8-37_8 Valid values of "D": 20_8-37_8	Move right rotated I/O-bus (source) data specified by the S-field to the I/O latches. Before outputting on I/O bus, shift data as specified by the D-field; then merge source and latched I/O data as specified by the L (length) field.	SC = WC = $\overline{LB}$ = $\overline{LB}$ =	L L L if "D" = 20_8-27_8 H if "D" =30_8-37_8	L H L if "D" = 20_8-27_8 H if "D" = 30_8-37_8

Table 1. SUMMARY OF 8X300 INSTRUCTION SET (Continued)

INSTRUC CLASS	OPCODE	FORMATS	DESCRIPTION	I/O CONT SIG	STATE OF CONTROL SIGNAL DURING INSTRUCTION CYCLE: INPUT PHASE (INSTRUCTION INPUT & DATA PROCESSING)	STATE OF CONTROL SIGNAL DURING INSTRUCTION CYCLE: OUTPUT PHASE (ADDRESS & I/O BUS)
ADD	1	Same as MOVE instruction class	(S) plus (AUX) → D Same as MOVE instruction class except that contents of AUX (R0) register are ADDed to the source data. If there is a "carry" from MSB, then OVF (overflow) = 1, otherwise OVF = 0.	Same as MOVE instruction class		
AND	2	Same as MOVE instruction class	(S) ∧ (AUX) → D Same as MOVE instruction class except that contents of AUX (R0) register are ANDed with source data.	Same as MOVE instruction class.		
XOR	3	Same as MOVE instruction class	(S) ⊕ (AUX) → D Same as MOVE instruction class except that contents of AUX (R0) register are exclusively ORed with source data.	Same as MOVE instruction class.		
XEC	4	**F3: Register Immediate** 0 1 2 | 3 4 5 6 7 | 8 9 10 11 12 13 14 15 OPCODE | S | J Invalid values of "S": 07_8, 17_8, 20_8-37_8 Valid values of "J": 000_8-377_8	**Execute instruction at current page address offset by J (literal) + (S). Return to normal instruction flow unless a branch is encountered.** Execute instruction at an address determined by replacing the low-order 8-bits of the Program Counter with the following derived sum: • Value of literal (J-field) plus • Contents of internal register specified by S-field The PC is not incremented and the overflow status (OVF) is not changed.	SC = WC = $\overline{LB}$ =	L L X	L L X
		F4: I/O Bus Immediate 0 1 2 | 3 4 5 6 7 | 8 9 10 | 11 12 13 14 15 OPCODE | S | L | J Valid values of "S": 20_8-37_8 Valid values of "J": 00_8-37_8	Execute instruction at an address determined by replacing the low-order 5-bits of Program Counter with the following derived sum: • 5-bit value of literal (J-field) plus • Value of rotated source data specified by S-field (The L-field specifies the length of source data starting from the LSB-position and, if less than 8-bits, the remaining bits are filled with zeros; the Program Counter is not incremented and the overflow status (OVF) is not changed.)	SC = WC = $\overline{LB}$ = $\overline{LB}$ =	L L L if "S" = 20_8-27_8 H if "S" = 30_8-37_8	L L X X
NZT	5	**F3: Register Immediate** 0 1 2 | 3 4 5 6 7 | 8 9 10 11 12 13 14 15 OPCODE | S | J Invalid values of "S": 07_8, 17_8, 20_8-37_8 Valid values of "J": 000_8-377_8	**If data specified by the S-field is not equal to zero, jump to current page address offset by value of J-field; otherwise, increment the Program Counter.** If contents of internal register specified by S-field is **non-zero**, transfer to address determined by replacing the low-order 8-bits of Program Counter with "J", otherwise, increment PC.	SC = WC = $\overline{LB}$ =	L L X	L L X
		F4: I/O Bus Immediate 0 1 2 | 3 4 5 6 7 | 8 9 10 | 11 12 13 14 15 OPCODE | S | L | J Valid values of "S": 20_8-37_8 Valid values of "J": 00_8-37_8	If right-rotated I/O bus data is **non-zero**, transfer to address determined by replacing low-order 5-bits of Program Counter with "J", otherwise, increment PC. (The L-field specifies the length of source I/O data starting from the LSB-position and, if less than 8-bits, the remaining bits are filled with zeroes.)	SC = WC = $\overline{LB}$ = $\overline{LB}$ =	L L L if "S" = 20_8-27_8 H if "S" = 30_8-37_8	L L X X
XMIT	6	**F3: Register Immediate** 0 1 2 | 3 4 5 6 7 | 8 9 10 11 12 13 14 15 OPCODE | D | J Invalid values of "D": 10_8, 20_8-37_8 Valid values of "J": 000_8-377_8	Transmit J → D Transmit and store 8-bit binary pattern in J-field to *internal register specified by D-field.*	SC = WC = $\overline{LB}$ = $\overline{LB}$ =	L L X X	H if D = 07_8 or 17_8 L H if D = 17_8 L if D = 07_8
		F4: I/O Bus Immediate 0 1 2 | 3 4 5 6 7 | 8 9 10 | 11 12 13 14 15 OPCODE | D | L | J Valid values of "D": 20_8-37_8 Valid values of "J": 00_8-37_8	Transmit binary pattern in J-field to I/O bus. Before putting data on I/O bus, shift literal value "J" as specified by the D-field and merge bits specified by the L-field with existing I/O bus data. If the L-field specifies more than 5-bits starting from the LSB-position, all remaining bits are set to zero.	SC = WC = $\overline{LB}$ = $\overline{LB}$ =	L L L if D = 20_8-27_8 H if D = 30_8-37_8	L H L if D = 20_8-27_8 H is D = 30_8-37_8
JMP	7	**F5: Address Immediate** 0 1 2 | 3 4 5 6 7 8 9 10 11 12 13 14 15 OPCODE | A Valid values of A: 00000_8-17777_8	Jump to address in program storage specified by A-field; this address is loaded into the Address Register and the Program Counter.	SC = WC = $\overline{LB}$ =	L L X	L L X

NOTES • $\overline{RB}$ is complement of $\overline{LB}$, X = Undefined

AND—data in source register or I/O-bus input is ANDed with content of AUX (R0) register and the result is placed in the destination register or I/O-bus output. Data can be shifted and/or masked, as required.

XOR—data in source register or I/O-bus input is exclusively ORed with contents of AUX (R0) register and the result is placed in the destination register or I/O-bus output. Data can be shifted and/or masked, as required.

XMIT—immediate data field of instruction word replaces data in destination register or I/O-bus output.

XEC—executes instruction at the program address which is formed by replacing the least significant bits of the last address with the sum of:

- Literal (J) field value of instruction plus,
- Value of data in source register or I/O-bus input.

NZT—least significant bits of program address are replaced by literal (J) field of instruction if the source register or I/O-bus is not equal to zero.

JMP—program address is replaced by address field of the instruction word.

Instruction Fields

As shown in Table 1, each instruction contains an operations code (OPCODE) field and from one-to-three operand fields. The operand fields are: Source (S), Destination (D), Rotate/Length (R/L), Literal (J), and Address (A). The OPCODE and operand fields are briefly described in the following paragraphs.

Operations Code Field: The three-bit OPCODE field specifies one of eight classes of 8X300 instructions; octal designations for this field and operands for each instruction class are shown in Table 1.

Source (S) and Destination (D) Fields: The five-bit (S) and (D) fields specify the source and destination of data for the operation defined by the OPCODE field. The AUXiliary (R0) register is an implied second operand for the ADD, AND, and XOR instructions, each of which require two source fields. That is, instructions of the form:

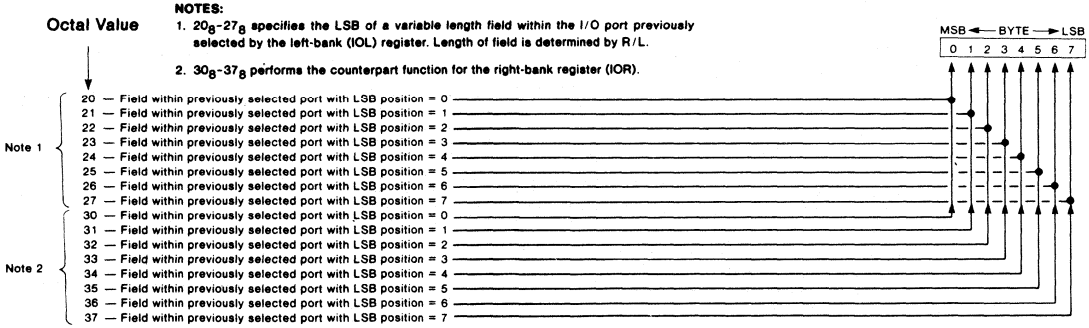
ADD X, Y

imply a third operand, say Z, located in the AUX (R0) register. Thus, the operation for the preceding expression is actually (X + Z), with the result stored in Y. The (S) and/or (D) fields can specify an internal 8X300 register or any one-to-eight bit I/O field; octal values for these registers and Source/Destination field assignments are provided in Table 2.

Table 2. OCTAL ADDRESSES OF 8X300 REGISTERS AND ADDRESS/BIT ASSIGNMENTS OF SOURCE/DESTINATION FIELDS

Octal Value	8X300 Register	Octal Value	8X300 Register
00	Auxiliary (R0)	10	OVF (Overflow Register)—used only as a source
01	R1	11	R11
02	R2	12	Unassigned
03	R3	13	Unassigned
04	R4	14	Unassigned
05	R5	15	Unassigned
06	R6	16	Unassigned
07	*IOL Register—Left Bank I/O Address Register; Used only as destination	17	*IOR Register—Right Bank I/O Address Register; Used only as destination

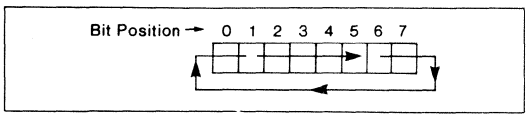
NOTE
*If IOL or IOR is specified as a source of data, the source data is all zeroes.



Rotate (R) and Length (L) Field: The three-bit R/L field performs one of two functions, specifying either the field length (L) or a right-rotate (R). For a given instruction, the specified function depends upon the contents of the source (S) and destination (D) fields.

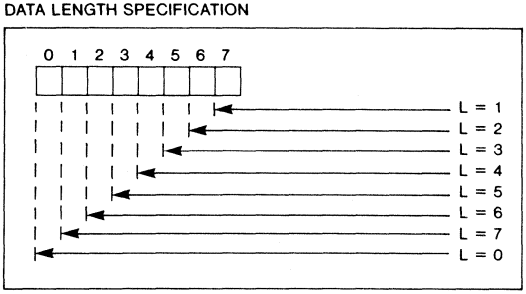
- When an internal register is specified by both the source and destination fields, the (R) field is invoked and it specifies a right-rotate of the data specified in the (S) field—see accompanying diagram. The source-register data (up to eight-bits) is right-rotated within one instruction cycle. (The right-rotate function is implemented on the bus and not in the source register.)

RIGHT-ROTATE FUNCTION



- When either or both of the source and destination fields specify a variable-length I/O data field, the (L) field specifies the length of the I/O data field—see accompanying diagram. If the source field specifies an I/O address (20_8-37_8) and the destination field specifies an internal register (00_8-06_8, 07_8, 11_8, or 17_8), the L-field specifies the length of source data; the source data is formed by right-rotating the I/O bus data according to the source address (Table 2) and then masking result as specified by L-field. If length is less than eight-bits, all remaining bits are set to zero prior to processing data in the ALU. If the source field specifies an internal register (00_8,-06_8, 10_8,or 11_8) and the destination field specifies I/O bus data (20_8-37_8), the L field specifies the length of the destination data. To form the destination data, the ALU output is left-shifted according to the destination address (Table 2) and then masked to the required length—see DATA LENGTH SPECIFICATION. The destination data is merged with data in the I/O latches to finalize the I/O bus data. Hence, a one-to-eight bit destination data field can be inserted into the existing eight-bit I/O port without modifying surrounding bits. If both the source and destination fields specify I/O bus data (20_8-37_8), the L-field specifies the length of both the source and destination data.

DATA LENGTH SPECIFICATION



To form the source data, the I/O bus data is right-rotated according to the source address (Table 2) and then masked to the required length—see preceding DATA LENGTH SPECIFICATION. If length is less than eight-bits, all remaining bits are set to zero before processing in the ALU. To form the destination data, the ALU output is left-shifted according to the destination address (Table 2) and masked to the required length specification. The destination data is then merged into the I/O bus data that was used to obtain the source; thus, if the source and destination addresses are on the same bank, the I/O bus data written to the destination register appears unmodified, except for bits changed during the shift-and-mask operations. If the source and destination addresses refer to different banks, the destination register is changed to contain the contents of the source register in those bit positions not affected by the destination data.

J-Field: The 5-bit or 8-bit (J) field is used to load a literal value (contained in the instruction) into a register, into a variable I/O data field, or to modify the low-order bits of the Program Counter. The bit-length of the (J) field is implied by the (S) field in the XEC, NZT, and XMIT instructions, based on the following considerations.

- When the source (S) field specifies an internal register, the literal value of the J-field is an 8-bit binary number.
- When the source (S) field specifies a variable I/O data field, the literal value of the J-field is a 5-bit binary number.

A-Field: The 13-bit (A) field is an address field which allows the 8X300 to directly address up 8192 locations in Program Storage memory.

INSTRUCTION SEQUENCE CONTROL

Formation of Instruction Address

The Address Register and Program Counter are used to generate addresses for accessing an instruction from program storage. The instruction address is formed in any one of four ways:

- For all except the JMP, XEC, and a "satisfied" NZT instruction, the Program Counter is incremented by one and placed in the Address Register.
- For the JMP instruction, the 13-bit A-field contained in the JMP instruction word replaces the contents of both the Address Register and Program Counter.
- For the XEC instruction, the Address Register is loaded with the high-order bits of the Program Counter modified as follows:
  XEC using I/O Bus Data: low order 5-bits of ALU output replaces counterpart bits in Address Register.
  XEC using Data from Internal Register: low order 8-bits of ALU output replaces counterpart bits in Address Register.
  The Program Counter is not modified for either of the above conditions.
- For a "satisfied" NZT instruction, the low order 5-bits (NZT source is I/O Bus Data) or low order 8/bits (NZT source is an Internal Register) of both the Address Register and Program Counter are loaded with the literal value specified by J-field of the instruction word.

Data Addressing

The source and/or destination addresses of the data to be operated upon are specified as part of the instruction word. As shown in Table 3, source/destination addresses are specified using a five-bit address (00_8 through 37_8). When the most significant octal digit is a 0 or 1, the source and/or destination address is an internal register; if the most significant digit is a 2 or 3, an I/O bus address is indicated—2 specifying a left-bank (LB) address and 3 specifying a right-bank (RB) address. The least significant octal digit (0 through 7) indicates either a specific internal register address or positioning information for the least significant bit when specifying I/O bus data. Referring to Table 1, the AUXiliary register (00) is the implied source of the second argument for the ADD, AND, and XOR operations. IOL (destination address 07_8) and IVR (destination address 17_8) provide a means of routing address information to I/O registers. With IOL or IOR specified as the destination address, the data is placed on the I/O bus during the output phase of the instruction cycle. Simultaneously, a select command (SC) is generated to inform all I/O devices that information on the I/O bus is to be considered as an I/O address. Since IOL and IOR are not harware registers, they should never be specified as a source address.

Control outputs $\overline{LB}$ and $\overline{RB}$ are used to partition I/O bus devices into two fields of 256 addresses. With LB in the active-low state and a source address of 20_8-27_8, the left bank of I/O devices are enabled during the input phase of the instruction cycle. With $\overline{RB}$ in the active-low state and a source address of 30_8-37_8, the right bank of devices are enabled. During the output phase, $\overline{RB}$ is low if the destination address is IOR (17_8) or 30_8-37_8; $\overline{LB}$ is low if the destination address is IOL (07_8) or 20_8-27_8. Each address field ($\overline{LB}$ and $\overline{RB}$) can have a different I/O device selected; thus, two devices can be directly accessed within one instruction cycle.

Table 3. SOURCE/DESTINATION ADDRESSES

SOURCE AND/OR DESTINATION FIELD (OCTAL)	SOURCE/DESTINATION
00	AUXiliary register (R0)
01-06	Working registers R1-R6, respectively
07	IOL Left-bank enable (Destination only)
10	Overflow status—OVF (Source only)
11	Working register R11
17	IOR Right-bank enable (Destination only)
2N (N = 0, 1, 2, 3 4, 5, 6, or 7)	If a source, I/O data is right-rotated (7 − N) bits and then masked as specified by the L-field. $\overline{LB}$ = low and $\overline{RB}$ = high generated during input phase. If a destination, I/O data is left-shift (7 − N) bits and merged (specified by L-field) with data contained in the I/O latches. $\overline{LB}$ = low and $\overline{RB}$ = high generated during output phase.
3N (N = 0, 1, 2, 3, 4, 5, 6, or 7)	If a source, I/O data is right-rotated (7 − N) bits and then masked as specified by the L-field. $\overline{LB}$ = high and $\overline{RB}$ = low generated during input phase. If a destination, I/O data is left-shifted (7 − N) bits and merged (specified by L-field) with data contained in the I/O latches. $\overline{LB}$ = high and $\overline{RB}$ = low generated during output phase.

DESIGN PARAMETERS

Hardware design of an 8X300-based system largely consists of the following operations:

- **Selecting and interfacing a Program Storage device—ROM, PROM, etc. (Pins 2 through 9 and 45 through 49 for 13-bit address interface; Pins 13 through 28 for 16-bit instruction interface.)**
- **Selecting and interfacing Input/Output devices—RAM, Multiplexers, I/O Ports, and other eight-bit addressable I/O devices. (Pins 33 through 36 and pins 38 through 41 for eight-bit I/O interface.)**
- **Choosing and implementing System Clock—Capacitor-Controlled, Crystal-Controlled, or Externally-Driven. (Pins 10 and 11 for System Clock interface.)**
- **Selection of 5-volt power supply and off-chip series-pass transistor.**
- **External logic, as required, to meet the control requirements of a particular application.**

All information required for easy implementation of these design requirements is provided under the following captions.

- **DC Characteristics**
- **AC Characteristics**
- **Timing Considerations**
- **Clock Considerations**
- **HALT/RESET Logic**
- **Voltage Regulator**

DC CHARACTERISTICS (Commercial Part) $4.75V \leq V_{CC} \leq 5.25V$, $0°C \leq T_A \leq 70°C$

PARAMETER		TEST CONDITIONS	LIMITS			UNIT	COMMENTS
			Min	Typ	Max		
V_{CC}	Supply voltage		475	5.0	5.25	V	5V ± 5%; pin 37 only
V_{IH}	High level input voltage		0.6 2.0		2.0	V	X1 and X2 All other pins
V_{IL}	Low level input voltage				0.4 0.8	V	X1 and X2 All other pins
V_{OH}	High level output voltage	V_{CC} = min; I_{OH} = −3mA	2.4	3.0		V	
V_{OL}	Low level output voltage	V_{CC} = min; I_{OL} = 6mA V_{CC} = min; I_{OL} = 16mA		0.39 0.39	0.55 0.55	V	A0 through A12 All other outputs
V_{CR}	Regulator voltage	V_{CC} = 5V		3.1		V	From series-pass transistor
V_{IC}	Input clamp voltage	V_{CC} = min; I_{IN} = −10mA			−1.5	V	Crystal inputs X1 and X2 do not have internal clamp diodes.
I_{IH}	High-level input current	V_{CC} = max; V_{IH} = 0.6V V_{IH} = 4.5V		1	3.0 50	mA μA	X1 and X2 All other pins
I_{IL}	Low-level input current	V_{CC} = max; V_{IL} = 0.4V		−0.13 −0.67 −0.23	−3 −0.2 −1.6 −0.4	mA	X1 and X2 $\overline{IV0}$-$\overline{IV7}$ I0-I15 $\overline{HALT}$ and $\overline{RESET}$
I_{OS}	Short circuit output current	V_{CC} = max; V_{CR} = V_{CRH} (Note: At any time, no more than one output should be connected to ground.)	−30		−140	mA	All output pins
I_{CC}	Supply current	V_{CC} = max; V_{CR} = V_{CRH}			160	mA	
I_{REG}	Regulator control	V_{CC} = 5.0V	−14		−21	mA	
I_{CR}	Regulator current	V_{CC} = max			230	mA	70°C
					265	mA	25°C
					290	mA	0°C

NOTES:
1. Operating temperature ranges are guaranteed after thermal equilibrium has been reached.
2. All voltages measured with respect to ground terminal.

AC CHARACTERISTICS (Commercial Part)

CONDITIONS: V_{CC} = 5V (±5%), V_{IN} = 0V or 3V, $0°C \leq T_A \leq 70°C$
LOADING: (See test circuits)

PARAMETER (NOTE 1)		LIMITS (INSTRUCTION CYCLE TIME = 250 ns)			LIMITS (INSTRUCTION CYCLE TIME > 250 ns)			UNITS	COMMENTS
		Min	Typ	Max	Min	Typ	Max		
T_{PC}	Processor cycle time	250			250			ns	
T_{CP}	X1 clock period	125			125			ns	
T_{CH}	X1 clock high time	62			62			ns	
T_{CL}	X1 clock low time	62			62			ns	
T_{MCH}	MCLK high delay	31	42	52	31	42	52	ns	
T_{MCL}	MCLK low delay	31	42	52	31	42	52	ns	
T_W	MCLK pulse width	55	62	69	T_{4Q}−7	T_{4Q}		ns	Note 2
T_{AS}	X1 falling edge to address stable	50	63	80	50	63	80	ns	Note 7

AC CHARACTERISTICS (Commercial Part) (Continued) **CONDITIONS:** V_{CC} = 5V (±5%), V_{IN} = 0V or 3V, 0°C ≤ T_A ≤ 70°C
LOADING: (See test circuits)

PARAMETER (NOTE 1)		LIMITS (INSTRUCTION CYCLE TIME = 250 ns)			LIMITS (INSTRUCTION CYCLE TIME > 250 ns)			UNITS	COMMENTS
		Min	Typ	Max	Min	Typ	Max		
T_{MAS}	MCLK falling edge to address stable	130	143	160	$T_{1Q}+T_{2Q}$ +5	$T_{1Q}+T_{2Q}$ +18	$T_{1Q}+T_{2Q}$ +35	ns	Notes 2, 3, & 7
T_{IA}	Instruction to address			170			T_{2Q}+108	ns	Notes 2, 3, & 8
T_{IVA}	Input data to address			105			105	ns	Notes 3 & 9
T_{IS}	Instruction set-up time (X1 rising edge)	−7			−7			ns	Note 10
T_{MIS}	MCLK falling edge to instruction stable			20			T_{1Q}−42	ns	Notes 2, 4, & 10
T_{IH}	Instruction hold time (X1 rising edge)	45			45			ns	Note 11
T_{MIH}	Instruction hold time) (MCLK falling edge)	60			T_{1Q}−2			ns	Notes 2 & 11
T_{WH}	X1 falling edge to SC/WC rising edge	40	49	58	40	49	58	ns	
T_{MWH}	MCLK falling edge to SC/WC rising edge	125	130	135	$T_{1Q}+T_{2Q}$	$T_{1Q}+T_{2Q}$ +5	$T_{1Q}+T_{2Q}$ +10	ns ns	Note 2
T_{WL}	X1 falling edge to SC/WC falling edge	40	49	58	40	49	58	ns	
T_{MWL}	MCLK falling edge to SC/WC falling edge	5	7	15	5	7	15	ns	
T_{IBS}	X1 falling edge to $\overline{LB}/\overline{RB}$ (Input phase)	48	60	70	48	60	70	ns	
T_{MIBS}	MCLK falling edge to $\overline{LB}/\overline{RB}$ (Input phase)	7	17	25	7	17	25	ns	
T_{IIBS}	Instruction to $\overline{LB}/\overline{RB}$ (Input phase)		27	35		27	35	ns	
T_{OBS}	X1 falling edge to $\overline{LB}/\overline{RB}$ (Output phase)	48	60	70	48	60	70	ns	
T_{MOBS}	MCLK falling edge to $\overline{LB}/\overline{RB}$ (Output phase)	132	137	147	$T_{1Q}+T_{2Q}$ +7	$T_{1Q}+T_{2Q}$ +12	$T_{1Q}+T_{2Q}$ +22	ns	Note 2
T_{IDS}	Input data set-up time (X1 falling edge)	25	16		25	16		ns	
T_{MIDS}	MCLK falling edge to input data stable		65	55		$T_{1Q}+T_{2Q}$ −60	$T_{1Q}+T_{2Q}$ −70	ns	Notes 2 & 5
T_{IDH}	Input data hold time (X1 falling edge)	40	30		40	30		ns	
T_{MDIH}	Input data hold time (MCLK falling edge)	125	112		$T_{1Q}+T_{2Q}$	$T_{1Q}+T_{2Q}$ −13		ns	Note 2
T_{ODH}	Output data hold time (X1 falling edge)	55	65	75	55	65	75	ns	
T_{MODH}	Output data hold time (MCLK falling edge)	11	20	25	11	20	25	ns	
T_{ODS}	Output data stable (X1 falling edge)	74	84	94	74	84	94	ns	Notes 12, 14, & 15
T_{MODS}	Output data stable (MCLK falling edge)	150	160	170	$T_{1Q}+T_{2Q}$ +25	$T_{1Q}+T_{2Q}$ +35	$T_{1Q}+T_{2Q}$ +45	ns	Notes 2, 12, 14, & 15

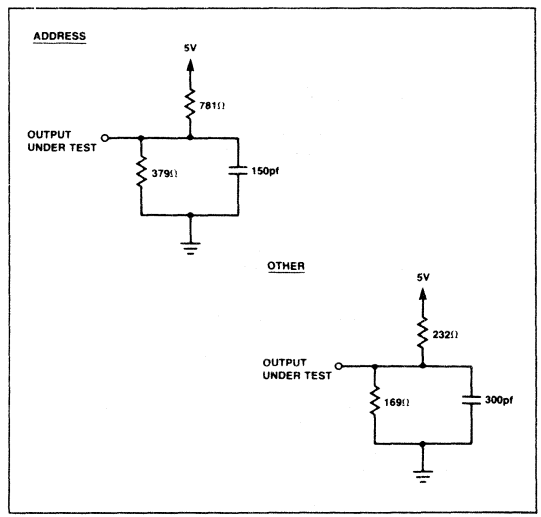
AC CHARACTERISTICS (Commercial Part) (Continued)

CONDITIONS: V_{CC} = 5V (±5%), V_{IN} = 0V or 3V, 0°C ≤ T_A ≤ 70°C
LOADING: (See test circuits)

PARAMETER (NOTE 1)		LIMITS (INSTRUCTION CYCLE TIME = 250 ns)			LIMITS (INSTRUCTION CYCLE TIME > 250 ns)			UNITS	COMMENTS
		Min	Typ	Max	Min	Typ	Max		
T_{DD}	Input data to output data	104	120	136	104	120	136	ns	Notes 13 & 15
T_{HS}	$\overline{HALT}$ set-up time (X1 rising edge)	0			0			ns	
T_{MHS}	MCLK falling edge to $\overline{HALT}$ falling edge			18			$T_{1Q}-44$	ns	Notes 2 & 6
T_{HH}	$\overline{HALT}$ hold time (X1 rising edge)	32			32			ns	
T_{MHH}	$\overline{HALT}$ hold time (MCLK falling edge)	50			$T_{1Q}-12$			ns	Note 2
T_{ACC}	Program storage access time			80				ns	
T_{IO}	I/O port output enable time ($\overline{LB}/\overline{RB}$ to valid IV data input)			30				ns	

NOTES:
1. X1 and X2 inputs are driven by an external pulse generator with an amplitude of 1.5 volts; all timing parameters are measured at this voltage level.
2. Respectively, T_{1Q}, T_{2Q}, T_{3Q}, and T_{4Q} represent time intervals for the first, second, third, and fourth quarter cycles.
3. Capacitive loading for the address bus is 150 picofarads.
4. Same as TIS but referenced to falling edge of MCLK.
5. Same as TIDS but referenced to falling edge of MCLK.
6. Same as THS but referenced to falling edge of MCLK.
7. TAS is obtained by forcing a valid instruction and an I/O bus input to occur earlier than the specified minimum set-up time; the TAS parameter then represents the earliest time that the address bus is valid.
8. TIA is obtained by forcing a valid instruction input to occur earlier than the minimum set-up time.
9. TIVA is obtained by forcing a valid I/O bus input to just meet the minimum set-up time.
10. TMIS represents the set-up time required by internal latches of the 8X300. In system applications, the instruction input may have to be valid before the worst-case set-up time in order for the system to respond with a valid I/O bus input that meets the I/O bus input set-up time (TIDS and TMIDS).
11. TIH represents the hold time required by internal latches of the 8X300. To generate proper LB/RB signals, the instruction must be held valid until the address bus changes.
12. TODS is obtained by forcing a valid I/O bus input to occur earlier than the I/O bus input set-up time (TIDS); this timing parameter represents the earliest time that the I/O output data can be valid.
13. TDD is obtained by forcing a valid I/O bus input to just meet the minimum I/O bus input set-up time; thus timing parameter represents the latest time that the I/O output data can be valid.
14. The minimum figure for these parameters represents the earliest time that I/O bus output drivers of the 8X300 will turn on.
15. For TIDS ≥ 35 ns, TODS or TMODS should be used to determine when the output data is stable.

TEST CIRCUITS



DC CHARACTERISTICS (Military Part)

S8X300-1 $-40°C \leq TC \leq 100°C$ $V_{CC} = 5V \pm 5\%$
S8X300-2 $-20°C \leq TC \leq 100°C$ $V_{CC} = 5V \pm 10\%$

PARAMETER		TEST CONDITIONS	LIMITS			UNIT
			Min	Typ	Max	
V_{IH}	High level input voltage X1, X2 All others		 0.6 2.0			 V V
V_{IL}	Low level input voltage X1, X2 All others				 0.4 0.8	 V V
V_{IC}	Input clamp voltage (Notes 1 & 5)	V_{CC} = min $I_I = -10mA$			−1.5	V
I_{IH}	High level input current X1, X2 All others	 V_{CC} = max $V_{IH} = 0.6V$ V_{CC} = max $V_{IH} = 4.5V$			 3.0 0.05	 mA
I_{IL}	Low level input current X1, X2	 V_{CC} = max $V_{IL} = 0.4V$			 −3.0	 mA
	$\overline{IV0}$-$\overline{IV7}$	V_{CC} = max $V_{IL} = 0.4V$			−0.3	mA
	I0-I15	V_{CC} = max $V_{IL} = 0.4V$			−1.6	mA
	$\overline{HALT}$, $\overline{RESET}$	V_{CC} = max $V_{IL} = 0.4V$			−0.4	mA
V_{OL}	Low level output voltage A0-A12 All others	 V_{CC} = min $I_L = 4.25mA$ V_{CC} = min $I_{OL} = 16mA$			 0.55 0.55	 V V
V_{OH}	High level output voltage	V_{CC} = min $I_{OH} = -3mA$	2.4			V
I_{OS}	Short circuit output current (Note 2)	V_{CC} = max	−30		−140	mA
I_{CC}	Supply current (Note 4)	V_{CC} = max			160	mA
I_{REG}	Regulator control	$V_{CC} = 5.0V$	−14		−21	mA
I_{CR}	Regulator current	V_{CC} = max			285	mA
I_{CR}	Regulator current	$TC \geq 25°C$ V_{CC} = max			330	mA
V_{CR}	Regulator voltage	$TC < 25°C$ (Note 3)		3.1		V

NOTES:
1. Crystal inputs X1 and X2 do not have clamp diodes.
2. Only one output may be grounded at a time.
3. From series-passed transistor under the following conditions:
   VCC = Max, HALT = RESET = ADDRESS = IVX = 0.0V, all other pins open.
4. Pin 37 only.
5. Test each input one at a time.
6. All voltages are with respect to ground terminal.
7. The operating temperature ranges are guaranteed after thermal equilibrium has been reached.
8. Storage temperature −65°C to +150°C.

AC CHARACTERISTICS (Military Part)

CONDITIONS: S8X300-1—V_{CC} = 5V (±5%) -40°C ≤ T_C ≤ 100°C
S8X300-2—V_{CC} = 5V (±10%) -20°C ≤ T_C ≤ 100°C

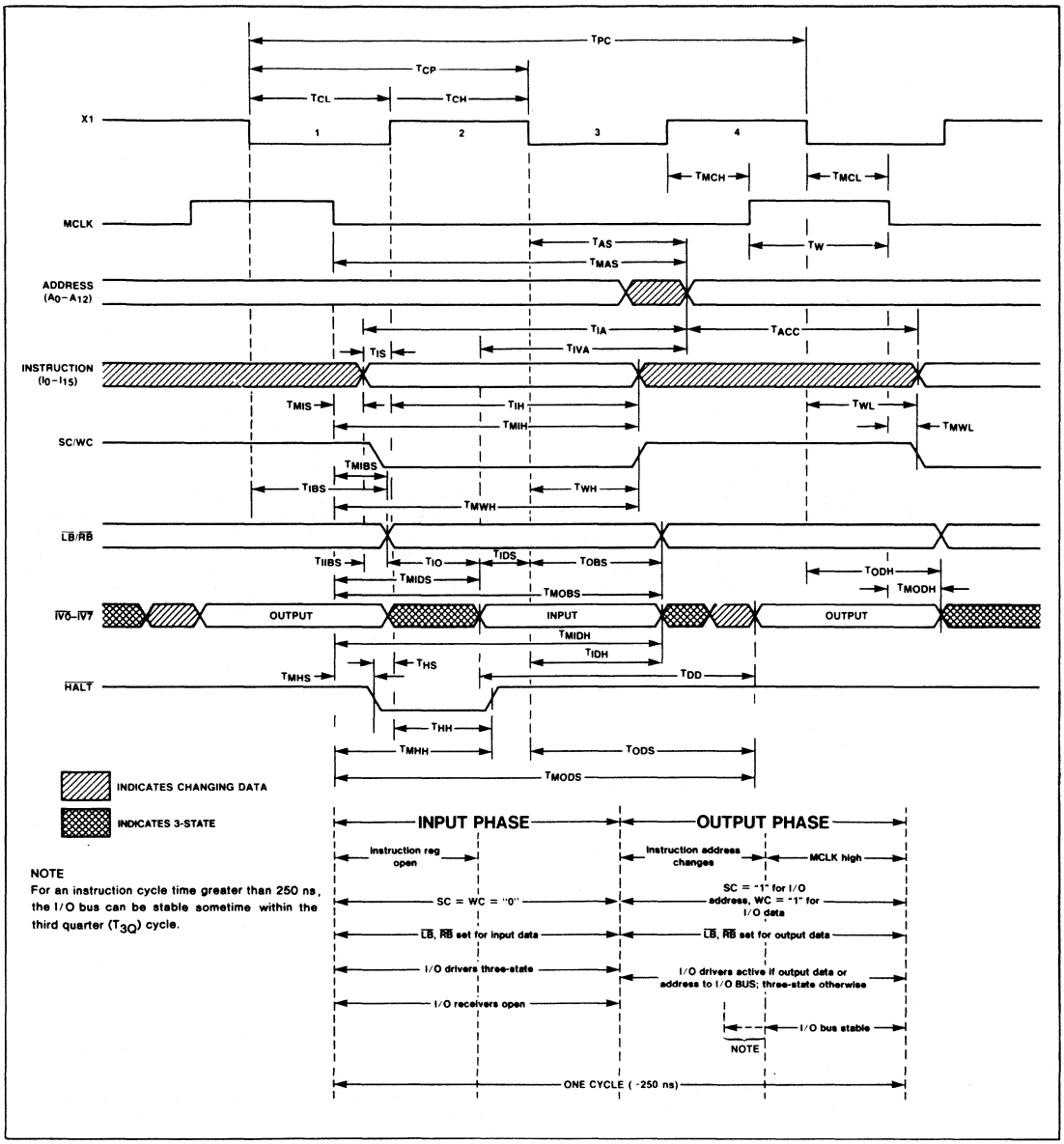
PARAMETER		TEST CONDITIONS (NOTES 1 & 2)	LIMITS			UNIT
			Min	Typ	Max	
Clock:						
T_{PC}	Processor cycle time		300			ns
T_{CP}	X1 clock period		150			ns
T_{CH}	X1 clock high time		62			ns
T_{CL}	X1 clock low time		62			ns
Controls:						
T_{HS}	$\overline{HALT}$ set-up time (X1 rising edge)		0			ns
T_{HH}	$\overline{HALT}$ hold time (X1 rising edge)		50			ns
Instructions:						
T_{AS}	X1 falling edge to address stable	CL = 100pF	35		92	ns
T_{IS}	Instruction set-up time (X1 rising edge)		0			ns
T_{IH}	Instruction hold time (X1 rising edge)		50			ns
T_{MCH}	MCLK high delay	X1 = 2.0V	20		55	ns
T_{MCL}	MCLK low delay	X1 = 2.0V	20		55	ns
T_{WH}	X1 falling edge to SC/$\overline{WC}$ rising edge				80	ns
T_{WL}	X1 falling edge to SC/$\overline{WC}$ falling edge				80	ns
T_{IIBS}	Instruction to $\overline{LB}$/$\overline{RB}$ (input phase)				52	ns
T_{IBS}	X1 falling edge to $\overline{LB}$/$\overline{RB}$ (input phase)		24			ns
T_{OBS}	X1 falling edge to $\overline{LB}$/$\overline{RB}$ (output phase)				90	ns
T_{IDS}	Input data set-up time (X1 falling edge)		36			ns
T_{IDH}	Input data hold time (X1 falling edge)		50			ns
T_{ODS}	Output data stable (X1 falling edge)				125	ns
T_{ODH}	Output data hold time (X1 falling edge)		35		85	ns
T_{ACC}	Instruction access time	Provided by worst case timing	80			ns
T_{IO}	Data I/O access time	Provided by worst case timing	40			ns

NOTES:
1. Operating temperature ranges are guaranteed after thermal equilibrium has been reached.
2. Unless otherwise noted CL = 300 pF, VIN = 3V.

TIMING CONSIDERATIONS (Commercial Part)

As shown in the "AC CHARACTERISTICS" table for this part, the minimum instruction cycle time is 25 ns, whereas, the maximum is determined by the on-chip oscillator frequency and can be any value the user chooses. With an instruction cycle time of 250 ns, the part can be characterized in terms of absolute values; these are shown in the first "LIMITS" column of the table. When the instruction cycle time is greater than 250 ns, certain parameters are cycle-time dependent; thus, these parameters are specified in terms of the four quarter cycles (T_{1Q}, T_{2Q}, T_{3Q}, and T_{4Q}) that make up one instruction cycle—see 8X300 TIMING DIAGRAM. As the time interval for each instruction cycle increases (becomes greater than 250 ns), the delay for all parameters that are cycle-time dependent is likewise increased. In some cases, these delays have a significant impact on timing relationships and other areas of systems design; subsequent paragraphs describe these timing parameters and reliable methods of calculation.

8X300 TIMING DIAGRAM



NOTE
For an instruction cycle time greater than 250 ns, the I/O bus can be stable sometime within the third quarter (T_{3Q}) cycle.

Timing parameters for the 8X300 are normally measured with reference to X1 or MCLK; those referenced to MCLK are prefaced with an "M" in the mnemonic—TMAS, TMIH, and so on. To determine the timing relationship between a particular signal, say "A" and MCLK, the user should, at all times, use the value specified in the table—DO NOT calculate the value by adding or subtracting two or more parameters that are referenced to X1. When deriving timing relationships between two signals (A to B, etc.) by adding or subtracting the parameter values, the user must consistently use the same parameter reference—MCLK or X1.

System determinants for the instruction cycle time are:

- Propagation delays within the 8X300
- Access time of Program Storage
- Enable time of the I/O port

Normally, the instruction cycle time is constrained by one or more of the following conditions:

Condition 1—Instruction or MCLK to $\overline{LB}/\overline{RB}$ (input phase) plus I/O port access time (TIO) $\leq$ IV data set-up time (Figure 4a).

Condition 2—Program storage access time (TACC) plus instruction to $\overline{LB}/\overline{RB}$ (input phase) plus I/O port access time (TIO) plus IV data (input phase) to address $\leq$ instruction time (Figure 4b).

Condition 3—Program storage access time plus instruction to address $\leq$ instruction cycle time (Figure 4c).

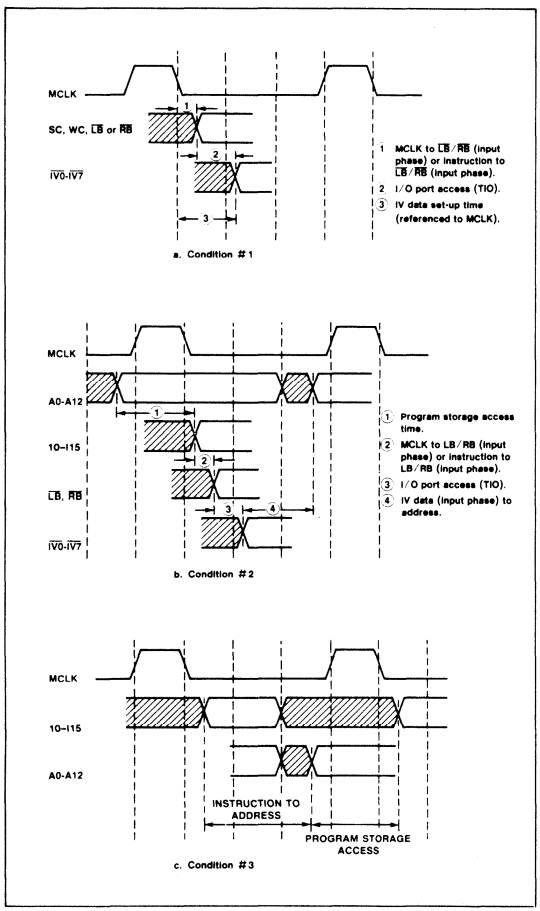


Figure 4. Constraints of 8X300 Instruction Cycle Time

From condition #1 and with an instruction cycle time of 250 ns, the I/O port access time (TIO) can be calculated as follows:

$$TMIBS + TIO \leq TMIDS$$
transposing, $TIO \leq TMIDS - TMIBS$
substituting, $TIO \leq 55ns - 25ns$
result, $TIO \leq 30\ ns$

Using 30 ns for TIO, the constraint imposed by condition #1 can also be used to calculate the minimum cycle time:

$$TMIBS + TIO \leq TMIDS$$
thus, $25ns + 30ns \leq T_{1Q} + T_{2Q} - 70$
$25ns + 30ns \leq \frac{1}{2}$ cycle $- 70$ therefore, the worst-case instruction cycle time is 250 ns. With subject parameters referenced to X1, the same calculations are valid:

$$TIBS + TIO + TIDS \leq \frac{1}{2} \text{ cycle}$$
thus, $70ns + 30ns + 25ns \leq \frac{1}{2}$ cycle therefore, the worst-case instruction cycle time is again 250 ns. From condition #2 and with an instruction cycle time of 250 ns, the program storage access time can be calculated:

$$TACC + TIIBS + TIO + TIVA \leq 250ns$$
transposing, $TACC \leq 250ns - TIIBS - TIO - TIVA$
substituting, $TACC \leq 250ns - 35ns - 30ns - 105ns$
thus, $TACC \leq 80ns$ hence, for an instruction cycle time of 250 ns, a program storage access time of 80 ns is implied. The constraint imposed by condition #3 can be used to verify the maximum program storage access time:

$$TIA + TACC \leq \text{Instruction Cycle}$$
thus, $TACC \leq 250ns - 170ns$
and, $TACC \leq 80ns$, confirming that a program storage access time of 80 ns is satisfactory.

For an instruction cycle time of 250 ns and a program storage access time of 80 ns (Condition #2/Figure 4b), the instruction should be valid 10 ns before the falling edge of MCLK. This relationship can be derived by the following equation:

$$250ns - TMAS - TACC$$
$$= 250ns - 160ns - 80ns$$
$$= 10ns$$

It is important to note that, during the input phase, the beginning of a valid $\overline{LB}/\overline{RB}$ signal is determined by either the instruction to $\overline{LB}/\overline{RB}$ delay (TIIBS) or the delay from the falling edge of MCLK to $\overline{LB}/\overline{RB}$ (TMIBS). Assuming the instruction is valid 10 ns before the falling edge of MCLK and adding the instruction-to-LB/RB delay (TIIBS) = 30ns), the $\overline{LB}/\overline{RB}$ signal will be valid 25 ns after the falling edge of MCLK. With a fast program storage memory and with a valid instruction more than 10 ns before the falling edge of MCLK—the $\overline{LB}/\overline{RB}$ signal will, due to the TMIBS delay, still be valid 25 ns after the falling edge of MCLK. Using a worst-case instruction cycle time of 250 ns, the user cannot gain a speed advantage by selecting a memory with faster access time. Under the same conditions, a speed advantage cannot be obtained by using an I/O port with fast access time (TIO) because the address bus will be stable 80 ns (TAS) after the beginning of the third quarter cycle—no matter how early the IV data input is valid.

Internal Timing and Timing Relationships

All timing and timing-control signals of the 8X300 are generated by the oscillator and sequencer shown in Figure 5. The sequencer outputs direct and control all of the timing parameters specified in the TIMING DIAGRAM. Observe that each input quarter cycle bears a fixed relationship to X1 via the propagation delay.

General and interactive timing relationships pertaining to I/O signals of the 8X300 are shown in Figure 6. Example—in the input phase, the switching point of the $\overline{LB}/\overline{RB}$ signal is caused by the worst-case delay from the instruction to $\overline{LB}/\overline{RB}$ or from the beginning of the first internal quarter cycle to $\overline{LB}/\overline{RB}$; the two arrows pointing to the $\overline{LB}/\overline{RB}$ transition indicate this "either/or" dependency. This information coupled with tabular values and the TIMING DIAGRAM provides the user with the wherewithal to calculate any and all system timing parameters.

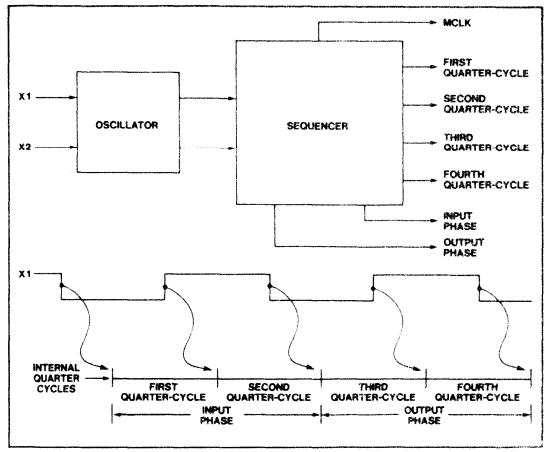


Figure 5. Timing and Timing Control Signals of the 8X300

CLOCK CONSIDERATIONS

The on-chip oscillator and timing-generation circuits of the 8X300 can be controlled by any one of the following methods:

Capacitor: if timing is not critical
Crystal: if precise timing is required
External Drive: if application requires that the 8X300 be synchronized with system clock

Capacitor Timing: A non-polarized ceramic or mica capacitor with a working voltage equal to or greater than 25-volts is recommended. The lead lengths of capacitor should be approximately the same and as short as possible; also, the timing circuits should not be in close proximity to external sources of noise. For various capacitor (C_X) values, the cycle time can be approximated as:

C_X (in pF)	APPROXIMATE CYCLE TIME
100	300 ns
200	500 ns
500	1.1 μs
1000	2.0 μs

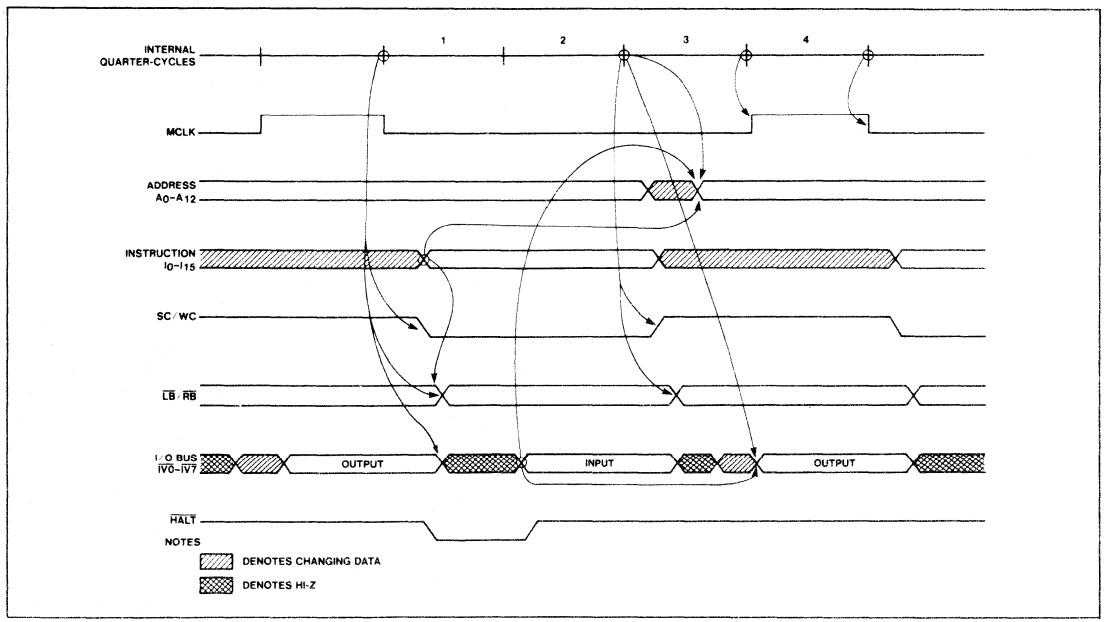


Figure 6. Timing Relationships of 8X300 I/O Signals

Crystal Timing: When a crystal is used, the on-chip oscillator operates at the resonant frequency (f_O) of the crystal; the series-resonant quartz crystal connects to the 8X300 via pins 10 (X1) and 11 (X2). The lead lengths of the crystal should be approximately equal and as short as possible; also, the timing circuits should not be in close proximity to external sources of noise. The crystal should be hermetically sealed (HC type can) and have the following electrical characteristics:
Type: Fundamental mode, series resonant
Impedance at Fundamental: 35-ohms maximum
Impedance at Harmonics and Spurs: 50-ohms minimum

The resonant frequency (f_O) of the crystal is related to the desired cycle time (T) by the equation $f_O = 2/T$; for a cycle time of 250 ns, f_O = 8MHz.

Using an External Clock: The 8X300 can be synchronized with an external clock by simply connecting appropriate drive circuits to the X1/X2 inputs. Figure 7 shows how the on-chip oscillator can be driven from the complementary outputs of a pulse generator. In applications where the microcontroller must be driven from a master clock, the X1/X2 lines can be interfaced to TTL logic as shown in Figure 8.

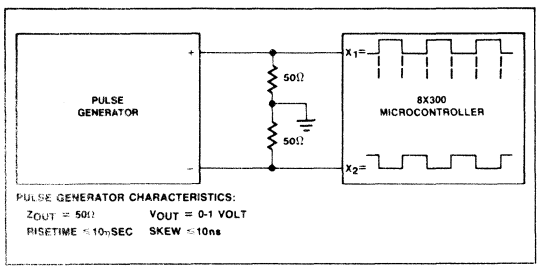


Figure 7. Clocking with a Pulse Generator

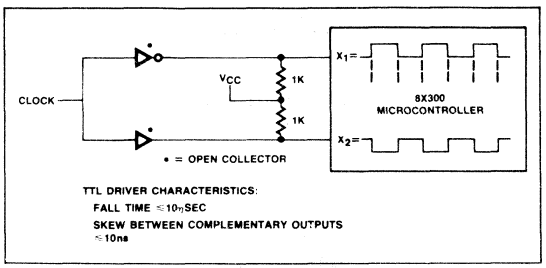


Figure 8. Clocking with TTL

RESET Logic

The $\overline{\text{RESET}}$ line (pin 43) can be driven from a high (inactive) state to a low (active) state at any time with respect to the system clock, that is, the reset function is asynchronous. To ensure proper operation, the $\overline{\text{RESET}}$ line should be held low (active) for one full instruction time. When the line is driven from a high state to an active-low state, several events occur—the precise instant of occurrence is basically a function of the propagation delay for that particular event. As shown in the acompanying $\overline{\text{RESET}}$ timing diagram, these events are:

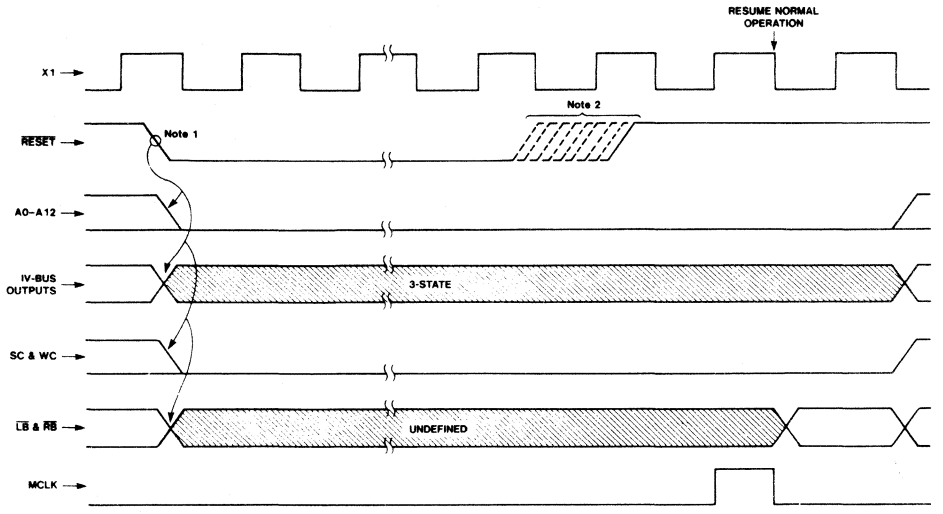
- **The Program Counter and Address Register are set to an all-zero configuration and remain in that state as long as the $\overline{\text{RESET}}$ line is low. Other than PC and AR, reset does not affect other internal registers.**
- **The input/output (IV) bus goes three-state and remains in that mode as long as the $\overline{\text{RESET}}$ line is low.**
- **The Select Command and Write Command signals are driven low and remain inactive as long as the $\overline{\text{RESET}}$ line is low.**
- **The Left Bank/Right Bank signals are undefined for the period in which the $\overline{\text{RESET}}$ line is low.**

During the time $\overline{\text{RESET}}$ is active-low, MCLK is inhibited; moreover, if the $\overline{\text{RESET}}$ line is driven low during the last two quarter cycles, MCLK can be shortened for that particular machine cycle. When $\overline{\text{RESET}}$ line is driven high (inactive)—one-quarter to one full instruction cycle later—MCLK appears just before normal operation is resumed. The $\overline{\text{RESET}}$/MCLK relationship is clearly shown by "B" in the timing diagram. As long as the $\overline{\text{RESET}}$ line is active-low, the $\overline{\text{HALT}}$ signal (described next) is not sampled by internal logic of the 8X300.

HALT Logic

The $\overline{\text{HALT}}$ signal is sampled via internal chip logic at the end of the first internal quarter of each instruction cycle. If, when sampled, the HALT signal is active-low, a halt is immediately executed and the current instruction cycle is terminated; however, the halt cycle does not inhibit MCLK nor does it affect any internal registers of the 8X300. As long as the $\overline{\text{HALT}}$ line is active-low, the SC and WC lines are low (inactive) and the input/output (IV) bus remains in the three-state mode of operation. The halt cycle continues until, when again sampled, the $\overline{\text{HALT}}$ line is found to be high; at this time, normal operation is resumed. Timing for the halt signal is shown in the accompanying diagram.

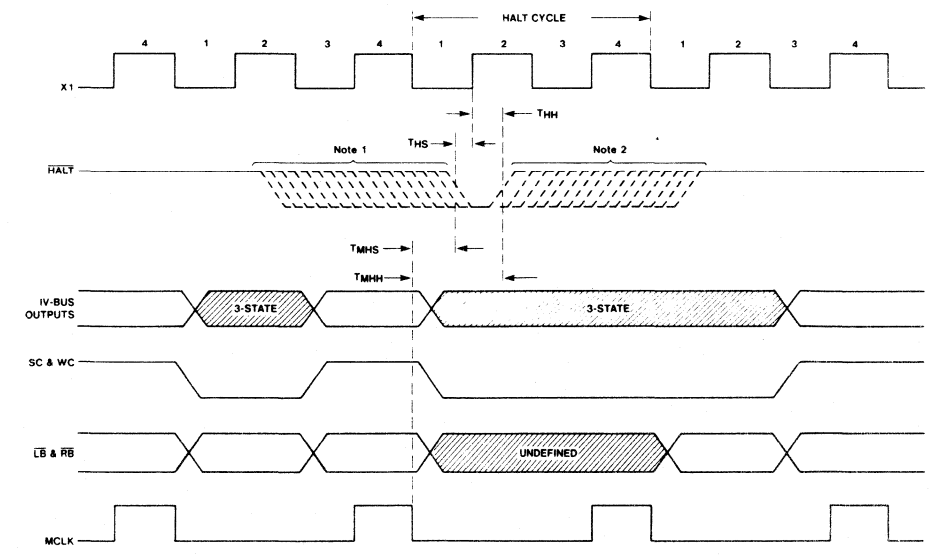
$\overline{\text{RESET}}$ TIMING DIAGRAM



NOTES

1. A High to Low transition of the $\overline{\text{RESET}}$ signal will force the Address Bus to an all-zero configuration.
2. The $\overline{\text{RESET}}$ signal can switch from Low to High at any point within this time interval and, in all cases, MCLK will occur at least one-quarter cycle time later as shown.

$\overline{\text{HALT}}$ TIMING DIAGRAM



NOTES

1. The $\overline{\text{HALT}}$ signal can switch from High to Low at any time during this interval.
2. The $\overline{\text{HALT}}$ signal can switch from Low to High at any time during this interval.

Timing Descriptions:

T_{HS}—set-up time from $\overline{\text{HALT}}$ to X1 (independent of instruction cycle time)

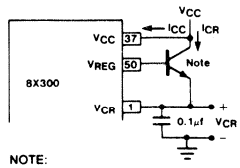
T_{HH}—hold time from X1 to $\overline{\text{HALT}}$ (independent of instruction cycle time)

T_{MHS}—set-up time from MCLK to $\overline{\text{HALT}}$ (dependent upon instruction cycle time)

T_{MHH}—hold time from MCLK to $\overline{\text{HALT}}$ (dependent upon instruction cycle time)

VOLTAGE REGULATOR

All internal logic of the 8X300 is powered by an on-chip voltage regulator that requires an external series-pass transistor. Electrical specifications for the off-chip power transistor and a typical hook-up are shown in the accompanying diagram. To minimize lead inductance, the transistor should be as close as possible to the 8X300 package and the emitter should be ac-grounded via a 0.1-microfarad ceramic capacitor.



PARAMETER	CONDITIONS	LIMITS
h_{fe}	V_{CE} = 2V; 100mA < I_C < 500mA	> 50
BE_{ON}	V_{CE} = 5V; I_C = 500mA	< 1V
V_{CESAT}	I_C = 500mA; I_B = 50mA	< 0.5V
BV_{CEO}		> 15V
f_t		> 30MHz

NOTE:
Typical approved parts—2N5320, 2N5337

MICROCONTROLLER

Originally published by Signetics January 1984

FEATURES

- **Fetch, Decode, and Execute a 16-bit Instruction in a minimum of 200 nanoseconds (one machine cycle)**
- **Bit-oriented instruction set (addressable single-or-multiple bit subfields)**
- **Separate buses for Instruction, Instruction Address and Three-State I/O**
- **Thirteen 8-bit general-purpose working registers**
- **Source/destination architecture**
- **Bipolar low-power Schottky technology/TTL inputs and outputs**
- **On-chip oscillator and timing generation**
- **Single +5V supply**
- **0.9-in. 50-pin DIP**

PRODUCT DESCRIPTION

The Signetics 8X305 MicroController (Figure 1) is a high-speed bipolar microprocessor implemented with low-power Schottky technology. In a single chip, the 8X305 combines speed, flexibility, and a bit-oriented instruction set. These features and other basic characteristics of the chip combine to provide cost-effective solutions for a broad range of applications. The 8X305 is particularly useful in systems that require high-speed bit manipulations — sophisticated controllers, data communications, very fast interface control, and other applications of a similar nature.

The 8X305 can fetch, decode, and execute a 16-bit instruction word in a minimum of 200 nanoseconds. Within one instruction cycle, the 8-bit data-processing path can be programmed to rotate, mask, shift, and/or merge single or multiple bit subfields and, in addition, perform an ALU operation; in the same instruction, an external data field can be input, processed, and output to a specified destination — likewise, single or multiple bit data fields can be internally moved from a given source to a given destination. To summarize, fixed or variable-length data fields can be fetched, processed, operated on by the ALU, and moved to a different location — all in a timeframe of 200 nanoseconds. To interface with I/O and program memory, the 8X305 uses a 13-bit instruction address bus, a 16-bit instruction bus, an 8-bit bidirectional multiplexed I/O data/address bus and a 5-bit I/O control bus.

A wide selection of I/O devices, interface chips, and special-purpose parts are available for systems use. In most applications, the more powerful 8X305 is functionally interchangeable with its predecessor — the 8X300.

ASSOCIATED DOCUMENTATION

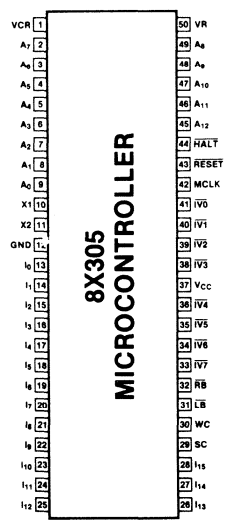
Other documents directly relating to *design* and *applications use* of the 8X305 MicroController are:

- Product Capabilities Manual
- 8X305 Users Manual

These documents and other current literature (Data Sheets, Product Bulletins, Applications Notes, etc.) are available at all Signetics Sales and Service Offices—see rear cover of this data sheet for the office in your locality.

PIN CONFIGURATION

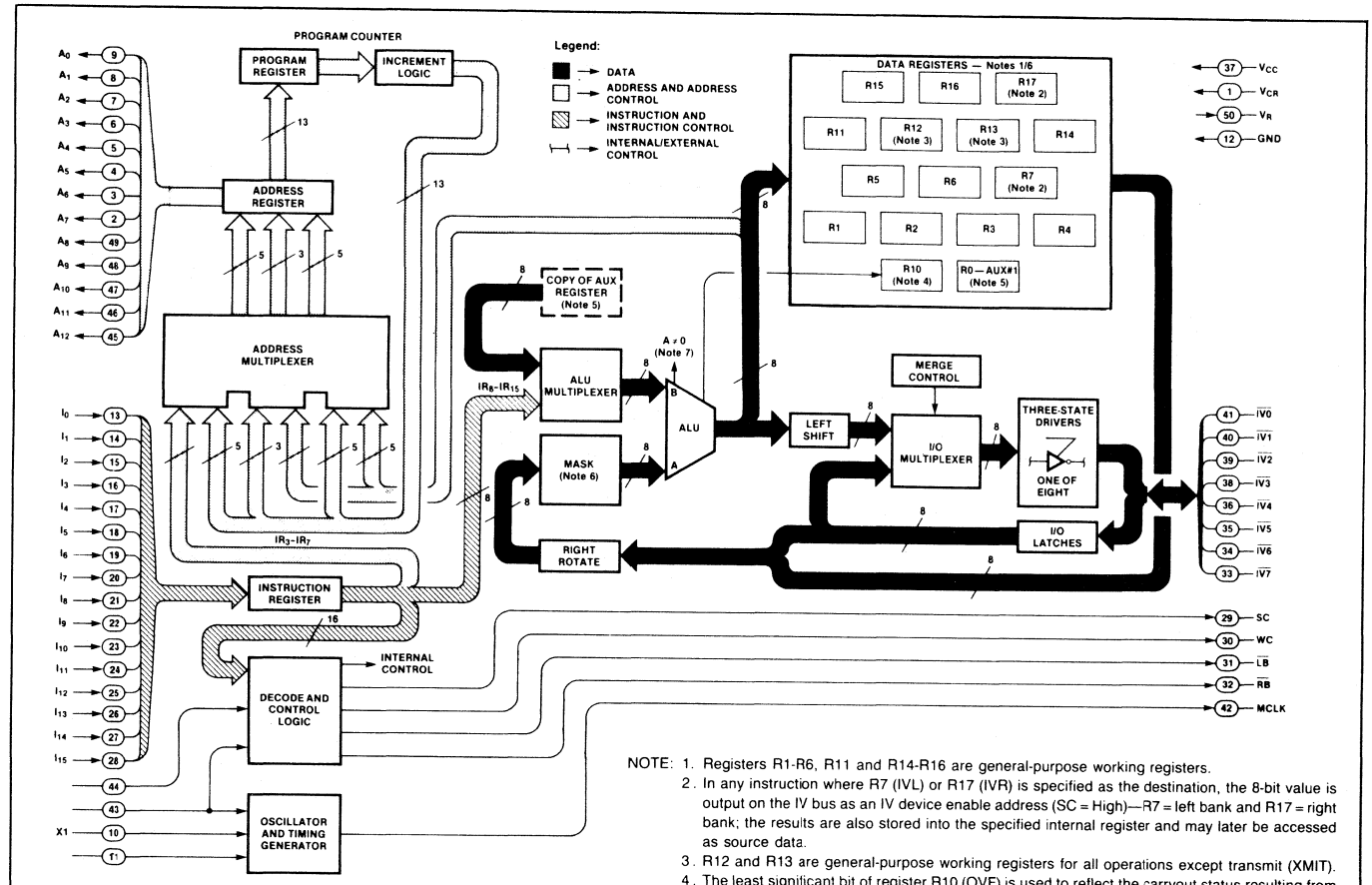
N, I PACKAGE



TOP VIEW

ORDER NUMBERS

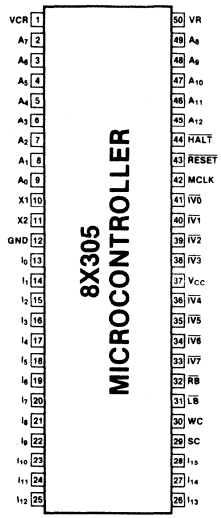
8X305N, 8X305I
S8X305I/883B, S8X305I/883C



NOTE: 1. Registers R1-R6, R11 and R14-R16 are general-purpose working registers.

2. In any instruction where R7 (IVL) or R17 (IVR) is specified as the destination, the 8-bit value is output on the IV bus as an IV device enable address (SC = High)—R7 = left bank and R17 = right bank; the results are also stored into the specified internal register and may later be accessed as source data.
3. R12 and R13 are general-purpose working registers for all operations except transmit (XMIT).
4. The least significant bit of register R10 (OVF) is used to reflect the carryout status resulting from the most recent ADD operation.
5. Auxiliary register R0 #1 is a general-purpose working register that holds the implied operand for Arithmetic and Logical operations; the content of this register is repeated in AUX #2 (shown dotted). The duplicate register is physically part of the ALU and is shown separate only for layout convenience.
6. Internal working registers cannot be operated on by the MASK logic.
7. During NZT instructions the ALU tests for all bits equal to "0" (Transfer if A ≠ 0)—refer to BASIC OPERATIONS that follow.

Figure 1. Architecture and Pin Designations for 8X305 MicroController



PIN NO.	IDENTIFIER	FUNCTION
1	VCR	Regulated voltage input from series-pass transistor (2N5320 or equivalent).
2-9, 45-49	$A_0 - A_{12}$	**Program Address Lines:** These active-high outputs permit direct addressing of up to 8192 words of program storage; A_{12} is least significant bit.
10, 11	X1, X2	Timing generator connections for a capacitor, a series resonant crystal, or an external clock source with complementary outputs.
12	GND	Ground.
13-28	$I_0 - I_{15}$	**Instruction Lines:** These active-high input lines receive 16-bit instructions from program storage; I_{15} is least significant bit.
29	SC	**Select Command:** When high (binary 1), an address is being output on pins $\overline{IV0}$ through $\overline{IV7}$.
30	WC	**Write Command:** When high (binary 1), data is being output on pins $\overline{IV0}$ through $\overline{IV7}$.
31	$\overline{LB}$	**Left Bank Control:** When low (binary 0), devices connected to the Left Bank are accessed. (Note. *Typically, the $\overline{LB}$ signal is tied to the $\overline{ME}$ input pin of I/O peripherals).*
32	RB	**Right Bank Control:** When low (binary 0), devices connected to the Right Bank are accessed (Note. *Typically, the $\overline{RB}$ signal is tied to the $\overline{ME}$ input pin of I/O peripherals).*
33-36, 38-41	$\overline{IV0}$-$\overline{IV7}$	**Interface Vector** (Input/Output Bus) — these bidirectional active-low three-state lines communicate data and/or addresses to I/O devices and memory locations. A low voltage level equals a binary "1"; $\overline{IV7}$ is Least Significant Bit.
37	V_{CC}	+5V power supply.
42	MCLK	**Master Clock:** This active-high output signal is used for clocking I/O devices and/or synchronization of external logic.
43	$\overline{RESET}$	When **$\overline{RESET}$** input is low (binary 0), the 8X305 is initialized — sets Program Counter/Address Register to zero and inhibits MCLK. For the period of time $\overline{RESET}$ is low, the Left Bank/Right Bank ($\overline{LB}$/$\overline{RB}$) signals are forced high asynchronously.
44	$\overline{HALT}$	When **$\overline{HALT}$** input is low (binary 0), internal operation of the 8X305 stops at the start of next instruction; MCLK is not inhibited nor is any internal register affected; however, both the Left Bank/Right Bank ($\overline{LB}$/$\overline{RB}$) signals are synchronously driven high during the first quarter of the instruction cycle time and remain high during the time HALT is low.
50	VR	Internally-generated reference output voltage for external series-pass regulator transistor.

Figure 2. Designations and Descriptions for Pins of 8X305 MicroController.

FUNCTIONAL OPERATION

Typical System Configuration

Although the system hookup shown in Figure 3 is of the simplest form, it provides a fundamental look at the 8X305 MicroController and peripheral relationships. As indicated, the 8X305 can directly address up to 8K words of program storage — either ROM or PROM. The user interface ($\overline{IV0}$ through $\overline{IV7}$) is capable of uniquely addressing 256 Input/Output locations and, with additional bank bits ($\overline{LB}$, $\overline{RB}$), this number is expanded to 512 — each bank comprising 256 addressable locations. The addressable locations of each bank can be used in a variety of ways; a simple method of implementation is shown in Figure 3. When $\overline{LB}$ is active low, the left bank is enabled and any one of 256 locations within the RAM memory can be accessed for input/output operations. A similar set of "enable/access" conditions are applicable to the right bank when $\overline{RB}$ is active low.

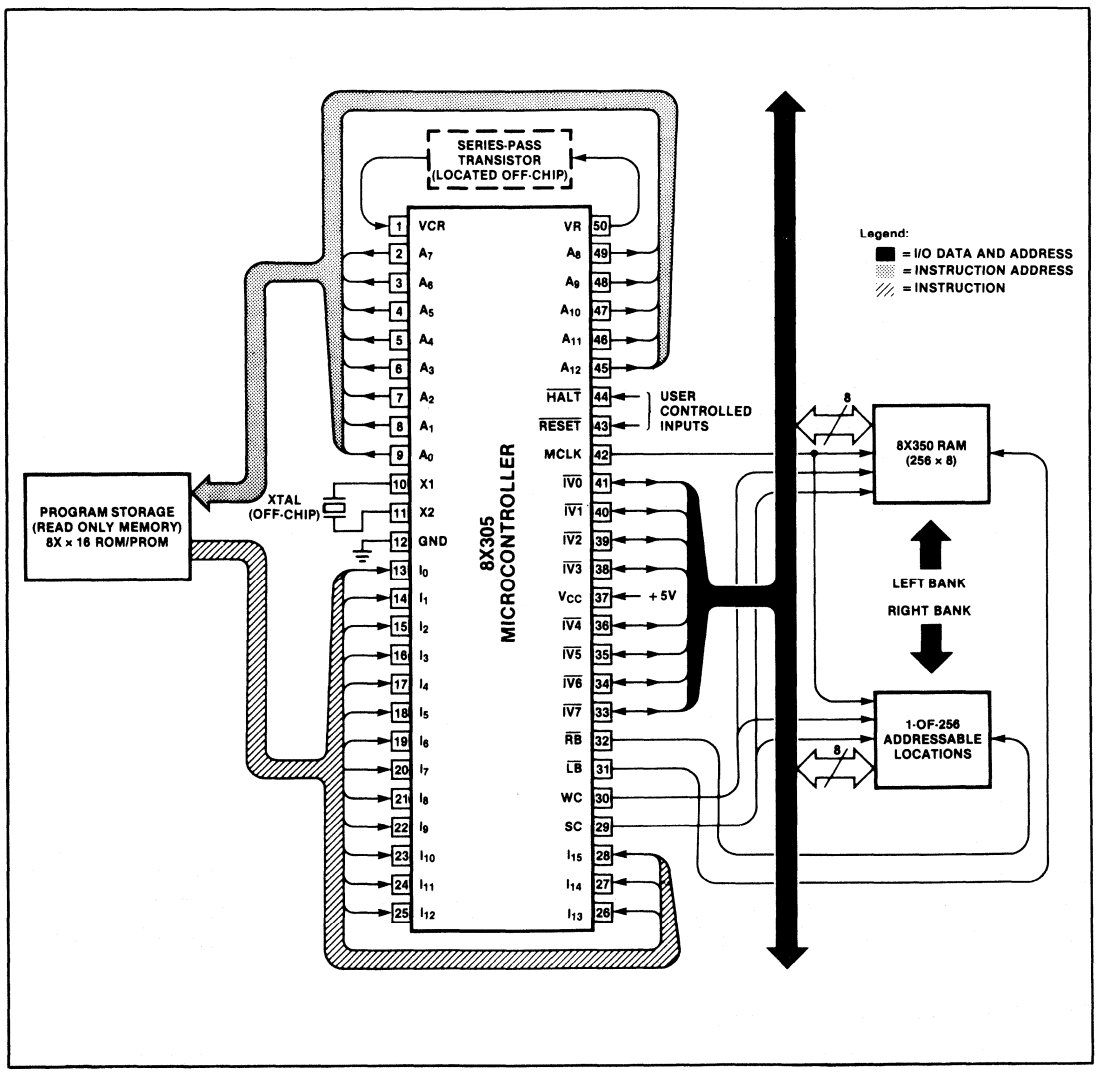


Figure 3. Typical 8X305 System Hookup

BASIC OPERATIONS OF 8X305

Refer to a later discussion of "Instruction Fields" for a detailed examination of all operand fields and subdivisions thereof—"S" (S_0, S_1), "D" (D_0, D_1), "R", "L", "J", and "A".

MOVE OPERATIONS

REGISTER-TO-REGISTER

SOURCE	DATA PROCESSING (PRE-ALU)	ALU	DATA PROCESSING (POST-ALU)	DESTINATION
R0-R17 as specified by "S" field of instruction.	Right rotate as specified by "R" field of instruction.	No operation.	No operation.	R0-R7, R11-R17 as specified by "D" field of instruction.

REGISTER-TO-$\overline{IV}$ BUS

SOURCE	DATA PROCESSING (PRE-ALU)	ALU	DATA PROCESSING (POST-ALU)	DESTINATION
R0-R17 as specified by "S" field of instruction.	No operation.	No operation.	Shift and merge as specified by "D_0" and "L" fields of instruction.	Variable length field of $\overline{IV}$ bus—Left Bank ($\overline{LB}$) or Right Bank ($\overline{RB}$) as specified by "D_0" and "D_1" fields of instruction.

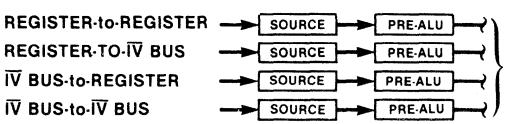
$\overline{IV}$ BUS-TO-REGISTER

SOURCE	DATA PROCESSING (PRE-ALU)	ALU	DATA PROCESSING (POST-ALU)	DESTINATION
Variable length field of $\overline{IV}$ bus—Left Bank ($\overline{LB}$) or Right Bank ($\overline{RB}$) as specified by "S_1" and "S_0" fields of instruction.	Right rotate and mask as specified by "S_0" and "L" fields of instruction.	No operation.	No operation.	R0-R7, R11-R17 as specified by "D" field of instruction.

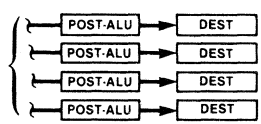
$\overline{IV}$ BUS-TO-$\overline{IV}$ BUS

SOURCE	DATA PROCESSING (PRE-ALU)	ALU	DATA PROCESSING (POST-ALU)	DESTINATION
Variable length field of $\overline{IV}$ bus—Left Bank ($\overline{LB}$) or Right Bank ($\overline{RB}$) as specified by "S_1" and "S_0" fields of instruction.	Right rotate and mask as specified by "S_0" and "L" fields of instruction.	No operation.	Shift and merge as specified by "D_0" and "L" fields of instruction.	Variable length field of $\overline{IV}$ bus—Left Bank ($\overline{LB}$) or Right Bank ($\overline{RB}$) as specified by "D_0" and "D_1" fields of instruction.

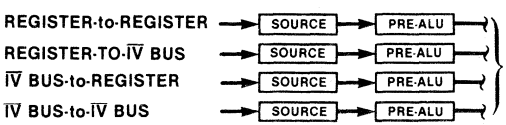
ADD OPERATIONS



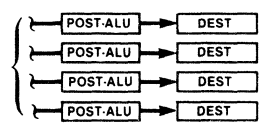
Same as MOVE operations, except source data is ADDed to contents of AUXiliary Register R0 via the ALU; if appropriate, Overflow Register R10 (OVF) is also set.



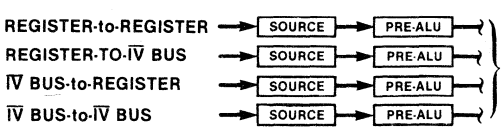
AND OPERATIONS



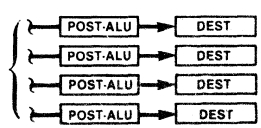
Same as MOVE operations, except source data is ANDed with contents of AUXiliary Register R0 via the ALU.



EXCLUSIVE OR (XOR) OPERATIONS



Same as MOVE operations, except source data is Exclusively ORed with contents of AUXiliary Register R0 via the ALU.



EXECUTE (XEC) OPERATIONS

XEC, REGISTER

SOURCE	DATA PROCESSING (PRE-ALU)	ALU	DATA PROCESSING (POST-ALU)	DESTINATION
R0-R17 as specified by "S" field of instruction.	No operation.	Add source data to 8-bit field specified by instruction literal ($0 \leq J \leq 377_8$).	No operation.	Replace 8 LSB of Address Register* with 8-bit sum from ALU. *PGM CTR unchanged.

XEC, $\overline{IV}$ BUS

SOURCE	DATA PROCESSING (PRE-ALU)	ALU	DATA PROCESSING (POST-ALU)	DESTINATION
Left or Right Bank of $\overline{IV}$ bus as specified by "S" field of instruction.	Rotate and mask as specified by "S_0" and "L" fields of instruction.	Add masked field of source data to 5-bit literal specified by "J" field of instruction ($0 \leq J \leq 37_8$).	No operation.	Replace 5 LSB of Address Register* with 5-bit sum from ALU. *PGM CTR unchanged.

NON-ZERO TRANSFER (NZT) OPERATIONS

NZT, REGISTER

SOURCE	DATA PROCESSING (PRE-ALU)	ALU	DESTINATION
R0-R17 as specified by "S" field of instruction.	No operation.	Test contents of source register for all zeroes.	If S = 0, increment PC by 1; if S ≠ 0, replace 8 LSB of AR and PC with literal specified by "J" field of instruction.

NZT, $\overline{IV}$ BUS

SOURCE	DATA PROCESSING (PRE-ALU)	ALU	DESTINATION
Left or Right Bank of $\overline{IV}$ bus as specified by "S" field of instruction.	Rotate and mask as specified by "S_0" and "L" fields of instruction.	Test contents of masked field for all zeroes.	If S = 0, increment PC by 1; if S ≠ 0, replace 5 LSB of AR and PC with literal specified by "J" field of instruction.

TRANSMIT (XMIT) OPERATIONS

XMIT, REGISTER

SOURCE	DATA PROCESSING (PRE-ALU)	ALU	DATA PROCESSING (POST-ALU)	DESTINATION
$0 \leq J \leq 377_8$ – value specified by "J" field of instruction.	No operation.	No operation.	No operation.	R0-R7, R11, R14-R17. Load 8-bit integer specified by "J" field into register specified by "D" field.

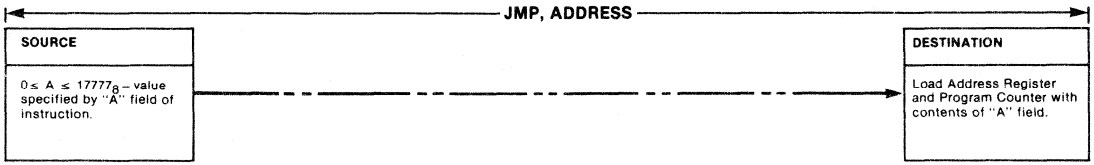
XMIT 8-BITS IMMEDIATE, $\overline{IV}$ BUS

SOURCE	DATA PROCESSING (PRE-ALU)	ALU	DATA PROCESSING (POST-ALU)	DESTINATION
$0 \leq J \leq 377_8$ – value specified by "J" field of instruction.	No operation.	No operation.	No operation.	Left (R12) or Right (R13) Bank of $\overline{IV}$ bus as specified by "D" field of instruction.

XMIT VARIABLE-BIT FIELD IMMEDIATE $\overline{IV}$ BUS

SOURCE	DATA PROCESSING (PRE-ALU)	ALU	DATA PROCESSING (POST-ALU)	DESTINATION
$0 \leq J \leq 37_8$ – value specified by "J" field of instruction.	No operation.	No operation.	Shift and merge source data as specified by "D_0" and "L" fields of instruction.	Left or Right Bank of $\overline{IV}$ bus as specified by "D" field of instruction.

JUMP (JMP) OPERATION



Program Storage Interface

As shown in Figure 3, program storage is connected to output address lines A_0 through A_{12} (A_{12} = LSB) and input instruction lines I_0 through I_{15}. An address output on A_0/A_{12} identifies one 16-bit instruction word in program storage. The instruction word is subsequently input on I_0/I_{15} and defines the MicroController operation which is to follow — one instruction word equals one completed operation. Any TTL-compatible memory can be used for program storage provided the worst-case access time is compatible with the instruction cycle time used for the application — see timing section for appropriate calculations.

I/O Interface and Control

An 8-bit bidirectional I/O bus, referred to as the Interface Vector ($\overline{IV}$) bus, provides a communication link between the MicroController and the two banks of I/O devices. The $\overline{LB}$ (Left Bank) and $\overline{RB}$ (Right Bank) control signals identify which bank is enabled; when both $\overline{LB}$ and $\overline{RB}$ are high (inactive), neither bank is enabled and the $\overline{IV}$ bus is inactive (three-state). A functional analysis of the Left and Right Bank signals is shown below:

$\overline{LB}$	$\overline{RB}$	FUNCTION
Low	Low	This state is not generated by the 8X305.
Low	High	Enable left bank devices.
High	Low	Enable right bank devices.
High	High	Disable all devices; $\overline{IV}$ bus is three-state.

Both data and I/O address information are multiplexed on the $\overline{IV}$ bus. The SC (Select Command) and WC (Write Command) signals distinguish between data and I/O address information as follows:

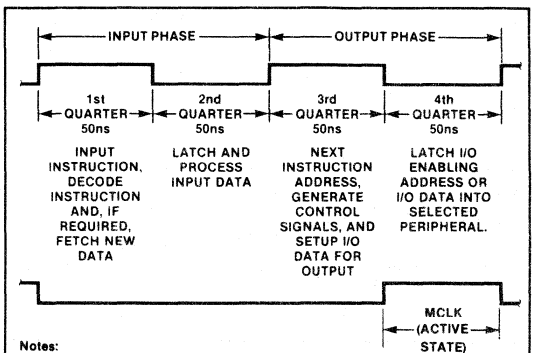
LB/RB	SC	WC	FUNCTION
High	Low	Low	The $\overline{IV}$ bus is three-state and not looking for input data.
Low	Low	Low	The $\overline{IV}$ bus is reading input data.
Low	Low	High	Data is being output.
Low	High	Low	Address is being output.
X	High	High	This condition is never generated.

Data Processing

Basically, the data processing path of the 8X305 consists of the Rotate/Mask logic, the Arithmetic Logic Unit (ALU), the Shift/ Merge functions, on-chip memory (sixteen 8-bit registers), and the bidirectional $\overline{IV}$ bus interface with its associated driver circuits and internal latches. The on-board memory and the $\overline{IV}$ bus are connected to both inputs and outputs of the ALU via internal 8-bit data paths — see Figure 1. Inputs to the ALU are preceded by right-rotate and data-mask functions; the ALU output is followed by the left-shift and merge operations. Depending on the desired operation, any one or all of the functions (Rotate/Mask/Shift/Merge) can operate on 8 bits of data in a single instruction cycle. For a summary of all data-processing capabilities, refer to BASIC OPERATIONS OF THE 8X305 described earlier in this data sheet.

Instruction Cycle

Each operation of the 8X305 is executed in a single instruction cycle. The instruction cycle is internally divided into four equal parts — each part being as short as 50 nanoseconds. Figure 4 shows the general functions that



Notes:
1. New instruction must be accepted and latched at end of first quarter cycle.
2. The I/O data latches are open for the first two quarter cycles, that is, for 100 nanoseconds.
3. The address changes during third quarter cycle.
4. $\overline{IV}$ bus drivers are active (turned on) during third and fourth quarter cycles.

Figure 4. Instruction Cycle and MCLK with: Crystal = 10MHz and Cycle Time = 200 nanoseconds.

occur during each quarter cycle; specifics regarding minimum/maximum timing and other critical values are described later in this data sheet. During the first quarter cycle, a new instruction from program storage is input via I_0-I_{15} and decoded. If an I/O operation is indicated, new data is fetched from a specified internal register or via the $\overline{IV}$ bus. At the end of the first quarter cycle, the new instruction is latched into the instruction register.

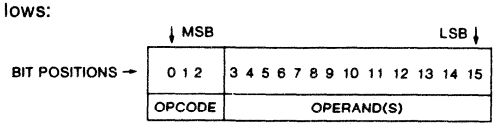
In the second quarter cycle, the I/O input data stabilizes and preliminary processing is completed; at the end of this quarter, the $\overline{IV}$ latches close and final processing can be accomplished, thus completing the input phase of the instruction cycle. During the third quarter cycle, the address for the next instruction is output to the instruction address bus, $\overline{IV}$ control signals are generated, and both data and destination are setup for the remainder of the output phase. During the fourth quarter cycle, a master clock signal (MCLK) generated by the 8X305 is used to latch either the I/O-enabling address or the I/O data into peripheral devices connected to the $\overline{IV}$ bus; MCLK can also be used to synchronize any external logic with timing circuits of the 8X305. To summarize the action, the first half of the instruction cycle deals primarily with input functions and the second half is mostly concerned with output functions.

INSTRUCTION SET

General Format and Operating Principles

The 16-bit instruction word (I_0 through I_{15}) from program storage is input to the instruction register (Figure 1) and is subsequently decoded to implement the events to occur during the current instruction cycle.

The general format for each instruction word is as follows:



The 3-bit operation code (OPCODE) define any one of eight classes of instructions; variations within each class are specified by the remaining thirteen operand bits. The eight instruction classes can be separated into two control areas — *data* and *program;* general functions within these areas are:

- Data Control —
  - ADD, AND, XOR } Arithmetic and Logic Operations
  - MOVE, XMIT } Movement of Data and Constants
- Program Control
  - XEC, NZT, JMP } Branch or Test

Instruction Fields

As shown in Table 1, each instruction word consists of an operation code (OPCODE) field and from one to three operand fields. The possible operand fields are: Source (S), Destination (D), Rotate/Length (R/L), Literal (J), and Address (A). The OPCODE and operand fields are described in the paragraphs that follow the table.

Table 1. FUNCTIONAL DESCRIPTION OF INSTRUCTION SET

INSTRUCTION WORD	DESCRIPTION	STATE OF CONTROL SIGNAL DURING INSTRUCTION CYCLE — SEE FIGURE 4		
		CONTROL SIGNAL	INPUT PHASE	OUTPUT PHASE
CLASS = **MOVE** OPCODE = **0** OPERATION = **(S) → D**				
Register-to-Register 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 OPCODE | S | R | D $S = 00_8$-17_8 $D = 00_8$-07_8, 11_8-17_8	Move content of internal register specified by S-field to internal register specified by D-field. Prior to the "MOVE" operation, right-rotate contents of internal source register by octal value (0 through 7) defined by the R-field.	SC	L	H if $D = 07_8, 17_8$
		WC	L	L
		$\overline{LB}$	H	L if $D = 07_8$
		$\overline{RB}$	H	L if $D = 17_8$
Register-to-$\overline{IV}$ Bus (Note) 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 OPCODE | S | L | D (D_1 : D_0) $S = 00_8$-17_8 $D = 20_8$-37_8	Move contents of internal register specified by the S-field to the $\overline{IV}$ bus. Before outputting on $\overline{IV}$ bus, data is shifted as specified by the least significant octal digit of the D-field and the bits specified by the L-field are merged with the latched I/O data.	SC	L	L
		WC	L	H
		$\overline{LB}$	L if $D = 20_8$-27_8	L if $D = 20_8$-27_8
		$\overline{RB}$	L if $D = 30_8$-37_8	L if $D = 30_8$-37_8

Table 1. FUNCTIONAL DESCRIPTION OF INSTRUCTION SET (Continued)

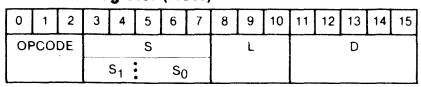
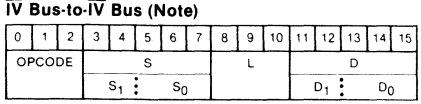
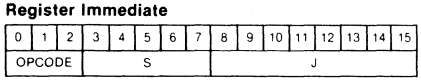
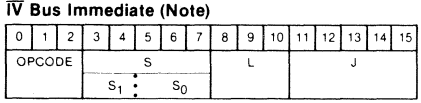
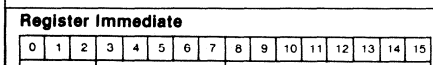
INSTRUCTION WORD	DESCRIPTION	STATE OF CONTROL SIGNAL DURING INSTRUCTION CYCLE — SEE FIGURE 3		
		CONTROL SIGNAL	INPUT PHASE	OUTPUT PHASE
CLASS = MOVE OPCODE = 0 OPERATION = (S) → D				
$\overline{IV}$ Bus-to-Register (Note)  $S = 20_8-37_8$ $D = 00_8-07_8, 11_8-17_8$	Move right-rotated $\overline{IV}$ bus (source) data specified by the S-field to internal register specified by the D-field. The L-field specifies the length of source data starting from the LSB-position and, if less than 8 bits, the remaining bits are filled with zeros.	SC	L	H if D = 07_8, 17_8
		WC	L	L
		$\overline{LB}$	L if S = 20_8-27_8	L if D = 07_8
		$\overline{RB}$	L if S = 30_8-37_8	L if D = 17_8
$\overline{IV}$ Bus-to-$\overline{IV}$ Bus (Note)  $S = 20_8-37_8$ $D = 20_8-37_8$	Move right-rotated $\overline{IV}$ bus (source) data specified by the S-field to the I/O latches. Before outputting on $\overline{IV}$ bus, shift data as specified by the D-field; then merge source and latched I/O data as specified by the L (length) field.	SC	L	L
		WC	L	H
		$\overline{LB}$	L if S = 20_8-27_8	L if D = 20_8-27_8
		$\overline{RB}$	L if S = 30_8-37_8	L if D = 30_8-37_8
CLASS = ADD OPCODE = 1 OPERATION = (S) + (AUX) → D				
Same as MOVE instruction class	Same as MOVE instruction class except that contents of AUX (R0) register are ADDed to the source data. If there ia a "carry" from MSB, then R10 (OVF) = 1 (overflow), otherwise OVF = 0.	Same as MOVE instruction class		
CLASS = AND OPCODE = 2 OPERATION = (S) ∧ (AUX) → D				
Same as MOVE instruction class	Same as MOVE instruction class except that contents of AUX (R0) register are ANDed with source data.	Same as MOVE instruction class		
CLASS = XOR OPCODE = 3 OPERATION = (S) ⊕ (AUX) → D				
Same as MOVE instruction class	Same as MOVE instruction class except that contents of AUX (R0) register are Exclusively ORed with source data.	Same as MOVE instruction class		
CLASS = XEC OPCODE = 4 OPERATION = Refer to Description				
Register Immediate  $S = 00_8-17_8$ $J = 000_8-377_8$	Execute instruction at current page address offset by J (literal) + (S). Return to normal instruction flow unless a branch is encountered. Execute instruction at an address determined by replacing the low-order 8 bits of the Address Register with the following derived sum: Value of literal (J-field) plus contents of internal register specified by S-field The PC is not incremented and the overflow status (OVF) is not changed.	SC	L	L
		WC	L	L
		$\overline{LB}$	H	H
		$\overline{RB}$	H	H
$\overline{IV}$ Bus Immediate (Note)  $S = 20_8-37_8$ $J = 00_8-37_8$	Execute instruction at an address determined by replacing the low-order 5 bits of Address Register with the following derived sum: 5-bit value of literal (J-field) plus value of rotated source data specified by S-field. The L-field specifies the length of source data starting from the LSB position and, if less than 8 bits, the remaining bits are filled with zeros; the Program Counter is not incremented and the overflow status (OVF) is not changed.	SC	L	L
		WC	L	L
		$\overline{LB}$	L if S = 20_8-27_8	H
		$\overline{RB}$	L if S = 30_8-37_8	H
CLASS = NZT OPCODE = 5 OPERATION = Refer to Description				
Register Immediate  $S = 00_8-17_8$ $J = 000_8-377_8$	If data specified by the S-field is not equal to zero, jump to current page address offset by value of J-field; otherwise, increment the Program Counter. If contents of internal register specified by S-field is non-zero, transfer to address determined by replacing the low-order 8 bits of Address Register and Program Counter with "J", otherwise, increment PC.	SC	L	L
		WC	L	L
		$\overline{LB}$	H	H
		$\overline{RB}$	H	H

Table 1. FUNCTIONAL DESCRIPTION OF INSTRUCTION SET (Concluded)

INSTRUCTION WORD	DESCRIPTION	STATE OF CONTROL SIGNAL DURING INSTRUCTION CYCLE — SEE FIGURE 3		
		CONTROL SIGNAL	INPUT PHASE	OUTPUT PHASE
CLASS = **NZT** OPCODE = **5** OPERATION = **Refer to Description**				
$\overline{IV}$ Bus Immediate (Note) 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 OPCODE | S (S_1 ⋮ S_0) | L | J $S = 20_8$-37_8 $J = 00_8$-37_8	If right-rotated and masked IV bus is non-zero, transfer to address determined by replacing low-order 5 bits of Address Register and Program Counter with "J", otherwise, increment PC. (The L-field specifies the length of source I/O data starting from the LSB-position and, if less than 8 bits, the remaining bits are filled with zeros.)	SC	L	L
		WC	L	L
		$\overline{LB}$	L if $S = 20_8$-27_8	H
		$\overline{RB}$	L if $S = 30_8$-37_8	H
CLASS = **XMIT** OPCODE = **6** OPERATION = **J → D**				
XMIT, Register 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 OPCODE | D | J $D = 00_8$-06_8, 11_8, 14_8-16_8 $J = 000_8$-377_8	Store 8-bit value specified by "J" into register specified by "D".	SC	L	L
		WC	L	L
		$\overline{LB}$	H	H
		$\overline{RB}$	H	H
XMIT, $\overline{IV}$ Bus Address 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 OPCODE | D | J $D = 07_8$, 17_8 $J = 000_8$-377_8	Enable I/O device on the bank specified by "D", whose address is the 8-bit integer specified by "J". Address "J" is stored in register "D"	SC	L	H
		WC	L	L
		$\overline{LB}$	H	L if $D = 07_8$
		$\overline{RB}$	H	L if $D = 17_8$
XMIT 8 Bits Immediate, $\overline{IV}$ Bus (Note) 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 OPCODE | D | J $D = 12_8$-13_8 $J = 000_8$-377_8	Store value of 8-bit integer in the previously enabled I/O port, at the bank destination ($\overline{LB}$ or $\overline{RB}$) specified by "D". Contents of R12 or R13 remain unchanged.	SC	L	L
		WC	L	H
		$\overline{LB}$	H	L if $D = 12_8$
		$\overline{RB}$	H	L if $D = 13_8$
XMIT Variable Bit Field Immediate, $\overline{IV}$ Bus (Note) 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 OPCODE | D (D_1 ⋮ D_0) | L | J $D = 20_8$-37_8 $J = 00_8$-37_8	Transmit Least Significant "L" bits of "J" field to "L-bit" field of $\overline{IV}$ bus specified by "D"; if "L" is greated than 5 bits, the MSB bits of destination field is filled with zeros.	SC	L	L
		WC	L	H
		$\overline{LB}$	L if $D = 20_8$-27_8	L if $D = 20_8$-27_8
		$\overline{RB}$	L if $D = 30_8$-37_8	L if $D = 30_8$-37_8
CLASS = **JMP** OPCODE = **7** OPERATION = **Refer to Description**				
Address Immediate 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 OPCODE | A $A = 00000_8$-17777_8	Jump to address in program storage specified by A-field; this address is loaded into the Address Register and the *Program Counter*.	SC	L	L
		WC	L	L
		$\overline{LB}$	H	H
		$\overline{RB}$	H	H

Note:
S_0 specifies the LSB of rotated input data field
S_1 specifies the bank of $\overline{IV}$ bus from which source data will be input
D_0 specifies bit position in I/O device with which LSB of processed data will be aligned and
D_1 specifies the bank of $\overline{IV}$ bus which will be the destination.

Operations Code Field. The 3-bit OPCODE field specifies one of eight classes of 8X305 instructions; octal designations for this field and operands for each instruction class are shown in the preceding table.

Source (S) and Destination (D) Fields. The 5-bit "S" and "D" fields specify the source and destination, respectively, for whatever operation is defined by the OPeration CODE. The "S" and/or "D" fields can specify an internal 8X305 register or any one-to-eight bit field within an I/O device; octal values and source/destination field assignments for all internal registers are shown in Table 2.

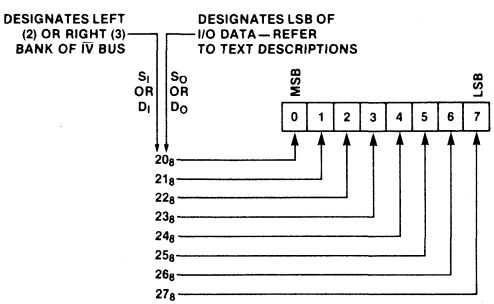
Table 2. OCTAL ADDRESSES AND SOURCE/DESTINATION FIELDS FOR 8X305 REGISTERS

ADDRESS	REGISTER DESIGNATION	SOURCE	DESTI-NATION	ADDRESS	REGISTER DESIGNATION	SOURCE	DESTI-NATION
00_8	R0 (AUX)—General purpose register	X	X	10_8	R10 (OVF—Overflow register)	X	
01_8	R1—General purpose register	X	X	11_8	R11—General purpose register	X	X
02_8	R2—General purpose register	X	X	12_8	R12—General purpose register (Note)	X	X
03_8	R3—General purpose register	X	X	13_8	R13—General purpose register (Note)	X	X
04_8	R4—General purpose register	X	X	14_8	R14—General purpose register	X	X
05_8	R5—General purpose register	X	X	15_8	R15—General purpose register	X	X
06_8	R6—General purpose register	X	X	16_8	R16—General purpose register	X	X
07_8	R7—Special purpose register (refer to next paragraph)	X	X	17_8	R17—Special purpose register (refer to next paragraph)	X	X

Note:
R12 and R13 function as general purpose working registers for all operations except transmit (XMIT). During a transmit instruction where R12 or R13 is the destination, the 8-bit "J" field is immediately transferred to the $\overline{IV}$ bus; for this operation, the contents of the designated register remain unchanged.

In instructions where $R7_8$ (IVL) or $R17_8$ (IVR) is specified as the destination, the 8-bit value is output on the $\overline{IV}$ bus as an I/O device address or memory location; register R7 selects the Left Bank and register R17 selects the Right Bank. The results are also stored into the specified internal register ($R7_8$ or $R17_8$) and may later be accessed as source data. When the $\overline{IV}$ bus is specified as a source and/or destination, the "S" and "D" fields are split into two parts, that is,

- Source (S) = S_1, S_0 and Destination (D) = D_1, D_0 where,
  S_0 specifies the LSB of rotated input data field
  S_1 specifies the bank of $\overline{IV}$ bus from which source data will be input
  D_0 specifies bit position in I/O device with which LSB of processed data will be aligned and
  D_1 specifies the bank of $\overline{IV}$ bus which will be the destination

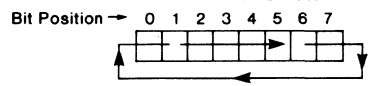


Notes:
1. The field length of 0-to-8 bits is specified by the "L" field.
2. For the Right Bank, 30_8-37_8 perform equivalent I/O functions.

Rotate (R) and Length (L) Field. The 3-bit R/L field performs one of two functions, specifying either the field length (L) for I/O operations or a right-rotate (R) for internal operations. For a given instruction, the specified function depends upon the contents of the Source (S) and Destination (D) fields.

When an internal register is specified by both the source and destination fields, the "R" field is invoked and it specifies a right-rotate of the data specified in the "S" field — see accompanying diagram. The source-register data (up to 8 bits) is right-rotated during the "input phase" of the instruction cycle (Figure 4) and this function is always performed prior to any ALU operation. *(Note: The right-rotate function is implemented on the bus and not in the source register.)*

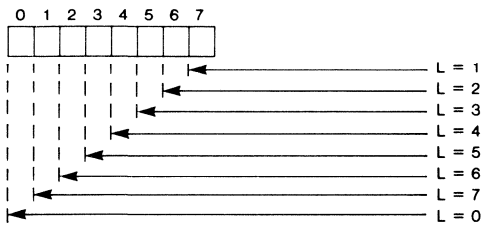
RIGHT-ROTATE FUNCTION



When either or both of the source and destination fields specify a variable-length I/O data field, the "L" field specifies the length of the I/O data field — see following diagram. If the source field specifies an $\overline{IV}$ address (20_8-37_8) and the destination field specifies an internal register (00_8-07_8, 11_8-17_8), the "L" field specifies the length of source data; the source data is formed by right-rotating the $\overline{IV}$ bus data according to the source address and then masking result as specified by the "L" field. If length is less than 8 bits, all remaining bits are set to zero prior to processing data in the ALU. If the source field

specifies an internal register (00_8–17_8) and the destination field specifies $\overline{IV}$ bus data (20_8–37_8), the "L" field specifies the length of the destination data. To form the destination data, the ALU output is left-shifted according to the destination address and then masked to the required length — see $\overline{IV}$ DATA LENGTH SPECIFICATION. The destination data is merged with data in the I/O latches to finalize the $\overline{IV}$ bus data. Hence, a one-to-eight bit destination data field can be inserted into the existing 8-bit I/O port without modifying surrounding bits. If both the source and destination fields specify $\overline{IV}$ bus data (20_8–37_8), the "L" field specifies the length of both the source and destination data.

$\overline{IV}$ DATA LENGTH SPECIFICATION
(No Rotate Function Specified)



To form the source data, the $\overline{IV}$ bus input data is right-rotated according to the source address and then masked to the required length—see $\overline{IV}$ DATA LENGTH SPECIFICATION. If length is less than 8 bits, all remaining bits are set to zero before processing in the ALU. To form the destination data, the ALU output is left-shifted according to the destination address and masked to the required length specification. The destination data is then merged into the $\overline{IV}$ bus data that was used to obtain the source; thus, if the source and destination addresses are on the same bank, the $\overline{IV}$ bus data written to the destination I/O Port appears unmodified, except for bits changed during the shift-and-mask operations. If the source and destination addresses refer to different banks, the destination I/O Port is changed to contain the contents of the source I/O Port in those bit positions not affected by the destination data.

J Field. The 5-bit or 8-bit "J" field is used to load a literal value (contained in the instruction) into a register, into a variable I/O data field, or to modify the low-order bits of the Program Counter. The bit length of the "J" field is implied by the "S" and "L" fields in the XEC, NZT, and XMIT instructions, based on the following conditions:

- When the Source (S) field specifies an internal register, the literal value of the "J" field is an 8-bit binary number.
- When the Source (S) field specifies a variable I/O data field, the literal value of the "J" field is a 5-bit binary number.

A Field. The 13-bit "A" field is an address field which allows the 8X305 to directly branch to any of the 8192 locations in Program Storage memory.

Formation of Instruction Address

The Address Register and Program Counter are used to generate addresses for accessing an instruction from program storage. The instruction address is formed in one of the following ways:

- For all except the JMP, XEC, and a "satisfied" NZT instruction, the Program Counter is incremented by one and placed in the Address Register.
- For the JMP instruction, the 13-bit "A" field contained in the JMP instruction word replaces the contents of both the Address Register and the Program Counter.
- For the XEC instruction, the Address Register is loaded with bits from the Program Counter modified as follows:

  XEC using $\overline{IV}$ Bus Data — low-order 5 bits of ALU output replaces counterpart bits in Address Register

  XEC using Data from Internal Register — low-order 8 bits of ALU output replaces counterpart bits in Address Register

  The Program Counter is not modified for either of the above conditions.
- For a "satisfied" NZT instruction, the low-order 5 bits (NZT source is $\overline{IV}$ bus data) or low-order 8 bits (NZT source is an internal register) of both the Address Register and Program Counter are loaded with the literal value specified by the "J" field of instruction word.

Data Addressing

The source and/or destination addresses of the data to be operated upon are specified as part of the instruction word. As shown earlier, source/destination addresses are specified using a 5-bit code (00_8–37_8). When the most significant octal digit is a "0" or "1", the source and/or destination address is an internal register; if the most significant digit is a 2 or 3, an $\overline{IV}$ bus operation is indicated — 2 specifying a Left-Bank ($\overline{LB}$) operation and 3 specifying a Right-Bank ($\overline{RB}$) operation. The least significant octal digit (0 through 7) indicates either a specific internal register address or positioning information for the least significant bit when specifying $\overline{IV}$ bus data. Referring to Table 1, AUXiliary register R0 (00_8) is the implied source

of the second argument for the ADD, AND, and XOR operations. IVL register R7 and IVR register R17 (destination addresses 07_8 and 17_8, respectively) provide a means of routing enabling address information to I/O peripherals. With IVL or IVR specified as the destination address, data is placed on the $\overline{IV}$ bus during the output phase of the instruction cycle; simultaneously, a Select Command (SC) is generated to inform all I/O devices that information on the $\overline{IV}$ bus is to be considered as an I/O address. Since the contents of IVL and IVR are preserved, either register may later be accessed as a source of data.

Control outputs $\overline{LB}$ and $\overline{RB}$ are used to partition I/O bus devices into two fields of 256 addresses. With $\overline{LB}$ in the active-low state and a source address of 20_8–27_8, the left bank of I/O devices are enabled during the input phase of the instruction cycle. With $\overline{RB}$ in the active-low state and a source address of 30_8–37_8, the right bank of devices are enabled. During the output phase, $\overline{LB}$ is low if the destination address is 07_8 or 20_8–27_8, whereas $\overline{RB}$ is low if the destination address is 17_8 or 30_8–37_8. Each address field ($\overline{LB}$ and $\overline{RB}$) can have a different I/O device selected, that is, data can be transferred from a device in one bank to a device in the other in one instruction cycle.

DESIGN PARAMETERS

Hardware design of an 8X305-based system largely consists of the following operations:

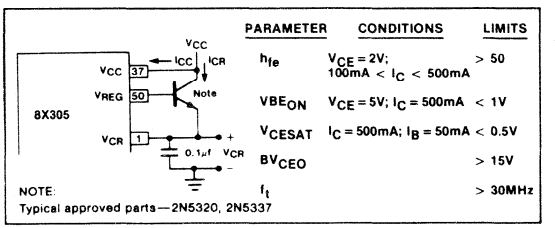
- Selecting and interfacing a Program Storage device — ROM, PROM, etc.
- Selecting and interfacing input/output devices — RAM, Ports, and other 8-bit addressable I/O devices.
- Choosing and implementing System Clock — Capacitor-Controlled, Crystal-Controlled, or Externally-Driven.
- Selection of an off-chip series-pass transistor.

All information required for easy implementation of these design requirements is provided under the following captions:

- Ordering Information
- Voltage Regulator
- DC Characteristics
- AC Characteristics
- Timing Considerations
- Clock Considerations
- $\overline{HALT/RESET}$ Logic

VOLTAGE REGULATOR

All internal logic of the 8X305 is powered by an on-chip voltage regulator that requires an external series-pass transistor. Electrical specifications for the off-chip power transistor and a typical hook-up are shown in the accompanying diagram. To minimize lead inductance, the transistor should be as close as possible to the 8X305 package and the emitter should be ac-grounded via a 0.1 microfarad ceramic capacitor.



PARAMETER	CONDITIONS	LIMITS
h_{fe}	V_{CE} = 2V; 100mA < I_C < 500mA	> 50
VBE_{ON}	V_{CE} = 5V; I_C = 500mA	< 1V
V_{CESAT}	I_C = 500mA; I_B = 50mA	< 0.5V
BV_{CEO}		> 15V
f_t		> 30MHz

NOTE:
Typical approved parts—2N5320, 2N5337

DC CHARACTERISTICS (Commercial Part) $4.75V \leq V_{CC} \leq 5.25V$, $0°C \leq T_A \leq 70°C$

ABSOLUTE MAXIMUM RATINGS

Storage Temperature (T_{STG}) ratings are from -65 to $+150°C$

PIN	DESCRIPTION	RATING	UNIT	PIN	DESCRIPTION	RATING	UNIT
V_{CC}	Supply voltage	+7.0	V	All other pins	Logic input voltage	5.5	V
X1, X2	Crystal input voltage	2.0	V				

PARAMETER		TEST CONDITIONS	LIMITS Min	LIMITS Typ	LIMITS Max	UNIT	COMMENTS
V_{CC}	Supply voltage		4.75	5.0	5.25	V	
V_{IH}	High level input voltage		0.6 2.0		2.0 5.5	V	X1 and X2 All other pins
V_{IL}	Low level input voltage				0.4 0.8	V	X1 and X2 All other pins
V_{OH}	High level output voltage	V_{CC} = min; $I_{OH} = -3mA$	2.4			V	
V_{OL}	Low level output voltage	V_{CC} = min; $I_{OL} = 6mA$ V_{CC} = min; $I_{OL} = 16mA$			0.55 0.55	V	A_0 through A_{12} All other outputs
V_{CR}	Regulator voltage	$V_{CC} = 5V$		3.1 2.9		V	$T_A = 0°C$ $T_A = 70°C$
V_{IC}	Input clamp voltage	V_{CC} = min; $I_{IN} = -10mA$			−1.5	V	Crystal inputs X1 and X2 do not have internal clamp diodes
I_{IH}	High level input current	V_{CC} = max $V_{IH} = 0.6V$ $V_{IH} = 4.5V$			4.0 50	mA μA	X1 and X2 All other pins
I_{IL}	Low-level input current	V_{CC} = max; $V_{IL} = 0.4V$			−3 −0.2 −1.6 −0.4	mA	X1 and X2 $\overline{IV0}$-$\overline{IV7}$ I0-I15 $\overline{HALT}$ and $\overline{RESET}$
I_{OS}	Short circuit output current	V_{CC} = max; (Note: At any time, no more than one output should be connected to ground.)	−30		−140	mA	All output pins
I_{CC}	Supply current	V_{CC} = max			180 195	mA	$T_A = 70°C$ $T_A = 0°C$
I_{REG}	Regulator control	$V_{CC} = 5.0V$	−10		−25	mA	Max available base drive for series-pass transistor
I_{CR}	Regulator current	V_{CC} = max			200 230	mA	$T_A = 70°C$ $T_A = 0°C$

Notes:
1. Operating temperature ranges are guaranteed after thermal equilibrium has been reached.
2. All voltages measured with respect to ground terminal.

AC CHARACTERISTICS (Commercial Part) CONDITIONS: $4.75V \leq V_{CC} \leq 5.25V$; $0°C \leq T_A \leq 70°C$

LOADING: (See test circuits)

PARAMETER (Note 1)		LIMITS (INSTRUCTION CYCLE TIME = 200ns) Min	Typ	Max	LIMITS (INSTRUCTION CYCLE TIME > 200ns) Min	Typ	Max	UNITS	COMMENTS
T_{PC}	Processor cycle time	200			200			ns	
T_{CP}	X1 clock period	100			100			ns	
T_{CH}	X1 clock high time	50			50			ns	
T_{CL}	X1 clock low time	50			50			ns	
T_{MCL}	MCLK low delay	15		40	15		40	ns	
T_W	MCLK pulse width	40		60	$T_{4Q} - 10$		$T_{4Q} + 10$	ns	Note 2
T_{MODO}	Output driver turn on time MCLK falling edge	125		145	$T_{1Q} + T_{2Q} + 25$		$T_{1Q} + T_{2Q} + 45$	ns	Note 9

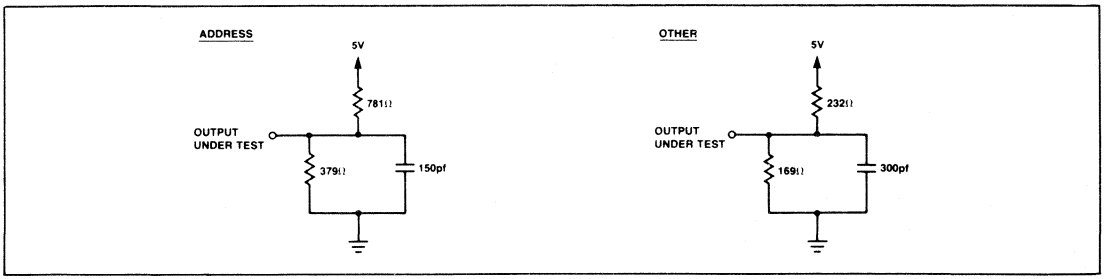
AC CHARACTERISTICS (Commercial Part) (Continued)

CONDITIONS: $4.75V \leq V_{CC} \leq 5.25V$; $0°C \leq T_A \leq 70°C$
LOADING: (See test circuits)

PARAMETER (Note 1)		LIMITS (INSTRUCTION CYCLE TIME = 200ns)			LIMITS (INSTRUCTION) CYCLE TIME > 200ns)			UNITS	COMMENTS
		Min	Typ	Max	Min	Typ	Max		
T_{DI}	Output driver turn-on time (SC/WC rising edge)	20			20			ns	Note 10
T_{DD}	Input data to output data	85		105	85		105	ns	
T_{MHS}	MCLK falling edge to $\overline{HALT}$ falling edge			30			$T_{1Q} - 20$	ns	Note 2
T_{MHH}	$\overline{HALT}$ hold time (MCLK falling edge)	65			$T_{1Q} + 15$			ns	Note 2
T_{ACC}	Program storage access time			60				ns	
T_{IO}	I/O port output enable time ($\overline{LR}/\overline{RB}$ to valide $\overline{IV}$ data input)			30				ns	
T_{MAS}	MCLK falling edge to address stable			140			$T_{1Q} + T_{2Q} + 40$	ns	Notes 2, 3 & 4
T_{IA}	Instruction to address			140			$T_{2Q} + 90$	ns	Notes 2, 3 & 5
T_{IVA}	Input data to address			85			85	ns	Notes 3 & 6
T_{MIS}	MCLK falling edge to instruction stable			30			$T_{1Q} - 20$	ns	Notes 2 & 10
T_{MIH}	Instruction hold time (MCLK falling edge)	55			$T_{1Q} + 5$			ns	Notes 2 & 8
T_{MWH}	MCLK falling edge to SC/WC rising edge	105		125	$T_{1Q} + T_{2Q} + 5$		$T_{1Q} + T_{2Q} + 25$	ns	Note 2
T_{MWL}	MCLK falling edge to SC/WC falling edge	5		15	5		15	ns	
T_{MIBS}	MCLK falling edge to $\overline{LB}/\overline{RB}$ (Input phase)	10		25	10		25	ns	
T_{IIBS}	Instruction to $\overline{LB}/\overline{RB}$ (Input phase)			25			25	ns	
T_{MOBS}	MCLK falling edge to $\overline{LB}/\overline{RB}$ (Output phase)	115		145	$T_{1Q} + T_{2Q} + 15$		$T_{1Q} + T_{2Q} + 45$	ns	Note 2
T_{MIDS}	MCLK falling edge to input data stable			55			$T_{1Q} + T_{2Q} - 45$	ns	Note 2
T_{MIDH}	Input data hold time (MCLK falling edge)	115			$T_{1Q} + T_{2Q} + 15$			ns	Note 2
T_{MODH}	Output data hold time (MCLK falling edge)	11			11			ns	
T_{MODS}	Output data stable (MCLK falling edge)	130		150	$T_{1Q} + T_{2Q} + 30$		$T_{1Q} + T_{2Q} + 50$	ns	Note 2

NOTES:
1. X1 and X2 inputs are driven by an external pulse generator with an amplitude of 1.5 volts; all timing parameters are measured at this voltage level.
2. Respectively, T_{1Q}, T_{2Q}, T_{3Q}, and T_{4Q} represent time intervals for the first, second, third, and fourth quarter cycles.
3. Capacitive loading for the address bus is 150 picofarads.
4. T_{MAS} is obtained by forcing a valid instruction and an I/O bus input to occur earlier than the specified minimum set up time.
5. T_{IA} is obtained by forcing a valid instruction input to occur earlier than the minimum set up time.
6. T_{IVA} is obtained by forcing a valid I/O bus input to meet the minium set up time.
7. T_{MIS} represents the setup time required by internal latches of the 8X305. In system applications, the instruction input may have to be valid before the worst-case set up time in order for the system to respond with a valid I/O bus input that meets the I/O bus input set up time (T_{IDS} and T_{MIDS}).
8. T_{MIH} represents the hold time required by internal latches of the 8X305. To generate proper $\overline{LB}/\overline{RB}$ signals, the instruction must be held valid until the address bus changes.
9. The minimum figure for these parameters represents the earliest time that I/O bus output drivers of the 8X305 will turn on.
10. This parameter represents the latest time that the output drivers of the input device should be turned off.

TEST CIRCUITS



ABSOLUTE MAXIMUM RATINGS

Storage Temperature (T_{STG}) ratings are from −65 to +150°C

PIN	DESCRIPTION	RATING	UNIT	PIN	DESCRIPTION	RATING	UNIT
V_{CC}	Supply voltage	+7.0	V	All other pins	Logic input pins	5.5	V
X1, X2	Crystal input voltage	2.0	V				

DC CHARACTERISTICS (Military Part)

4.5V ≤ V_{CC} ≤ 5.5V, −55°C ≤ T_C ≤ +125°C

PARAMETER		TEST CONDITIONS	LIMITS Min	LIMITS Typ	LIMITS Max	UNIT	COMMENTS
V_{CC}	Supply voltage		4.5	5.0	5.5	V	
V_{IH}	High level input voltage		0.6 2.0		2.0	V	X1 and X2 All other pins
V_{IL}	Low level input voltage				0.4 0.8	V	X1 and X2 All other pins
V_{OH}	High level output voltage	V_{CC} = min; I_{OH} = −3mA	2.4			V	
V_{OL}	Low level output voltage	V_{CC} = min; I_{OL} = 6mA V_{CC} = min; I_{OL} = 16mA			0.55 0.55	V	A_0 through A_{12} All other outputs
V_{CR}	Regulator voltage	V_{CC} = 5V		3.5 3.1 2.6		V	T_C = −55°C T_C = 0°C T_C = 125°C
V_{IC}	Input clamp voltage	V_{CC} = min; I_{IN} = −10mA			−1.5	V	Crystal inputs X1 and X2 do not have internal clamp diodes.
I_{IH}	High level input current	V_{CC} = max; V_{IH} = 0.6V V_{IH} = 4.5V			4.0 50	mA μA	X1 and X2 All other pins
I_{IL}	Low-level input current	V_{CC} = max; V_{IL} = 0.4V			−3 −0.3 −1.6 −0.4	mA	X1 and X2 $\overline{IV0}$-$\overline{IV7}$ I0-I15 $\overline{HALT}$ and $\overline{RESET}$
I_{OS}	Short circuit output current	V_{CC} = max; (Note: At any time, no more than one output should be connected to ground.)	−30		−140	mA	All output pins
I_{CC}	Supply current	V_{CC} = max			175 205	mA	T_C = 125°C T_C = −55°C
I_{REG}	Regulator control	V_{CC} = 5.0V	−10		−25	mA	Max available base drive for series-pass transistor
I_{CR}	Regulator current	V_{CC} = max			180 260	mA	T_C = 125°C T_C = −55°C

NOTES:
1. Operating temperature ranges are guaranteed after thermal equilibrium has been reached.
2. All voltages measured with respect to ground terminal.

AC CHARACTERISTICS (Military Part)

CONDITIONS: $4.5V \leq V_{CC} \leq 5.5V$; $-55°C \leq T_C \leq 125°C$
LOADING: (See test circuits)

PARAMETER (Note 1)		LIMITS (INSTRUCTION CYCLE TIME = 250ns)			LIMITS (INSTRUCTION) CYCLE TIME > 250ns)			UNITS	COMMENTS
		Min	Typ	Max	Min	Typ	Max		
T_{PC}	Processor cycle time	250			250			ns	
T_{CP}	X1 clock period	125			125			ns	
T_{CH}	X1 clock high time	62			62			ns	
T_{CL}	X1 clock low time	62			62			ns	
T_{MCL}	MCLK low delay	15		40	15		40	ns	
T_W	MCLK pulse width	47		72	$T_{4Q} - 15$		$T_{4Q} + 10$	ns	Note 2
T_{MODO}	Output driver turn-on time (MCLK falling edge)	145		175	$T_{1Q} + T_{2Q} + 20$		$T_{1Q} + T_{2Q} + 50$	ns	Note 9
T_{DI}	Output driver turn-on time (SC/WC rising edge)	20			20			ns	Note 10
T_{DD}	Input data to output data	80		115	80		115	ns	
T_{MHS}	MCLK falling edge to $\overline{HALT}$ falling edge			40			$T_{1Q} - 22$	ns	Note 2
T_{MHH}	$\overline{HALT}$ hold time (MCLK falling edge)	80			$T_{1Q} + 18$			ns	Note 2
T_{ACC}	Program storage access time			90				ns	
T_{IO}	I/O port output enable time ($\overline{LB}/\overline{RB}$ to valid $\overline{IV}$ data input)			40				ns	
T_{MAS}	MCLK falling edge to address stable			160			$T_{1Q} + T_{2Q} + 35$	ns	Notes 2, 3 & 4
T_{IA}	Instruction to address			160			$T_{2Q} + 98$	ns	Notes 2, 3 & 5
T_{IVA}	Input data to address			90			90	ns	Notes 3 & 6
T_{MIS}	MCLK falling edge to instruction stable			40			$T_{1Q} - 22$	ns	Notes 2 & 10
T_{MIH}	Instruction hold time (MCLK falling edge)	70			$T_{1Q} + 8$			ns	Notes 2 & 8
T_{MWH}	MCLK falling edge to SC/WC rising edge	127		154	$T_{1Q} + T_{2Q} + 2$		$T_{1Q} + T_{2Q} + 29$	ns	Note 2
T_{MWL}	MCLK falling edge to SC/WC falling edge	5		25	5		25	ns	
T_{MIBS}	MCLK falling edge to $\overline{LB}/\overline{RB}$ (Input phase)	10		35	10		35	ns	
T_{IIBS}	Instruction to $\overline{LB}/\overline{RB}$ (Input phase)			30			30	ns	
T_{MOBS}	MCLK falling edge to $\overline{LB}/\overline{RB}$ (Output phase)	140		170	$T_{1Q} + T_{2Q} + 15$		$T_{1Q} + T_{2Q} + 45$	ns	Note 2
T_{MIDS}	MCLK faling edge to input data stable			75			$T_{1Q} + T_{2Q} - 50$	ns	Note 2
T_{MIDH}	Input data hold time (MCLK falling edge)	140			$T_{1Q} + T_{2Q} + 15$			ns	Note 2
T_{MODH}	Output data hold time (MCLK falling edge)	11			11			ns	
T_{MODS}	Output data stable (MCLK falling edge)	150		180	$T_{1Q} + T_{2Q} + 25$		$T_{1Q} + T_{2Q} + 55$	ns	Note 2

NOTES:
1. X1 and X2 inputs are driven by an external pulse generator with an amplitude of 1.5 volts; all timing parameters are measured at this voltage level.
2. Respectively, T_{1Q}, T_{2Q}, T_{3Q}, and T_{4Q} represent time intervals for the first, second, third, and fourth quarter cycles.
3. Capacitive loading for the address bus is 150 picofarads.
4. T_{MAS} is obtained by forcing a valid instruction and an I/O bus input to occur earlier than the specified minimum set up time.
5. T_{IA} is obtained by forcing a valid instruction input to occur earlier than the minimum set up time.
6. T_{IVA} is obtained by forcing a valid I/O bus input to meet the minium set up time.
7. T_{MIS} represents the setup time required by internal latches of the 8X305. In system applications, the instruction input may have to be valid before the worst-case set up time in order for the system to respond with a valid I/O bus input that meets the I/O bus input set up time (T_{IDS} and T_{MIDS}).
8. T_{MIH} represents the hold time required by internal latches of the 8X305. To generate proper $\overline{LB}/\overline{RB}$ signals, the instruction must be held valid until the address bus changes.
9. The minimum figure for these parameters represents the earliest time that I/O bus output drivers of the 8X305 will turn on.
10. This parameter represents the latest time that the output drivers of the input device should be turned off.

TIMING CONSIDERATIONS (Commercial Part)

As shown in the AC CHARACTERISTICS table for the commercial part, the minimum instruction cycle time is 200 nanoseconds; whereas, the maximum is determined by the on-chip oscillator frequency and can be any value the user chooses. With an instruction cycle time of 200 nanoseconds, the part can be characterized in terms of absolute values; these are shown in the first "LIMITS" column of the table. When the instruction cycle time is greater than 200 nanoseconds, certain parameters are cycle-time dependent; thus, these parameters are specified in terms of the four quarter cycles (T_{1Q}, T_{2Q}, T_{3Q}, and T_{4Q}) that make up one instruction cycle — see 8X305 TIMING DIAGRAM. As the time interval for each instruction cycle increases (becomes greater than 200 nanoseconds), the delay for all parameters that are cycle-time dependent is likewise increased. In some cases, these delays have a significant impact on timing relationships and other areas of systems design; subsequent paragraphs describe these timing parameters and reliable methods of calculation.

Timing parameters for the 8X305 are normally measured with reference to MCLK.

System determinants for the instruction cycle time are:

- Propagation delays within the 8X305
- Access time of Program Storage
- Enable time of the I/O port

Normally, the instruction cycle time is constrained by one or more of the following conditions:

Condition 1 — Instruction or MCLK to $\overline{LB}/\overline{RB}$ (input phase) plus I/O port access time (TIO) ≤ $\overline{IV}$ data set up time (Figure 5a).

Condition 2 — Program storage access time (TACC) plus instruction to $\overline{LB}/\overline{RB}$ (input phase) plus I/O port access time (TIO) plus $\overline{IV}$ data (input phase) to address ≤ instruction cycle time (Figure 5b).

Condition 3 — Program storage access time plus instruction to address ≤ instruction cycle time (Figure 5c).

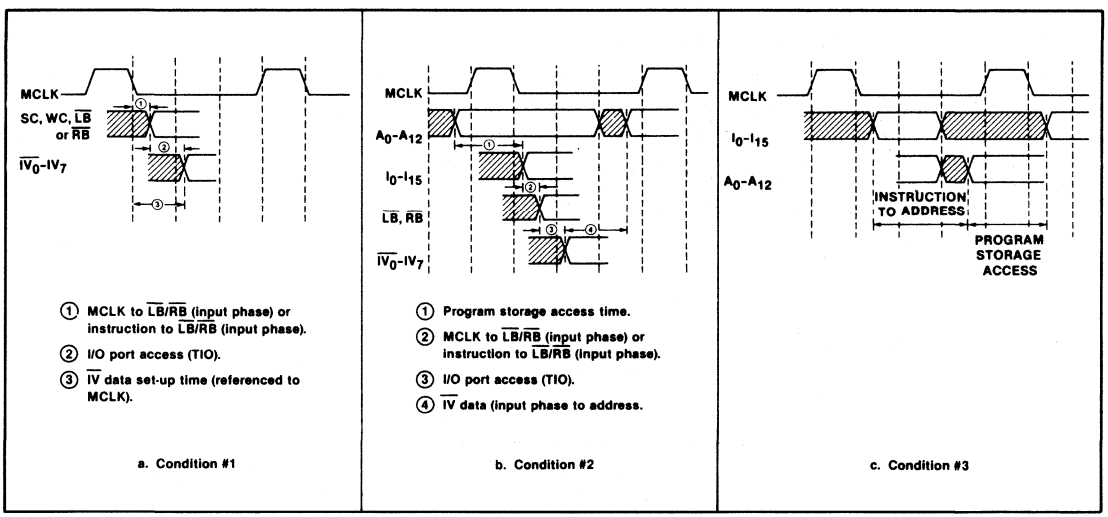
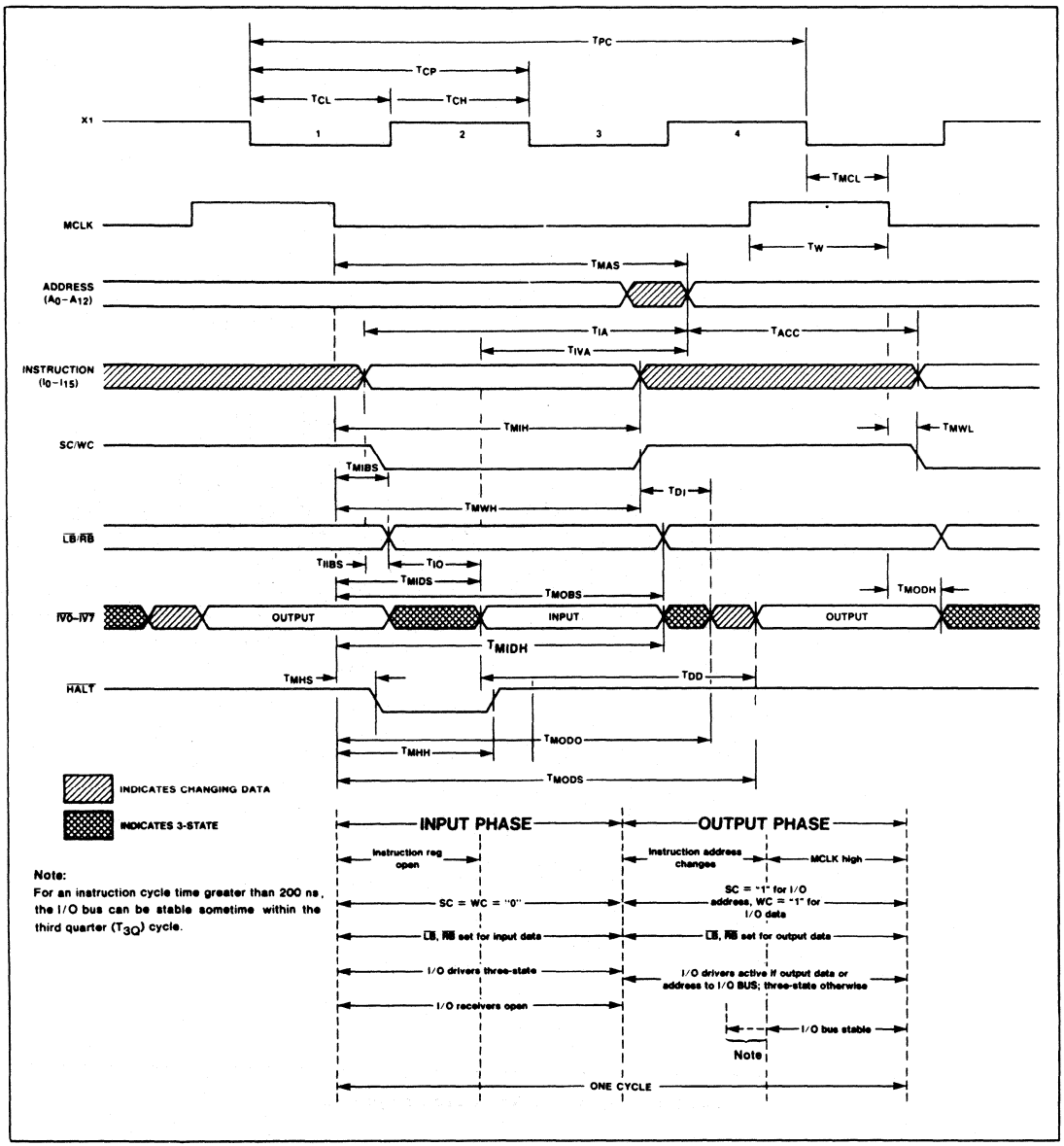


Figure 5. Constraints of 8X305 Instruction Cycle Time

8X305 TIMING DIAGRAM



Note:
For an instruction cycle time greater than 200 ns, the I/O bus can be stable sometime within the third quarter (T_{3Q}) cycle.

From condition #1 and with an instruction cycle time of 200ns, the I/O port access time (TIO) can be calculated as follows:

$$TMIBS + TIO \leq TMIDS$$
transposing, $TIO \leq TMIDS = TMIBS$
substituting, $TIO \leq 55ns - 25ns$
result, $TIO \leq 30ns$

Using 30ns for TIO, the constraint imposed by condition #1 can also be used to calculate the minimum cycle time:

$$TMIBS + TIO \leq TMIDS$$
thus, $25ns + 30ns \leq T_{1Q} + T_{2Q} - 45$
$25ns + 30ns \leq 1/2$ cycle $- 45$ therefore,

the worst-case instruction cycle time is 200ns. With subject parameters referenced to X1, the same calculations are valid:

$$TIBS + TIO + TIDS \leq 1/2 \text{ cycle}$$
thus, $45ns + 30ns + 25ns \leq 1/2$ cycle therefore,

the worst-case instruction cycle time is again 200ns. From condition #2 and with an instruction cycle time of 200ns, the program storage access time can be calculated:

$$TACC + TIIBS + TIO + TIVA \leq 200ns$$
transposing, $TACC \leq 200ns - TIIBS - TIO - TIVA$
substituting, $TACC \leq 200ns - 25ns - 30ns - 85ns$
thus, $TACC \leq 60ns$ hence, for an instruction

cycle time of 200ns, a program storage access time of 60ns is implied. The constraint imposed by condition #3 can be used to verify the maximum program storage access time:

$$TIA + TACC \leq \text{Instruction Cycle}$$
thus, $TACC \leq 200ns - 140ns$
and, $TACC \leq 60ns$, confirming that a program

storage access time of 60ns is satisfactory.

For an instruction cycle time of 200ns and a program storage access time of 60ns (Condition #2/Figure 5b), the instruction should be valid at the falling edge of MCLK. This relationship can be derived by the following equation:

$$200ns - TMAS - TACC$$
$$= 200ns - 140ns - 60ns$$
$$= 0ns$$

It is important to note that, during the input phase, the beginning of a valid $\overline{LB}/\overline{RB}$ signal is determined by either the instruction to $\overline{LB}/\overline{RB}$ delay (TIIBS) or the delay from the falling edge of MCLK to $\overline{LB}/\overline{RB}$ (TMIBS). Assuming the instruction is valid at the falling edge of MCLK and adding the instruction-to-$\overline{LB}/\overline{RB}$ delay (TIIBS = 25ns), the $\overline{LB}/\overline{RB}$ signal will be valid 25ns after the falling edge of MCLK. With a fast program storage memory and with a valid instruction before the falling edge of MCLK — the $\overline{LB}/\overline{RB}$ signal will, due to the TMIBS delay, still be valid 25ns after the falling edge of MCLK. Using a worst-case instruction cycle time of 200ns, the user cannot gain a speed advantage by selecting a memory with faster access time. Under the same conditions, a speed advantage cannot be obtained by using an I/O port with fast access time (TIO) because the address bus will be stable 55ns (TAS) after the beginning of the third quarter cycle — no matter how early the $\overline{IV}$ data input is valid.

CLOCK CONSIDERATIONS

The on-chip oscillator and timing-generation circuits of the 8X305 can be controlled by any one of the following methods:

Capacitor — if timing is not critical
Crystal — if precise timing is required
External Drive — if application requires that the 8X305 be driven from a system clock

Capacitor Timing. A non-polarized ceramic or mica capacitor with a working voltage equal to or greater than 25 volts is recommended. The lead lengths of capacitor should be approximately the same and as short as possible; also, the timing circuits should not be in close proximity to external sources of noise. For various capacitor (C_X) values, the cycle time can be approximated as:

C_X (in pF)	APPROXIMATE CYCLE TIME
100	300ns
200	500ns
500	1.1μs
1000	2.0μs

Crystal Timing. When a crystal is used, the on-chip oscillator operates at the resonant frequency (f_o) of the crystal; the series-resonant quartz crystal connects to the 8X305 via pins 10 (X1) and 11 (X2). The lead lengths of the crystal should be approximately equal and as short as possible; also, the timing circuits should not be in close proximity to external sources of noise. The crystal should be hermetically sealed (HC type can) and have the following electrical characteristics:

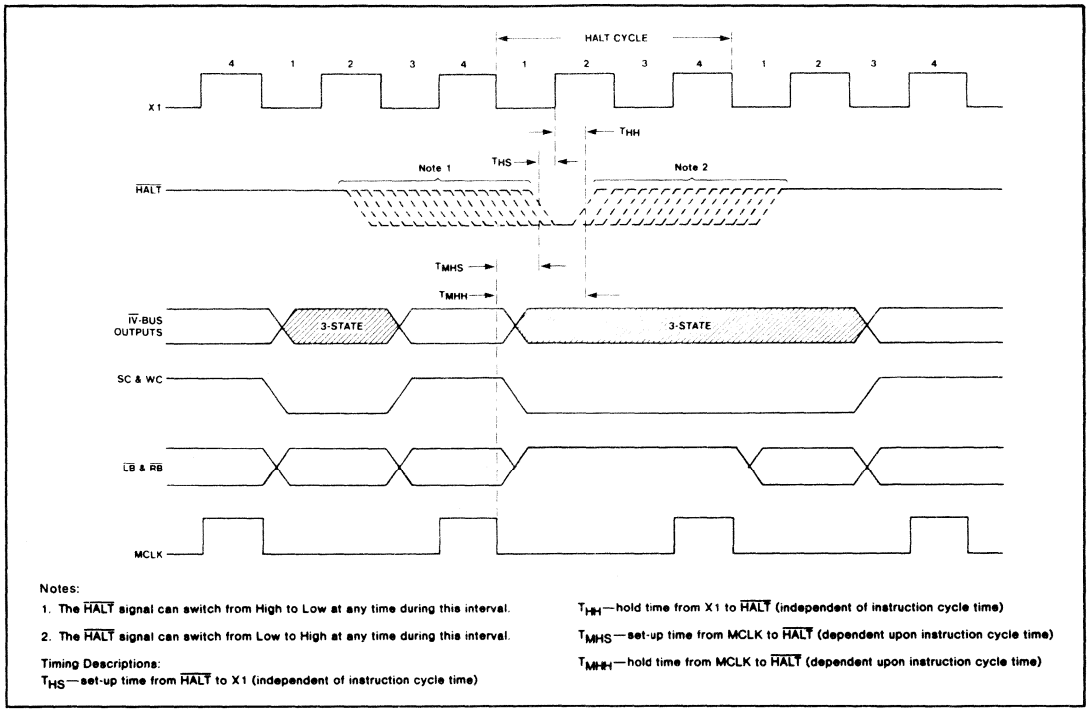
Type — Fundamental mode, series resonant
Impedance at Fundamental — 35 ohms maximum
Impedance at Harmonics and Spurs — 50 ohms minimum

The resonant frequency (f_o) of the crystal is related to the desired cycle time (T) by the equation: $f_o = 2/T$; thus, for a cycle time of 200 nanoseconds, $f_o = 10MHz$.

$\overline{HALT}$ Logic

The $\overline{HALT}$ signal is sampled via internal chip logic at the end of the first internal quarter of each instruction cycle. If, when sampled, the $\overline{HALT}$ signal is active-low, a halt is immediately executed and the current instruction cycle is terminated; however, the halt cycle does not inhibit MCLK nor does it affect any internal registers of the 8X305. As long as the $\overline{HALT}$ line is active-low, the SC and WC lines are low (inactive), the Left Bank ($\overline{LB}$)/Right Bank ($\overline{RB}$) signals are high (inactive), and the $\overline{IV}$ bus remains in the three-state mode of operation. Normal operation resumes at the next cycle in which $\overline{HALT}$ is high when sampled — see $\overline{HALT}$ TIMING DIAGRAM.

HALT TIMING DIAGRAM



Notes:

1. The $\overline{HALT}$ signal can switch from High to Low at any time during this interval.
2. The $\overline{HALT}$ signal can switch from Low to High at any time during this interval.

Timing Descriptions:

T_{HS}—set-up time from $\overline{HALT}$ to X1 (independent of instruction cycle time)

T_{HH}—hold time from X1 to $\overline{HALT}$ (independent of instruction cycle time)

T_{MHS}—set-up time from MCLK to $\overline{HALT}$ (dependent upon instruction cycle time)

T_{MHH}—hold time from MCLK to $\overline{HALT}$ (dependent upon instruction cycle time)

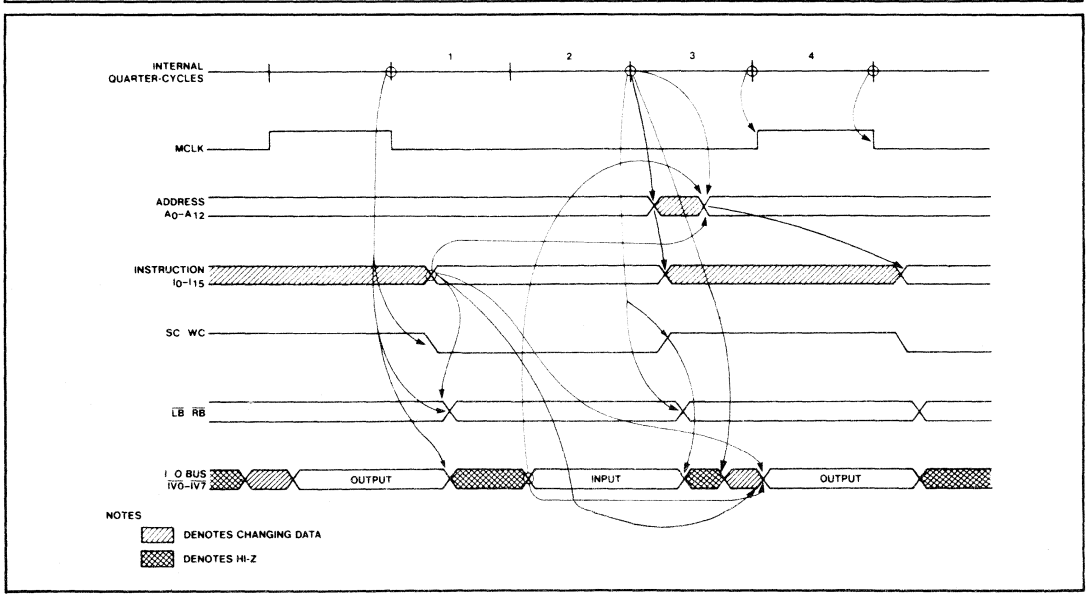


Figure 7. Timing Relationships of 8X305 I/O Signals

Using an External Clock. The 8X305 can be synchronized with an external clock by simply connecting appropriate drive circuits to the X1/X2 inputs. Figure 8 shows how the on-chip oscillator can be driven from the complementary outputs of a pulse generator. In applications where the MicroController must be driven from a master clock, the X1/X2 lines can be interfaced to TTL logic as shown in Figure 9.

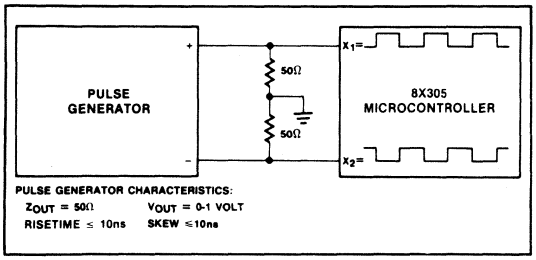


Figure 8. Clocking with a Pulse Generator

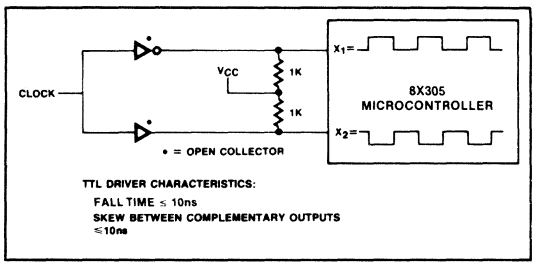


Figure 9. Clocking with TTL

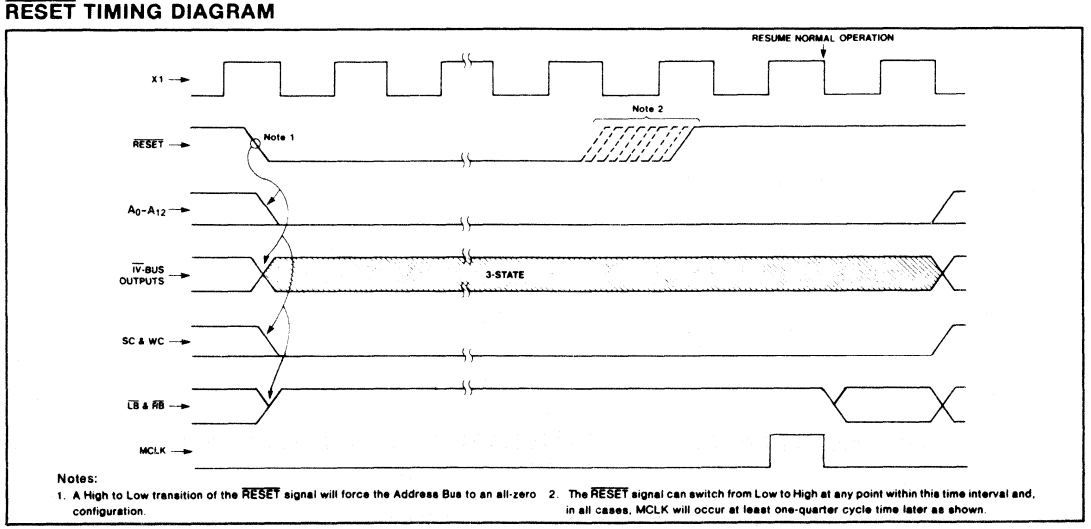
RESET Logic

RESET (pin 43) can be driven from a high (inactive) state to a low (active) state at any time with respect to the system clock, that is, the reset function is asynchronous. To ensure proper operation, RESET must be held low (active) for one full instruction time. When the line is driven from a high state to an active-low state, several events occur — the precise instant of occurrence is basically a function of the propagation delay for that particular event. As shown in the RESET TIMING DIAGRAM, these events are:

- The Program Counter and Address Register are set to address zero and remain in that state as long as the RESET line is low. Other than PC and AR, RESET does not affect other internal registers.
- The input/output (IV) bus goes three-state and remains in that condition as long as the RESET line is low.
- The Select Command and Write Command signals are driven low and remain low as long as the RESET line is low.
- The Left Bank/Right Bank (LB/RB) signals are forced high asynchronously for the period in which the RESET line is low.

During the time RESET is active-low, MCLK is inhibited; moreover, if the RESET line is driven low during the last two quarter cycles, MCLK may be shortened for that particular machine cycle. When RESET line is driven high (inactive)—one quarter to one full instruction cycle later, MCLK appears just before normal operation is resumed. The RESET/MCLK relationship is clearly shown by "B" in the timing diagram. As long as the RESET line is active-low, the HALT signal (described next) is not sampled by internal logic of the 8X305.

RESET TIMING DIAGRAM



INTERRUPT CONTROL COPROCESSOR

Originally published by Signetics January 1984

FEATURES

- **Three prioritized interrupts**
- **Subroutine handling capabilities**
- **4-level LIFO stack for return address storage**
- **Interrupt masking by software and hardware**
- **Stack full flag**
- **Directly compatible with 8X305 MicroController**
- **Bipolar ISL (Integrated Schottky Logic) and low-power Schottky technology**
- **Single +5 volt power supply**
- **0.6 inch, 40-pin DIP**

PRODUCT DESCRIPTION

The Signetics 8X310 Interrupt Control Coprocessor (ICC) supports the 8X305 MicroController in systems that are interrupt driven and those that require subroutine handling capabilities.

As shown in Figure 1, the ICC provides three prioritized interrupt request lines, INT 0 (highest priority), INT 1 and INT 2. A low-to-high transition applied to any of these input lines latches in an interrupt request which may be serviced when sampled by the ICC once each instruction cycle of the MicroController. When an interrupt request is serviced, the ICC forces the MicroController to jump to one of three fixed locations in program memory; instruction addresses 4, 5, and 6 correspond to INT 0, INT 1 and INT 2. At each of these addresses, the user programs a JMP instruction to another address where the user's interrupt service routine begins.

During interrupt servicing, the ICC also stores the proper return address into a four deep, Last-In-First-Out (LIFO) stack. At the conclusion of the interrupt service routine, the user program instructs the ICC to return to the main program at the location previously stored in the stack. The return operation is implemented by coding a special RETURN instruction which is decoded directly off the instruction bus by the ICC. There are five such special instructions relating to interrupt and subroutine handling functions performed by the ICC. These instruction codes are all treated as non-operational instructions (NOPs) by the MicroController.

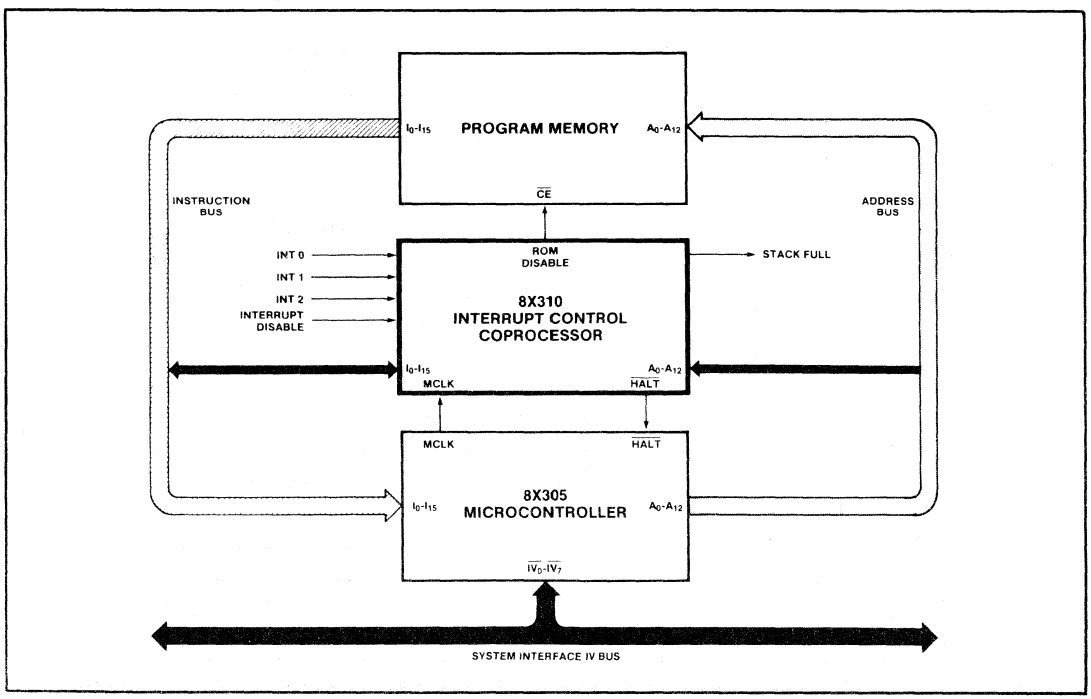


Figure 1. Typical System Connections Using ICC

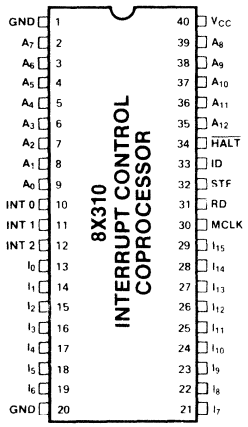
An internal one-bit mask is used to inhibit interrupt servicing. Whenever the mask is set, the ICC does not respond to any pending interrupt requests; however, any requests remain latched for future servicing. The mask can be set and cleared either by the user program or automatically during certain ICC functions. The special instructions SET MASK and CLEAR MASK are provided for user control. The Interrupt Disable input also inhibits interrupt request servicing.

The ICC provides a facility for implementing subroutines in the user program. A special PUSH instruction directs the ICC to store the return address into the stack in a manner similar to interrupt servicing. The jump to the subroutine, however, is performed by the user program. Subroutines may be nested (called from within other subroutines) depending on remaining vacancies in the four deep stack.

In general, the ICC adds some useful and very flexible facilities to the 8X305-based system. It offers both hardware and software capabilities that can improve efficiency and decrease program size. These features, from both a chip and system aspect, are described in subsequent paragraphs.

8X310 PACKAGE AND PIN DESIGNATIONS

N, I PACKAGE



TOP VIEW

ORDER NUMBERS

N8X310N, N8X310I
S8X310I/883B, S8X310I/883C

Pin No.	Identifier	Function
1, 20	GND	Ground. (Note: The printed circuit board should not use the ICC as a bridge for external ground.)
2-9, 35-39	A_7-A_0, A_{12}-A_8	Program address input lines from MicroController. Active high. A_0 is MSB.
10-12	INT 0-INT 2	Interrupt request input pins. INT 0 has the highest priority and INT 2 the lowest — Edge-triggered on a low-to-high transition.
13-19, 21-29	I_0-I_6, I_7-I_{15}	Bidirectional instruction bus; I_0 is MSB. When acting as an input, the ICC decodes the instruction flow (binary pattern on I_0-I_{15}) between program storage and the MicroController. During an interrupt or return cycle, the ICC outputs a JMP instruction to the MicroController via these lines —refer to FUNCTIONAL OPERATION of ICC.
30	MCLK	Master CLocK — active high input from 8X305 MicroController used for a timing reference and system synchronization.
31	RD	ROM (or PROM) Disable — active high output used to disable normal program storage so that the ICC can force an instruction to the MicroController.
32	STF	STack Full — active high output. When the LIFO stack is full, STF goes high and remains high until at least one register in the 4-level stack is empty.
33	ID	Interrupt Disable — active high. When this input pin is driven high, servicing of all interrupt requests is suspended.
34	$\overline{\text{HALT}}$	Active low output. Suspends all processing operations of the MicroController during period when the source of instruction data is changing between the ICC and program storage.
40	V_{CC}	+5 volt power supply.

FUNCTIONAL OPERATION

Basic Functions

The ICC performs the three general functions indicated below.

Function 1: Provides a means for the 8X305 MicroController to respond to interrupt requests by diverting the program flow of the 8X305 MicroController to the proper interrupt service routine or, in the case of a subroutine, the ICC stores the return address in the 4-level LIFO stack (Figure 2).

Function 2: Returns the user to the proper point in the main program for both interrupt and subroutine activities.

Function 3: Provides both automatic and programmed masking capabilities.

Interrupt Requests and Priority Considerations

An interrupt is requested when any one of the ICC input pins INT 0, INT 1, or INT 2 undergoes a low-to-high transition; this request is temporarily stored in an internal edge-triggered latch that corresponds to the affected interrupt input. The interrupt request latches are part of the Priority and Mask Logic shown in Figure 2. Unless masked or otherwise disabled, the ICC samples these latches once each instruction cycle. Any or all of the latches may be set when sampled by the ICC; however, only the interrupt of highest priority will be serviced — the remaining interrupts will be held in queue. Thus, if INT 0, INT 1 and INT 2 simultaneously compete for service, INT 0 is the first to be serviced followed, in order, by INT 1 and INT 2; likewise, if INT 1 and INT 2 compete for service, INT 1, being of higher priority will be serviced first. The CLEAR INTERRUPT instruction resets all interrupt request latches without affecting an interrupt service routine that is already in progress.

The highest priority interrupt request will be serviced when sampled by the ICC provided interrupts in general are not inhibited and a previous interrupt of equal or higher priority is not currently being serviced. The general masking of interrupts is discussed later. To determine priorities, the ICC keeps track of any interrupt that is serviced until the corresponding service routine returns. A subsequent interrupt request may interrupt a service routine in progress only if it is of a higher priority than that of the current interrupt being serviced. If, for example, INT 1 is requested and serviced, then before its service routine finishes, a request on INT 0 can be serviced as a second level interruption. However, a request on INT 2 or a second request on INT 1 must wait until the original INT 1 service routine returns. The interrupt service routine that was interrupted will resume execution at the point of interruption when the higher priority service routine returns (i.e. in the same manner as when returning to the main program).

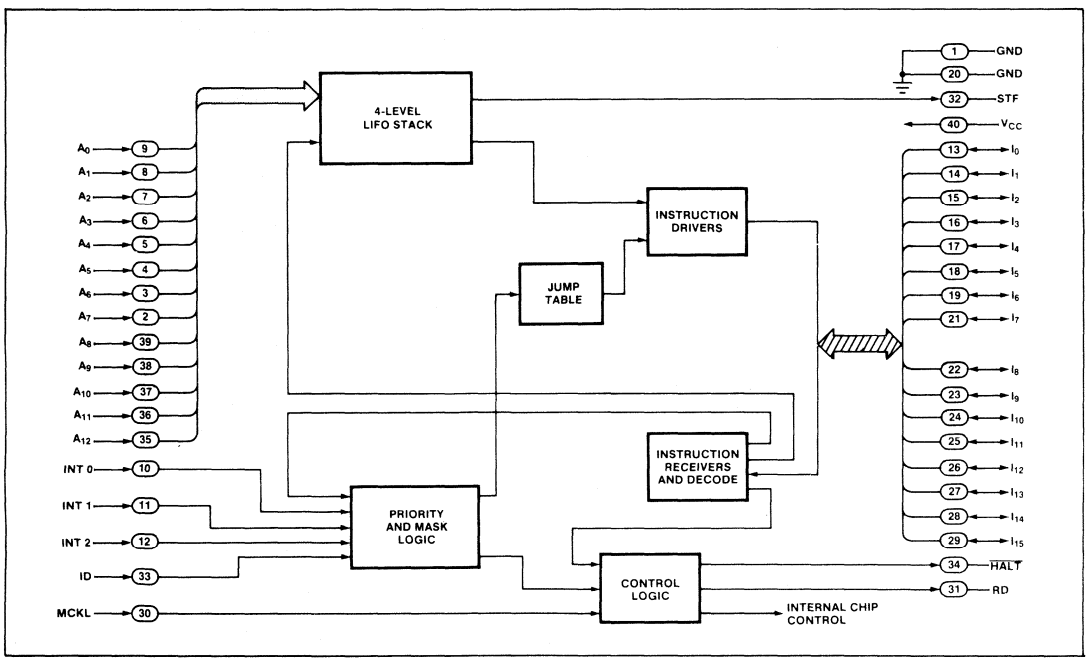


Figure 2. 8X310 Interrupt/Control Coprocessor — Functional Block Diagram

Interrupt Servicing

Interrupts are sampled only at the conclusion of an instruction cycle while the next instruction is being fetched from Program Memory.

When an interrupt request is serviced, the following general steps are performed:

- Address of the instruction that would normally be executed next is pushed into the 4-level LIFO Stack (Figure 2) for subsequent return to the main program.
- The ICC disables program storage and forces a JMP instruction onto the Instruction bus of the 8X305 MicroController. (Note: Because of timing considerations, the HALT signal is driven low to suspend operation of the MicroController for one instruction cycle; this permits the source of instruction data to change from program storage to the ICC without conflict.) The JMP instruction from the ICC transfers the MicroController to one of the three fixed program locations shown below. In each of these addresses, the user will normally store a JMP instruction to the interrupt service routine for that particular interrupt. Details of these operations are described later.

INT 0 ..... Address 4
INT 1 ..... Address 5
INT 2 ..... Address 6

Return from Interrupt Service Routine

Upon completion of the interrupt service routine, the user codes the special RETURN instruction. When executed, the ICC performs the following steps

- The return address is popped from the LIFO stack.
- The ICC disables program storage and forces a JMP instruction onto the MicroController instruction bus with the return address from the stack. (The HALT signal is driven low for one instruction cycle.)
- The JMP instruction from the ICC transfers the MicroController to the instruction that was about to execute at the time the interrupt was taken

A typical structure for a user program which handles interrupts is shown in the following example:

Address		Instruction	Comment
0		(any)	First instruction executed after system reset
•		•	
•		•	
3		JMP MAIN	Jump around interrupt vector locations.
4		JMP SERV0	Service INT 0 interrupt.
5		JMP SERV1	Service INT 1 interrupt.
6		JMP SERV2	Service INT 2 interrupt.
7	MAIN	(any)	Continue main program.
•		•	
•		•	
	SERV0	(any)	Begin INT 0 service routine.
•		•	
•		•	
		MAIN R6,R6	ICC RETURN instruction. End INT 0 service routine (resume main program execution).
	SERV1	(any)	Begin INT 1 service routine.
•		•	
•		•	
		MOVE R6,R6	RETURN from INT 1 service routine.
	SERV2	(any)	Begin INT 2 service routine.
•		•	
•		•	
		MOVE R6,R6	RETURN from INT 2 service routine.

Instruction Set

The five instructions shown in Table 1 allow the user to efficiently manage both the interrupt and subroutine capabilities of the ICC in an 8X310/8X305-based system. When an ICC instruction appears in the program, it is interpreted as a NOP by the 8X305 but is captured and decoded by the ICC to perform the desired function. The capture-and-decode functions of the chip are automatic. Assembly and object codes for each ICC instruction are shown in Table 1.

Table 1. INSTRUCTION SET FOR ICC

Instruction	Instruction Codes		8X305 Operation	Description of ICC Operation
	Assembler	Binary		
		I_0 = MSB I_{15} = LSB		
SET MASK	MOVE R5,R5	000 00101 000 00101	NOP	When executed, sets interrupt mask, thus inhibiting all interrupt servicing.
CLEAR MASK	MOVE R4,R4	000 00100 000 00100	NOP	When executed, clears interrupt mask for all interrupts
RETURN	MOVE R6,R6	000 00110 000 00110	NOP	When executed returns program to address at top of LIFO stack.
PUSH	MOVE R3,R3	000 00011 000 00011	NOP	Pushes "address + 2" onto stack if PUSH is programmed on odd address in program memory and "address + 1" if PUSH is programmed on even address.
CLEAR INTERRUPT	MOVE R2,R2	000 00010 000 00010	NOP	Clears all interrupt requests; an interrupt service routine that is in progress is unaffected.

Interrupt Masking Operations

Certain operations performed by the ICC and also some system considerations require that program execution not be interrupted for a specified interval of time. The servicing of interrupts by the ICC can be inhibited in a number of ways. Any time interrupts are inhibited, the ICC ignores any latched interrupt requests. However, the interrupt request latches are not cleared so that any previously pending requests remain latched. Also, during an interval when interrupt servicing is inhibited, any new interrupt signals received will get latched. As soon as interrupt servicing is enabled, any latched requests can be serviced on a priority basis.

The primary means of inhibiting interrupt servicing is the internal one-bit mask (latch). This mask can be set (to inhibit interrupts) or cleared under control of the user program using the special ICC instructions SET MASK and CLEAR MASK — See Table 1. With these instructions, segments of the user program can be isolated so as to proceed without interruptions. Frequently, uninterruptable segments are needed at the very beginning of the user program (initialization routine) and at the beginning of, or throughout an interrupt service routine. To facilitate this, the ICC automatically sets the mask whenever the MicroController executes address zero (typically resulting from a system reset) and whenever the ICC services an interrupt. The ICC also automatically clears the mask after performing a RETURN operation from an interrupt service routine; a RETURN from a subroutine does not affect the status of the interrupt mask.

The Interrupt Disable (ID) input pin may also be used to inhibit interrupt servicing. Interrupt servicing remains disabled as long as a high level is applied to the input. The ID input has no effect, however, on the status of the internal interrupt mask.

To ensure proper program flow, the ICC suspends interrupt servicing momentarily during certain situations. During the cycle in which the MicroController encounters an XEC (Execute) instruction an interrupt will not be serviced. This is because the XEC causes the MicroController to issue an address of an instruction to be executed out of the sequence of normal program flow. This would not be a valid address.

Interrupts are also suspended during execution of a PUSH or RETURN instruction and the instruction immediately following. This ensures proper operation of the LIFO stack. In addition, no interrupts are latched or serviced and no special ICC instructions are decoded at address zero which resets the ICC.

Subroutine Calling

The ICC provides for subroutine calling by storing the proper return address into the LIFO stack under control of the user program. Two instructions are required to implement a subroutine call — a PUSH instruction executed by the ICC and a JMP to the subroutine executed by the MicroController. The PUSH instruction is normally programmed at an odd-numbered address in program memory immediately followed by the JMP. When the PUSH instruction is executed, the ICC finds the address of the next instruction (JMP to subroutine) on the MicroController's address bus. internally changes the least-significant bit to one (effectively adds one to the address) and stores this into the stack. Program execution proceeds normally and the MicroController makes the jump to the beginning of the subroutine. The subroutine may be located at any convenient place in program memory.

Upon completion of the subroutine, the user codes the RETURN instruction in the same manner as for an interrupt service routine. At that point, the ICC forces the MicroController to resume execution of the main program at the instruction immediately following the JMP-to-subroutine instruction.

The code for a typical subroutine call-and-return is shown in the following example.

Address	Instruction	Comment
X (any odd-numbered address)	MOVE R3,R3	PUSH instruction initiates subroutine call by causing ICC to push the address X+2 onto the stack. (The PUSH instruction is interpreted as a NOP by the MicroController.)
X+1 (even)	JMP SUBR	The MicroController JMPs to the beginning of the subroutine.
X+2 (odd)	(any instruction)	Main program execution resumes here after RETURN from subroutine.
•	•	
•	•	
•	•	
SUBR (any address)	(any instruction)	Execution of subroutine starts here.
•	•	
•	•	
•	•	
(any address)	MOVE R6,R6	RETURN instruction causes ICC to transfer program back to X+2.

Stack Operation

The LIFO stack holds up to four 13-bit program addresses which allows the ICC to return from a subroutine or interrupt service routine. When all four stack locations are filled, the STack Full (STF) output pin is driven high and remains high until a RETURN (or reset) operation occurs. If an additional interrupt is serviced or subroutine called while the stack is full, the stack will overflow and the oldest return address will be overwritten and lost. That is, the stack retains the four most recent entries. After an overflow, the status of the STF output is not valid (until a reset operation occurs).

To prevent an interrupt from overflowing the stack, the user can connect the STF output directly to the Interrupt Disable (ID) input of the ICC. Then, even if the internal mask and priorities permit interrupt servicing, the interrupt request must still wait for the most recent service routine or subroutine to return.

Because subroutine calling is controlled explicitly by the user software, the user can always ensure that subroutine nesting alone could not overflow the stack. However, care must be taken whenever calling a subroutine from within an interrupt service routine since the number of remaining stack locations may vary at the time the interrupt is taken. If, for example, three stack locations are already filled (STF is low) at the time an interrupt is serviced, then a subroutine call executed within the interrupt service routine would cause the stack to overflow and the earliest return address to be lost.

As mentioned earlier, whenever a RETURN operation is performed from an interrupt service routine, the internal interrupt mask is automatically cleared. A RETURN from a

subroutine, however, does not affect the status of the mask. To accomplish this, a flag bit is added to each of the four stack locations which records whether each address pushed into the stack is caused by an interrupt or a subroutine call. This flag is read during a RETURN operation to determine whether or not the interrupt mask is cleared. This allows interrupt servicing and subroutine calls to be intermixed in any order.

Initialization

The ICC decodes address zero as a reset command to perform certain initialization functions. (Zero is the first address generated after the MicroController is reset.) Specifically, the Instruction-bus drivers are placed in a high-impedance state, HALT output is set high, RD output is set low, and all interrupt request latches are cleared. The interrupt mask is set so that any initialization routine by the user will not be interrupted until a CLEAR MASK instruction (MOVE R4,R4) is executed. The LIFO stack is reset to an empty state and the STF output is set low.

SYSTEM TIMING RELATIONSHIPS

Interrupt Servicing

Interrupt servicing begins at the end of a MicroController instruction cycle when the 8X305's MCLK signal goes from low-to-high. Starting from this point, processing of the interrupt proceeds as follows:

From the rising edge of MCLK 1:

- Interrupt mask is set to inhibit other interrupts.
- HALT output is driven low to stop internal operation of the 8X305 MicroController for one instruction cycle; MCLK is unaffected. ROM Disable (RD) output is driven high to disable program memory.

From the falling edge of MCLK 1:

- Takes address of next instruction from address bus and pushes it onto the top of stack to be used as the return address to the main program.

From the rising edge of MCLK 2:

- The ICC forces a JMP onto the instruction bus to one of three fixed vector addresses:

INT 0	Address 4
INT 1	Address 5
INT 2	Address 6

- Releases HALT (high) which allows the MicroControiler to complete the JMP to the above specified vector location in program memory.

From the rising edge of MCLK 3:

- Instruction-bus drivers of the ICC are disabled.
- ROM Disable (RD) pin is cleared (low) enabling the program memory which resumes control of the Instruction bus.

Return Operation

When the interrupt service routine or subroutine is completed, the RETURN instruction initiates the following sequence of events:

From the rising edge of MCLK 1:

- Interrupts are temporarily inhibited through third MCLK cycle.
- HALT output is driven low to stop MicroController for one instruction cycle.
- RD output is set high to disable program memory.

From the rising edge of MCLK 2:

- HALT output is driven high (cleared).
- A JMP instruction to address stored at top of LIFO stack is forced onto the Instruction bus by the ICC. (The stack is popped.)

From the rising edge of MCLK 3:

- Instruction-bus drivers of the ICC are disabled.
- RD is cleared enabling the program memory.
- If returning from an interrupt service routine (condition recorded in extra stack bit) the interrupt mask is cleared; otherwise the mask remains unaffected.

Once the preceding return actions are completed, the 8X305 MicroController will resume execution of the instruction at the return address.

APPLICATION HINTS

- When programming an interrupt service routine or subroutine, certain system operations typically need to be considered. In many interrupt-driven systems, a handshake signal is required to acknowledge the servicing of an interrupt request. The acknowledge signal may be transmitted by the interrupt service routine using a standard I/O port from the 8X300 Family.
- If the user wants to allow a higher priority interrupt request to interrupt a service routine, then the CLEAR MASK instruction should be programmed (perhaps after completing any critical operations).
- For both service routines and subroutines, the user may need to save the contents of some or all of the working registers of the MicroController so that operation of the main program is not upset. Registers may be written out to a working storage RAM such as 8X350 near the beginning of the routine, and restored from RAM just before returning to the main program.
- Certain subroutine calling techniques may be used to increase the efficiency of the user program. As shown in the following examples, a subroutine can automatically be repeated two, three or four times, if desired, without programming a loop.

SUBROUTINE AUTOMATICALLY EXECUTES TWICE			
Address		**Instruction**	
X (even)	SUBR2	MOVE R3,R3	Push X+1 onto stack.
X+1 (odd)		(start of subroutine)	
•		•	
•		•	
		MOVE R6,R6	RETURN — First time jumps to X+1; second time jumps back to main program.
SUBROUTINE AUTOMATICALLY EXECUTES THREE TIMES			
X (odd)	SUBR3	MOVE R3,R3	Push X+2 onto stack.
X+1 (even)		MOVE R3,R3	Push X+2 onto stack.
X+2 (odd)		(start of subroutine)	
•		•	
•		•	
SUBROUTINE AUTOMATICALLY EXECUTES FOUR TIMES			
X (even)	SUBR4	MOVE R3,R3	Push X+1 onto stack.
X+1 (odd)		MOVE R3,R3	Push X+3 onto stack.
X+2 (even)		NOP	
X+3 (odd)		(start of subroutine)	
•		•	
•		•	

- In a manner similar to the MicroController multi-way branch technique, one of several subroutines can be selected according to an index value.

SUBROUTINE CALL SELECTED BY VALUE IN R1			
Address		**Instruction**	
X (odd)		MOVE R3,R3	Push X+2 onto stack.
X+1 (even)		XEC TABLE (R1)	Execute JMP at TABLE + (R1)
X+2 (odd)		(any)	Subroutine returns here.
•		•	
•		•	
(any)	TABLE	JMP SUB0	Call SUB0 if R1 = 0.
		JMP SUB1	Call SUB1 if R1 = 1.
		•	
		•	

DC ELECTRICAL CHARACTERISTICS

COMMERCIAL: V_{CC} = 5.0 V (±5%); 0° C ≤ T_A ≤ 70° C
MILITARY: V_{CC} = 5.0 V (±10%); T_A ≥ -55° C
T_C ≤ 125° C

ABSOLUTE MAXIMUM RATINGS

Parameter		Rating	Unit	Parameter		Rating	Unit
V_{CC}	Power supply voltage	+7	V DC	V_O	Off-state output voltage	+5.5	V DC
V_{IN}	Input voltage	+5.5	V DC	T_{STG}	Storage temperature range	−65 to +150	°C

Parameter		Test Conditions		Limits (Commercial) Min	Typ	Max	Limits (Military) Min	Typ	Max	Unit
V_{IH}	High Level Input Voltage			2.0			2.0			V
V_{IL}	Low Level Input Voltage					0.8			0.8	V
V_{OH}	High Level Output Voltage	V_{CC} = Min.; I_{OH} = −1.0 mA		2.4			2.4			V
V_{OL}	Low Level Output Voltage	V_{CC} = Min.	COMMERCIAL: I_{OL} = 8 mA			0.55				V
			MILITARY: I_{OL} = 4.25 mA						0.55	
V_{CL}	Input Clamp-Diode Voltage	V_{CC} = Min; I_{CL} = −10 mA				−1.5			−1.5	V
I_{IH}	High Level Input Current	V_{CC} = Max; V_{IH} = 2.7 V				100			100	μA
I_{IL}	Low Level Input Current	V_{CC} = Max; V_{IL} = 0.4 V				−550			−700	μA
I_{OS}	Short Circuit Output Current	V_{CC} = Max; V_O = 0 V		−15		−80	−15		−80	mA
I_{CC}	Supply Current	V_{CC} = Max; I_0-I_{15} = High-Z								mA
			T_A = 0° C[2]			200				
			T_A = 70° C			185				
			T_A = −55° C[2]						230	
			T_C = 125° C						170	

AC ELECTRICAL CHARACTERISTICS

LOADING: See TEST LOADING CIRCUITS

COMMERCIAL: V_{CC} = 5.0 V (±5%); 0° C ≤ T_A ≤ 70° C
MILITARY: V_{CC} = 5.0 V (±10%); T_A ≥ -55° C
T_C ≤ 125° C

Parameter		References From	To	Test Conditions	Limits (Commercial) Min	Typ	Max	Limits (Military) Min	Typ	Max	Unit
Pulse Widths:											
t_{WIH}	Interrupt High	↑INT_i	↓INT_i		30			30			ns
t_{WIL}	Interrupt Low	↓INT_i	↑INT_i		35			35			ns
t_{WMH}	MCLK High	↑MCLK	↓MCLK	For all Functions	40			47			ns
Propagation Delays:											
t_{PRH}	RD High	↑MCLK	↑RD	Interrupt or Return			70			75	ns
t_{PRL}	RD Low	↑MCLK	↓RD	Interrupt or Return			15			17	ns
t_{PHL}	HALT Low	↑MCLK	↓HALT	Interrupt or Return			70			87	ns
t_{PHH}	HALT High	↑MCLK	↑HALT	Interrupt or Return			65			75	ns
t_{PSH}	Stack Full High	↓MCLK	↑STF	Interrupt or Subroutine Call			105			105	ns
t_{PSL}	Stack Full Low	↓MCLK	↓STF	Return or Reset			110			115	ns

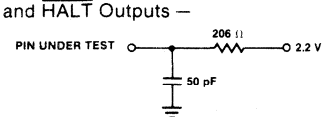
AC ELECTRICAL CHARACTERISTICS (CONTINUED)

Parameter		References From	References To	Test Conditions	Limits (Commercial) Min	Typ	Max	Limits (Military) Min	Typ	Max	Unit
Setup Times:											
t_{SIH}	Interrupt Input Setup[3]	↑INT_i	↑MCLK		35			35			ns
t_{SA}	Address Setup	A_0-A_{13}	↑MCLK	Interrupt, Subroutine Call, or Reset	0			0			ns
t_{SC}	Instruction Setup[5]	I_{0-15}	↓MCLK	All Commands	Note 5			Note 5			ns
t_{SD}	Interrupt Disable Setup[3]	ID	↑MCLK		30			30			ns
Hold and Reset Recovery Times:											
t_{HIL}	Interrupt Low Input Hold[3]	↑MCLK	↑INT_i		15			15			ns
t_{HA}	Address Hold	↓MCLK	A_{0-13}	Subroutine Call or Reset	75			90			ns
t_{HC}	Instruction Hold	↓MCLK	I_{0-15}	All Commands	55			55			ns
t_{HD}	Interrupt Disable Hold[3]	↑MCLK	ID		25			25			ns
t_{RI}	Interrupt Reset Recovery[4]	↓MCLK	↑INT_i	Reset or Cancel Command	70			70			ns
Output Enable/Disable Delays:											
t_{OEC}	Instruction Output Enable	↑MCLK	I_{0-15}	Interrupt or Return			70			87	ns
t_{ODC}	Instruction Output Disable	↑MCLK	I_{0-15}	Interrupt or Return			40			47	ns

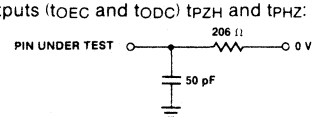
Notes:
1. All electrical characteristics are guaranteed after power is applied and thermal equilibrium has been reached.
2. The 200 and 230 milliampere values are worst case over the entire temperature range for the Commercial and Military parts, respectively.
3. Parameters t_{SIH}, t_{HIL}, T_{SD}, and t_{HD} are used only to determine whether an interrupt request will be serviced during the current or a subsequent instruction cycle. The INT_i and ID inputs are asynchronous and transitions on either input may safely occur at any time with respect to MCLK. A low-to-high transition on INT_i occurring after t_{SIH} and before t_{HIL} means only that it cannot be determined for sure whether or not the interrupt request will be honored during the current instruction cycle. Similarly, transitions on ID between t_{SD} and t_{HD} make it uncertain as to whether or not masking applies during the current instruction cycle.
4. When clearing interrupt requests (including a reset operation), any new low-to-high transitions appearing at the INT_i inputs that occur before t_{RI} risk being cleared and therefore ignored; however, any transition after t_{RI} is certain to be latched.
5. COMMERCIAL: t_{SC} (minimum) = 15 ns — t_{PRL} (actual).
   MILITARY: t_{SC} (minimum) = 17 ns — t_{PRL} (actual).
   (The required instruction enable time for the program memory depends on the sum of the t_{PRL} and t_{SC}.)

TEST SETUPS

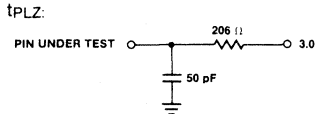
RD, STF, and $\overline{HALT}$ Outputs —



I_0-I_{15} Outputs (t_{OEC} and t_{ODC}) t_{PZH} and t_{PHZ}:



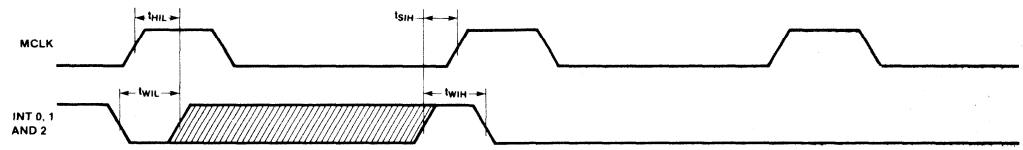
t_{PZL} and t_{PLZ}:



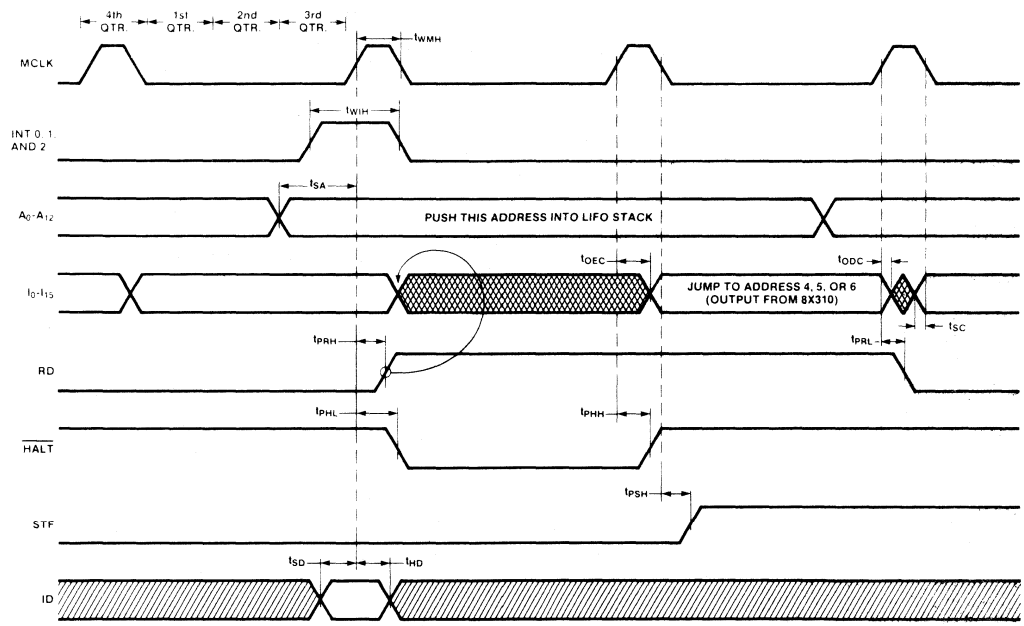
Notes:
1. V_{MEAS} = 1.5 V for all input signals and RD, STF, and HALT outputs.
2. For I_0-I_{15} (t_{OEC}): V_{MEAS} = 1.5 V
   For I_0-I_{15} (t_{ODC}): V_{MEAS} (t_{PLZ}) = V_{OL} + 0.5 V
   V_{MEAS} (t_{PHZ}) = V_{OH} − 0.5 V

TIMING DIAGRAMS

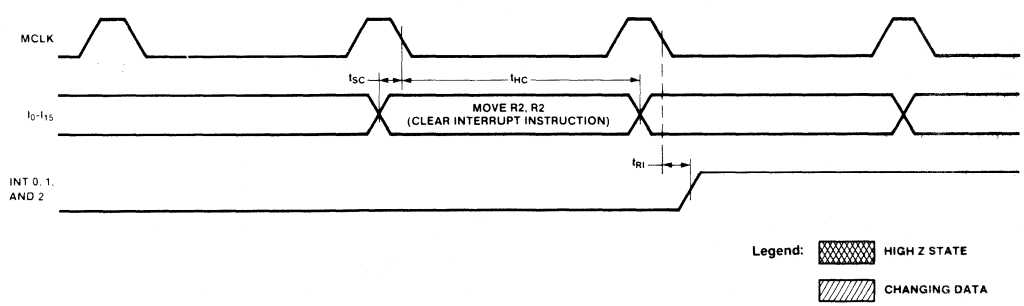
INTERRUPT REQUEST TIMING:



INTERRUPT SERVICE TIMING:

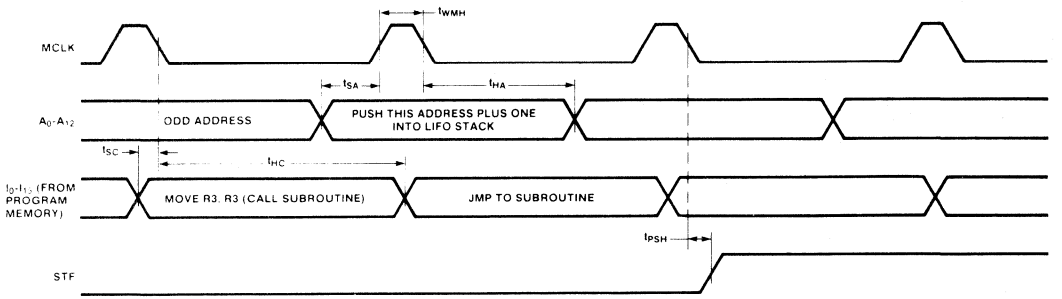


CLEAR INTERRUPT INSTRUCTION TIMING:

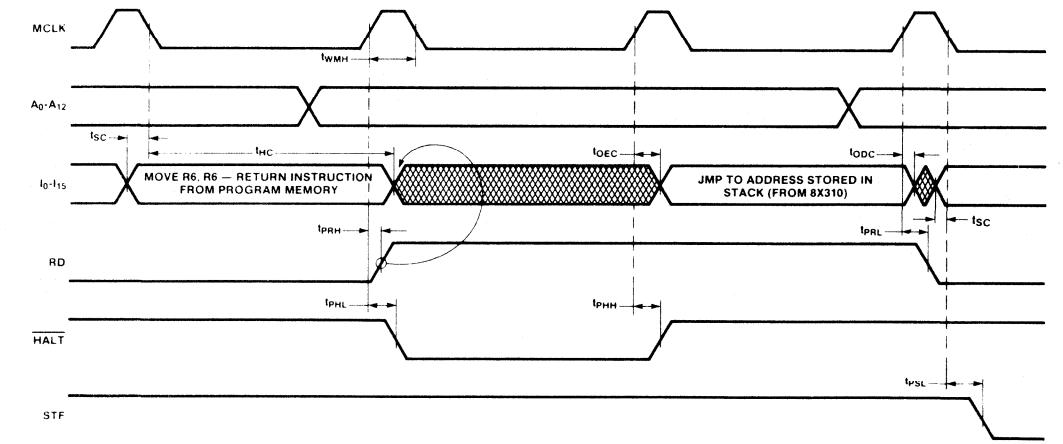


TIMING DIAGRAMS (Continued)

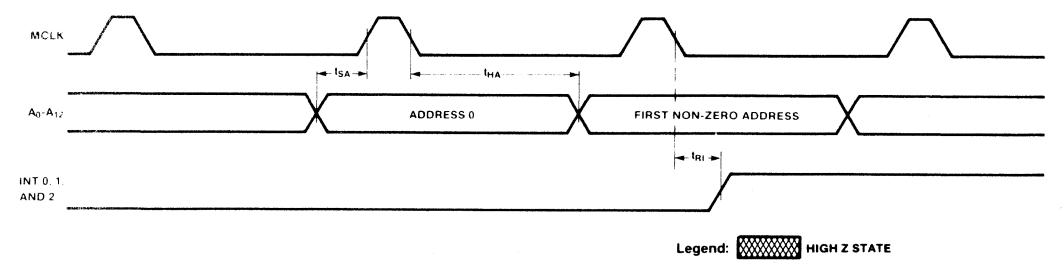
CALL SUBROUTINE TIMING:



RETURN TIMING (FROM INTERRUPT OR SUBROUTINE):



RESET TIMING:



BUS INTERFACE REGISTER ARRAY

Originally published by Signetics January 1984

FEATURES

- **16-byte/2-port interface**
- **8- or 16-bit primary port (Host) interface (User selectable)**
- **8-bit secondary port interface**
- **Two 8-bit flag registers (handshake control)**
- **DMA or programmed I/O operation**
- **Two three-state bidirectional ports**
- **Secondary port is bus compatible with 8X305**
- **Single 5V supply**
- **40-pin package**

ARCHITECTURAL OVERVIEW

The Signetics 8X320 Bus Interface Register Array (Figure 1) is a dual-port RAM memory designed for use between a host processor and a peripheral processor. Specifically, the register array provides a convenient and economical interface between the 8X305 (or 8X300) Microcontroller (secondary port) and User's Host System (primary port); the host can be almost any bus-oriented device—another processor, a minicomputer, or a mainframe computer. The host has 8-bit (byte) or 16-bit (word) access to the primary port; data can be read-from or written-into any memory location as determined by the primary-port address and control lines. The secondary port (8X305 bus) consists of eight input/output lines and four bus control lines. To implement the secondary-port interface, an 8-bit memory location is addressed during one machine cycle and, during another cycle, data is read or written under control of the secondary (8X305) processor. Both primary and secondary ports feature three-state outputs and both ports are bidirectional.

Besides the convenience and economy of a two-port memory, the array also provides simple handshake control via two 8-bit flag registers, logic to facilitate DMA transfers, and a write-protect feature for the primary port in both byte and word modes of operation.

BLOCK DIAGRAM

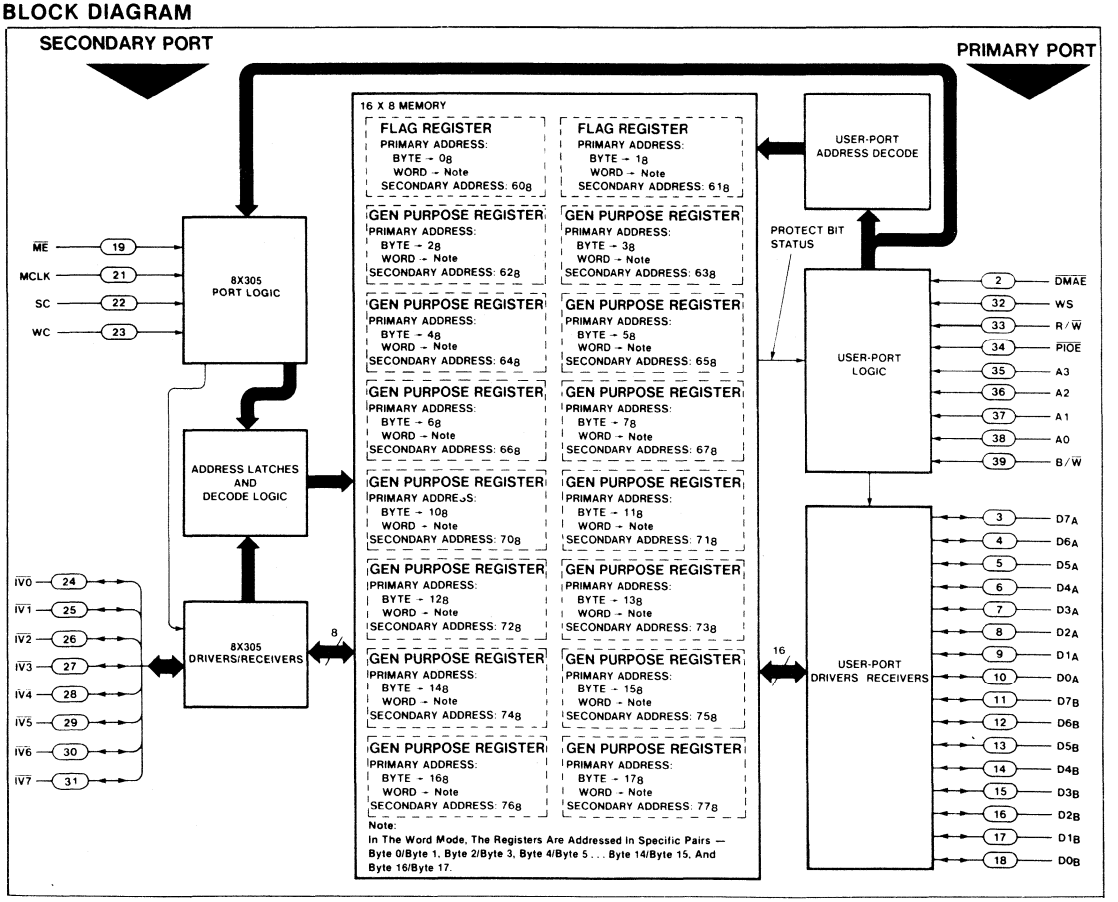
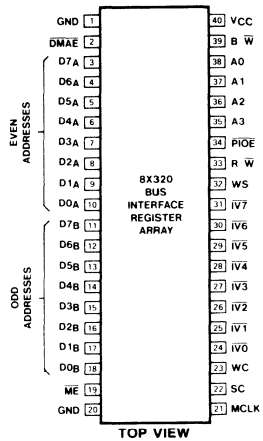


Figure 1. Block Diagram of 8X320 Bus Interface Register Array

N, I PACKAGE



TOP VIEW

ORDER NUMBERS

N8X320N, N8X320I

PIN NO.	PARAMETER		FUNCTION
1, 20	GND	Ground	Circuit ground.
2	$\overline{DMAE}$	Direct Memory Access Enable	Enables primary port to facilitate DMA transfers; does not affect secondary port.
3-18	$D0_A$-$D7_A$/ $D0_B$-$D7_B$	Primary Data Port	Sixteen 3-state lines used for data transfers to-and-from the primary data port; most significant bit is $D0_B$ and least significant bit is $D7_A$.
19	$\overline{ME}$	Master Enable	Enables secondary port when active low ($\overline{ME}$).
21	MCLK	Master Clock	When MCLK is high, and 8X320 is enabled ($\overline{ME}$ = Low), a register location may be either selected or written-into under control of SC and WC.
22	SC	Select Command	With SC high, WC low, MCLK high and $\overline{ME}$ low, data on $\overline{IV0}$ through $\overline{IV7}$ is interpreted as an address. If any one of the 16 register addresses (60_B-77_B) matches that on the I/O (IV) bus, that particular register is selected and remains selected until another address on the same bank (i.e. $\overline{ME}$ = low) is output on the I/O bus—at which time, the old register is deselected and a new register may or may not be selected.
23	WC	Write Command	With WC high, SC low, MCLK high, and $\overline{ME}$ low, the selected register stores contents of $\overline{IV0}$-$\overline{IV7}$ as data.
24-31	$\overline{IV0}$-$\overline{IV7}$	Secondary Data Port	Eight 3-state lines used to transfer data or I/O address to-and-from the secondary data port; most significant bit is $\overline{IV0}$ and least significant bit is $\overline{IV7}$.
32	WS	Write Strobe	When active high, data appearing at the primary port ($D0_A$-$D7_A$/$D0_B$-$D7_B$) is stored in the register array if the primary port is in the *write* mode.
33	R/$\overline{W}$	Read/Write Control	When this signal is high, primary port is in *read* mode; when signal is low, primary port is in *write* mode.
34	$\overline{PIOE}$	Programmed I/O Enable	When active low, primary port operates in programmed input/output mode with register to be read-from or written-into selected by A0-A3.
35-38	A0-A3	Primary Port Address Select	Selects register or register-pair that primary port is to read-from or write-into. Most significant bit is A3; least significant bit is A0.
39	B/$\overline{W}$	Byte/Word	When signal is high, the primary port operates in the byte (8-bit) mode; when signal is low, the primary port operates in the word (16-bit) mode.
40	V_{CC}	Power	+5 volts.

All barred symbols ($\overline{DMAE}$, etc.) denote signals that are asserted (or active) when low (logical 0); signals that are not barred are asserted in the high state (logical 1).

OPERATING CHARACTERISTICS

Memory Organization

Memory and address correlation for the 16-register array is shown in Figure 2. From the primary port, the sixteen 8-bit registers can be addressed in either (8-bit) or word (16-bit) format; in the word mode, the registers are addressed in pairs—$0_8/1_8$, $2_8/3_8$, $4_8/5_8$, . . . $14_8/15_8$, and $16_8/17_8$. From the secondary port, all registers are addressed in byte format—60_8 through 77_8. The memory consists of two 8-bit flag registers and fourteen 8-bit general-purpose registers. The flag registers facilitate information transfers between the two ports and, in addition, they protect certain registers from being written into from the primary port.

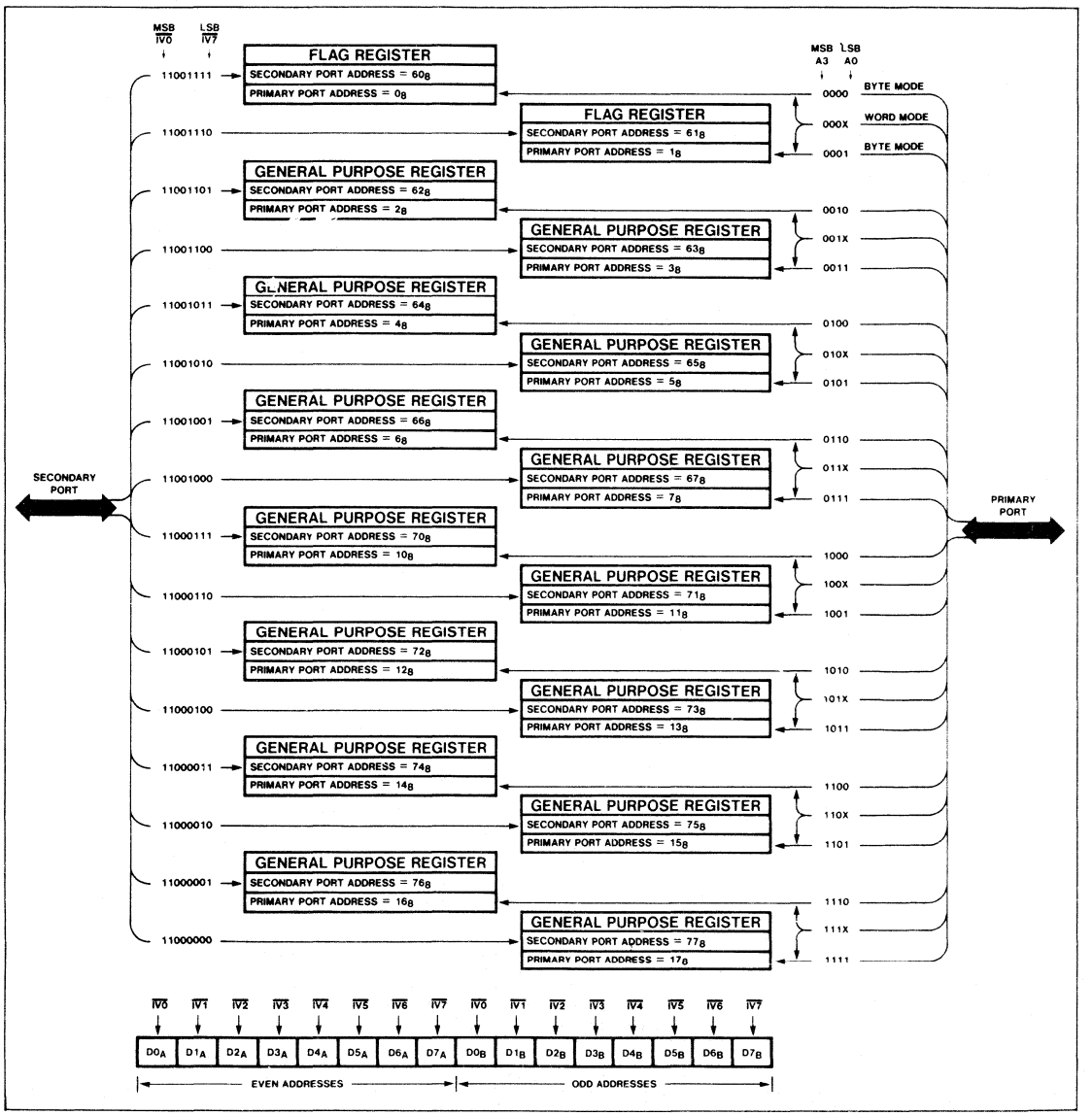


Figure 2. Memory and Address Organization for the 8X320

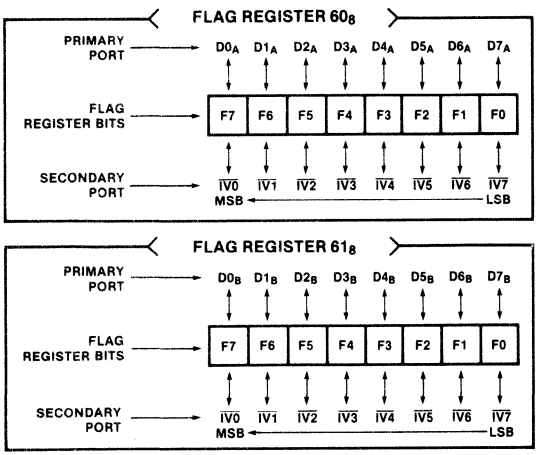
In either byte or word mode, the write-protect logic, implemented by bits F0 and F1 of register 60_8, inhibits the primary port from writing into addresses 16_8 and 17_8, respectively. Both write-protect bits (F0 and F1) can be read or written from the secondary port; the bits are read-only from the primary port.

As shown in Table 1, flag bits F2 through F7 of 60_8 and F0 through F7 of 61_8 are controlled by the fourteen general-purpose registers. When any one of these registers is written into by either port, the corresponding flag bit for that register is automatically set by internal logic of the 8x320. When information is read from any register, the corresponding flag bit must be reset by user software. Except for the write-protect bits, all other flag bits can be read or reset from the primary or the secondary port. Table 2 shows the relationship between bits of the flag registers and bits of the primary and secondary ports.

Table 1. CONTROL OF THE TWO FLAG REGISTERS

Flag Registers	60_8 (0_8)	F2	F3	F4	F5	F6	F7								
	61_8 (1_8)	F0	F1	F2	F3	F4	F5	F6	F7						
Octal Address of Controlling Byte	Primary	2	3	4	5	6	7	10	11	12	13	14	15	16	17
	Secondary	62	63	64	65	66	67	70	71	72	73	74	75	76	77

Table 2. RELATIONSHIP BETWEEN FLAG REGISTER BITS AND THOSE OF PRIMARY AND SECONDARY PORTS



FUNCTION AND CONTROL OF PRIMARY PORT

The primary port provides an 8-bit (byte) or 16-bit (word) interface between the 16-byte memory and the user's host system. If the host is an 8-bit system (or 16-bit system operating in Byte mode), the sixteen bidirectional I/O lines must be tied together ($D0_A$ to $D0_B$, $D1_A$ to $D1_B$, . . . and $D7_A$ to $D7_B$); when data is input or output on $D0_A$ through $D7_A$, the remaining eight lines ($D0_B$ through $D7_B$) are high-Z and vice-versa.

Other than the Byte/$\overline{\text{Word}}$ control line, specific operating characteristics of the primary port are controlled by two signals—$\overline{\text{PIOE}}$ (Programmed I/O Enable) and $\overline{\text{DMAE}}$ (Direct Memory Access Enable). When $\overline{\text{PIOE}}$ is active (low) and $\overline{\text{DMAE}}$ is inactive (high), the primary port operates in the programmed I/O mode—refer to Table 3; in this mode of operation, the register to be read-from or written-into is determined by four address lines (A0 through A3) and the Byte/$\overline{\text{Word}}$ control line—see Figure 2 and Table 4. In the DMA mode of operation, A1, A2, and A3 are not used; data is read-from or written-into preassigned registers: bytes 16_8 (76_8) and 17_8 (77_8) for the byte mode of operation and bytes 14_8 (74_8)/15_8 (75_8) and 16_8 (76_8)/17_8 (77_8) for the word mode of operation. In both cases, switching between bytes 16_8 and 17_8 in the byte mode and 14_8/15_8 and 16_8/17_8 in the word mode is controlled by A0 (the least significant address bit). Refer to Table 5.

Table 3. MODE CONTROL OF PRIMARY PORT

MODE	$\overline{\text{PIOE}}$	$\overline{\text{DMAE}}$
Disabled (output)	1	1
Programmed I/O	0	1
DMA	X	0

X = Don't Care

Table 4 defines programmed I/O operation of the primary port in terms of read/write functions and Byte/$\overline{\text{Word}}$ control. In the byte mode, data is read-from or written-into the even addresses (0_8, 2_8, 4_8, 6_8, 10_8, 12_8, 14_8, and 16_8) via data lines $D0_A$ through $D7_A$; data is read-from or written-into odd addresses (1_8, 3_8, 5_8, 7_8, 11_8, 13_8, 15_8, and 17_8) via data lines $D0_B$ through $D7_B$. When A0 is low (logical 0), even addresses are selected and when A0 is high (logical 1), odd addresses are selected; thus, A0 is the LSB of a 4-bit address. In the word mode, the state of A0 is irrelevant, since both the odd and even bytes are, simultaneously, read-from or written-into; thus, a register pair is selected by a 3-bit address, A1 being the LSB.

In the DMA mode of operation with $\overline{\text{DMAE}}$ set to 0 and other conditions satisfied, data is directly transferred to-or-from specified memory locations under control of Byte/$\overline{\text{Word}}$, R/$\overline{\text{W}}$, and A0. The state of the Byte/$\overline{\text{Word}}$ control line determines whether the data word is 8 bits or 16 bits. The A0 address line correlates eight of

Table 4. PRIMARY PORT OPERATING IN PROGRAMMED I/O MODE

MODE	B/$\overline{\text{W}}$	A0	$D0_A$–$D7_A$ (Even Addresses)	$D0_B$–$D7_B$ (Odd Addresses)
Read	0 (Word)	X	Stored Data	Stored Data
Read	1 (Byte)	0	Stored Data	HI-Z
Read	1 (Byte)	1	HI-Z	Stored Data
Write	0 (Word)	X	Write	Write
Write	1 (Byte)	0	Write	No Change
Write	1 (Byte)	1	No Change	Write

X = Don't Care

the sixteen data lines ($D0_A$-$D7_A$ or $D0_B$-$D7_B$) with the proper byte/word location. Thus, in the word mode, the exchange of data between the memory and the primary port occurs via $D0_A$-$D7_A$ for bytes 14_8 and 16_8 and via $D0_B$-$D7_B$ for bytes 15_8 and 17_8. The byte mode of operation is similar, except that the unused eight lines are three-stated.

FUNCTION AND CONTROL OF SECONDARY PORT

The secondary port provides an 8-bit interface between the sixteen memory registers and the 8X305 (or other processor). As shown in Table 6, the secondary-port interface is controlled by five input signals and a status latch. The status latch is set when SC is high (MCLK high/$\overline{ME}$ low) and a valid memory address (60_8-77_8) is presented to the 8X320 via the secondary data port ($\overline{IV0}$-$\overline{IV7}$). The latch is cleared by internal logic when an invalid memory address is presented at the secondary port. In all read/write operations from the secondary port, the status latch acts like a master enable; data can be transferred only if the status latch is set.

Table 5. DMA OPERATION OF THE PRIMARY PORT

MODE	BYTE/WORD	A0	$D0_A$-$D7_A$	$D0_B$-$D7_B$
Read	0 (Word)	0	Data stored in byte 14_8	Data stored in byte 15_8
Read	0 (Word)	1	Data stored in byte 16_8	Data stored in byte 17_8
Read	1 (Byte)	0	Data stored in byte 16_8	HI-Z
Read	1 (Byte)	1	HI-Z	Data stored in byte 17_8
Write	0 (Word)	0	Write to byte 14_8	Write to byte 15_8
Write	0 (Word)	1	Write to byte 16_8	Write to byte 17_8
Write	1 (Byte)	0	Write to byte 16_8	HI-Z
Write	1 (Byte)	1	HI-Z	Write to byte 17_8

Table 6. FUNCTIONAL CONTROL OF SECONDARY PORT

$\overline{ME}$	SC[1]	WC[1]	MCLK	R/$\overline{W}$	STATUS LATCH	FUNCTION OF SECONDARY BUS
L	L	L	X	X	Set	Output data from 8X320 memory to 8X305
L	L	H	H	H	Set	Data from 8X305 is input and written-into a previously-selected memory location of the 8X320 (Note 2).
L	L	H	H	L	Set	With the primary port in the write mode (R/$\overline{W}$ = 0), the secondary port is overridden and cannot write to the same register addressed by the primary port; however, the register addressed by the primary port can be read and any other register can be read-from or written-into from the secondary port (Note 2).
L	H	L	H	X	X	Data transmitted to the secondary port via the $\overline{IV}$ bus is interpreted as an address; if address is within range of 60_8-77_8 the memory status latch is subsequently set.
L	L	H	L	X	X	Inactive
L	H	L	L	X	X	Inactive
L	L	X	X	X	Not Set	Inactive
H	X	X	X	X	X	Inactive

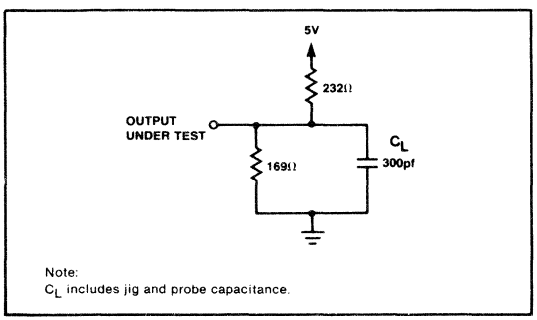
Notes:
1. The SC and WC lines should never both be high at the same time; the 8X305 processor never generates this condition.
2. During read or write operations, the same register can be simultaneously addressed from either port. For any write operation by both ports on the same register, the primary port has priority; other than this, the 8X320 does not indicate error conditions or resolve conflicts.
3. X = Don't Care.

DC CHARACTERISTICS $0°C \leq T_A \leq 70°C$; $4.75V \leq V_{CC} \leq 5.25V$

PARAMETER		TEST CONDITIONS[1, 2]	LIMITS			UNIT
			Min	Typ	Max	
V_{CC}	Supply voltage		4.75	5	5.25	V
$V_{IN\ (L)}$	Low level input voltage				0.80	V
$V_{IN\ (H)}$	High level input voltage		2.0			V
V_{OL}	Low level output voltage	V_{CC} = 4.75V; I_{OL} = 16mA			0.55	V
V_{OH}	High level output voltage	V_{CC} = 4.75V; I_{OH} = −3mA	2.40			V
V_{CL}	Input clamp voltage	I_I = −5mA			−1.00	V
I_{CC}	Supply current	V_{CC} = 5.25V (Both ports high-Z)			270	mA
I_{OS}	Short circuit output current[3]	V_{CC} = 4.75V	−20		−100	mA
$I_{IN\ (L)}$	WC, MCLK, SC, & $\overline{ME}$	V_{CC} = 5.25V; V_{IL} = 0.50V			−1.0	mA
$I_{IN\ (L)}$	B/$\overline{W}$	V_{CC} = 5.25V; V_{IL} = 0.50V			−1.6	mA
$I_{IN\ (L)}$	A0-A3	V_{CC} = 5.25V; V_{IL} = 0.50V			−1.0	mA
$I_{IN\ (L)}$	$\overline{DMAE}$	V_{CC} = 5.25V; V_{IL} = 0.5V			−800	μA
$I_{IN\ (L)}$	WS, $\overline{PIOE}$, & R/$\overline{W}$	V_{CC} = 5.25V; V_{IL} = 0.5V			−400	μA
$I_{IN\ (L)}$	$\overline{IV0}$-$\overline{IV7}$	V_{CC} = 5.25V; V_{IL} = 0.5V			−400 each line	μA
$I_{IN\ (L)}$	$D0_A$-$D7_A$/$D0_B$-$D7_B$	V_{CC} = 5.25V; V_{IL} = 0.5V			−400 each line	μA
$I_{IN\ (H)}$	WC, SC, MCLK, & $\overline{ME}$	V_{CC} = 5.25V; V_{IH} = 5.25V			100	μA
$I_{IN\ (H)}$	B/$\overline{W}$	V_{CC} = 5.25V; V_{IH} = 5.25V			240	μA
$I_{IN\ (H)}$	A0	V_{CC} = 5.25V; V_{IH} = 5.25V			120	μA
$I_{IN\ (H)}$	A1-A3	V_{CC} = 5.25V; V_{IH} = 5.25V			60	μA
$I_{IN\ (H)}$	$\overline{DMAE}$	V_{CC} = 5.25V; V_{IH} = 5.25V			120	μA
$I_{IN\ (H)}$	WS, $\overline{PIOE}$, & R/$\overline{W}$	V_{CC} = 5.25V; V_{IH} = 5.25V			60	μA
$I_{IN\ (H)}$	$\overline{IV0}$-$\overline{IV7}$ and $D0_A$-$D7_A$/$D0_B$-$D7_B$	V_{CC} = 5.25V; V_{IH} = 5.25V			100	μA

Notes:
1. *Operating temperature ranges are guaranteed after terminal equilibrium has been reached.*
2. All voltages are measured with respect to ground terminal.
3. Short only one output at a time.

TEST CIRCUIT



Note:
C_L includes jig and probe capacitance.

AC CHARACTERISTICS OF PRIMARY PORT

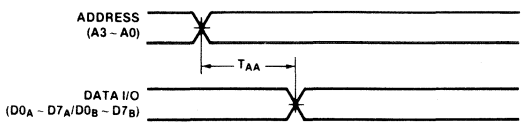
$0°C \leq T_A \leq 70°C$; $4.75V \leq V_{CC} \leq 5.25V$
Loading: See Test Circuit

Legend:

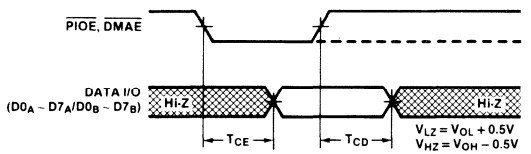
CHANGING DATA

Hi-Z

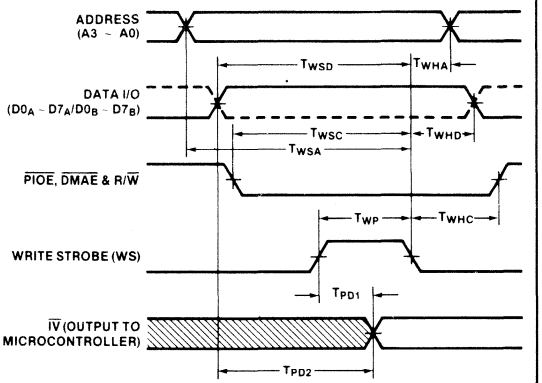
ADDRESS ACCESS TIME



PRIMARY PORT ENABLE/DISABLE TIMES



WRITE CYCLE

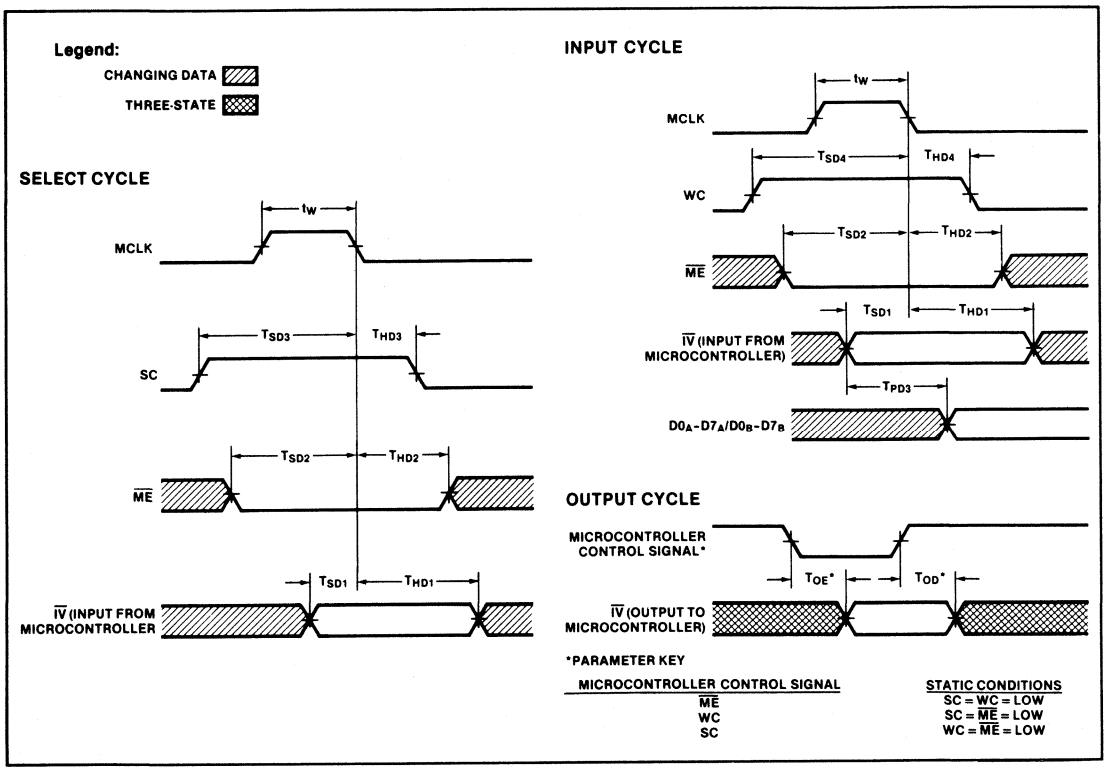


	PARAMETER	FROM	TO	LIMITS Min	LIMITS Typ	LIMITS Max	UNIT
T_{AA}	Address Access Time	A3-A0	$D0_A$-$D7_A$/$D0_B$-$D7_B$			45	ns
T_{CE}	Primary port enable time	↓$\overline{PIOE}$ ↓$\overline{DMAE}$	$D0_A$-$D7_A$/$D0_B$-$D7_B$			30	ns
T_{CD}	Primary port disable time	↑$\overline{PIOE}$ ↑$\overline{DMAE}$	$D0_A$-$D7_A$/$D0_B$-$D7_B$			35	ns
T_{WSA}	Address setup time	A3-A0	↓WS	40			ns
T_{WHA}	Address hold time	↓WS	A3-A0	0			ns
T_{WSD}	Primary port data setup time	$D0_A$-$D7_A$/$D0_B$-$D7_B$	↓WS	30			ns
T_{WHD}	Primary port data hold time	↓WS	$D0_A$-$D7_A$/$D0_B$-$D7_B$	0			ns
T_{WSC}	Write mode control setup time	$\overline{PIOE}$ $\overline{DMAE}$ R/$\overline{W}$	↓WS ↓WS ↓WS	30 40 30			ns ns ns
T_{WHC}	Write mode control hold time	↓WS	$\overline{PIOE}$ $\overline{DMAE}$ R/$\overline{W}$	10 10 10			ns ns ns
T_{WP}	Write strobe pulse width			25			ns
T_{PD1}[1]	Primary port data delay	$D0_A$-$D7_A$/$D0_B$-$D7_B$	$\overline{IV0}$-$\overline{IV7}$			75	ns
T_{PD2}[2]	Primary port data delay from WS	↑WS	$\overline{IV0}$-$\overline{IV7}$			75	ns

Notes:
1. Measurement with Write Strobe set High and the control signals of the secondary port set for output data from the same register.
2. Measurement with primary port data stable and control signals of secondary port set for output data from the same register.

AC CHARACTERISTICS OF SECONDARY PORT

$0°C \leq T_A \leq 70°C$, $4.75V \leq V_{CC} \leq 5.25V$
Loading: See Test Circuit



	PARAMETER	FROM	TO	LIMITS			UNIT
				Min	Typ	Max	
t_W	MCLK pulse width			30			ns
T_{SD1}	Data setup time	$\overline{IV0}$-$\overline{IV7}$	↓MCLK	35			ns
T_{SD2}	$\overline{ME}$ setup time	$\overline{ME}$	↓MCLK	30			ns
T_{SD3}	SC setup time	SC	↓MCLK	30			ns
T_{SD4}	WC setup time	WC	↓MCLK	30			ns
T_{HD1}	Data hold time	↓MCLK	$\overline{IV0}$-$\overline{IV7}$	0			ns
T_{HD2}	$\overline{ME}$ hold time	↓MCLK	$\overline{ME}$	0			ns
T_{HD3}	SC hold time	↓MCLK	SC	0			ns
T_{HD4}	WC hold time	↓MCLK	WC	0			ns
T_{PD3} (Note)	$\overline{IV}$ propagation delay	$\overline{IV}$	$D0_A$-$D7_A$/$D0_B$-$D7_B$			45	ns
T_{OE}	Output enable	$\overline{ME}$, SC, or WC	$\overline{IV0}$-$\overline{IV7}$			30	ns
T_{OD}	Output disable	$\overline{ME}$, SC, or WC	$\overline{IV0}$-$\overline{IV7}$			20	ns

Note:
Measured with MCLK = High and control signals of the primary port set for output data from the same register.

FLOPPY DISK FORMATTER/CONTROLLER

Originally published by Signetics January 1984

FEATURES

- **Single or double density encoding/decoding**
- **On-chip data separator**
- **Programmable:**
  - **FM, MFM, and M²FM encoding/decoding**
  - **Preamble Polarity**
  - **Data transfer rate**
  - **Address mark encoding/decoding**
  - **Sector length**
  - **Output port (7-bits disk command)**
  - **Input port (5-bits disk status)**
- **Write Precompensation with on/off control**
- **On-chip phase lock loop**
- **CRC generator with software-controlled error correction capabilities**
- **40-pin package**
- **+5 volt operation**

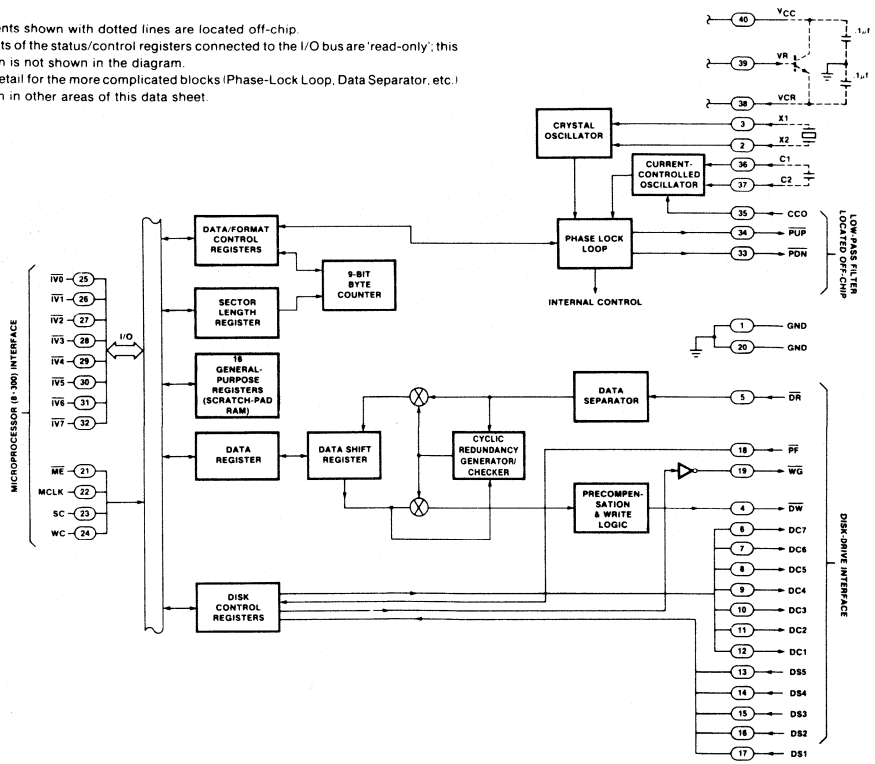
PRODUCT DESCRIPTION

The Signetics 8X330 Floppy Disk Formatter/Controller is a monolithic peripheral device of the 8X300 Family. The chip uses Bipolar-Schottky/I²L-Technology and some very unique features to provide 8X330 customers with a competitive edge in both simple and complicated disk-controller designs. The competitive advantage is measurable in terms of "systems parts count", "error correction capabilities", and "overall design concepts" that are applications oriented. Except for a crystal, a capacitor, an external transistor acting as a series-pass element for the on-chip voltage regulator, an active low-pass filter, and an optional off-chip voltage controlled oscillator (refer to Features and Option), the 8X330 contains all processing circuits and the required control logic to encode/decode double-density (MFM/M²FM) and single-density (FM) codes. Even the data-separation and write-precompensation logic are located on the chip; in addition, 16-bytes of scratch-pad RAM are provided for storage of various control/status parameters.

OPTION: External Voltage Controlled Oscillator (VCO). For critical applications, window margins can be improved by as much as 6% with the use of an external VCO.

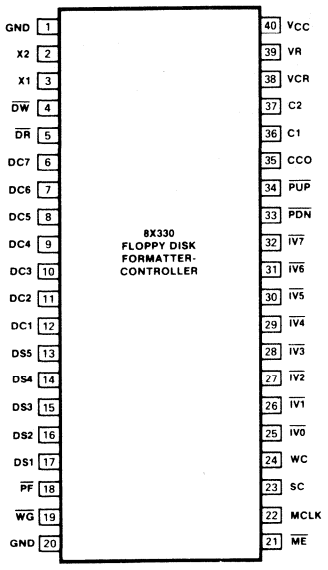
BLOCK DIAGRAM

NOTES
1. Components shown with dotted lines are located off-chip.
2. Certain bits of the status/control registers connected to the I/O bus are 'read-only'; this distinction is not shown in the diagram.
3. Greater detail for the more complicated blocks (Phase-Lock Loop, Data Separator, etc.) are shown in other areas of this data sheet.



8X330 PACKAGE/PIN DESIGNATIONS

N PACKAGE



TOP VIEW

ORDER NUMBER

N8X330N

PIN NO.	MNEMONIC & DEFINITION		FUNCTION
1, 20	GND	Ground	Circuit ground
2, 3	X1, X2	Crystal inputs	Inputs from a crystal that determines frequency of an on-chip crystal oscillator.
4	$\overline{DW}$	Data write	A series of negative-going pulses transmitted to the disk drive. The data write signal produces pulses (with precompensation, if required) for data and clock in accordance with the applicable encoding rules (FM, MFM or M²FM).
5	$\overline{DR}$	Data read	Negative-going pulses transmitted from the disk drive to a Schmitt-trigger input of the 8X330; these pulses represent encoded data and clock from the disk media.
6-12	DC1-DC7	Disk commands	Seven outputs from the 8X330 that allow general-purpose control, of one or more disk drives.
13-17	DS1-DS5	Disk status	Five general-purpose Schmitt-trigger inputs from the disk drive (or drives) that provide status information for the 8X300.
18	$\overline{PF}$	Power fail	Schmitt-trigger input from external logic that is active (low) when the "user-sensed" power supply voltage drops below a predetermined value.
19	$\overline{WG}$	Write gate	When active (low), this 40-milliampere open-collector output enables writing to the disk media. When $\overline{PF}$ is low, the write gate is inhibited during periods of power supply uncertainty.
21	$\overline{ME}$	Master enable	When this input signal is active (low), the 8X330 can be accessed and enabled by the 8X300. (Refer to the $\overline{LB}$ and $\overline{RB}$ pinout discriptions of the 8X300 for further detail.)
22	MCLK	Master clock	When active high and with $\overline{ME}$ in the active-low state, this input signal provides a means whereby the I/O output from the 8X300 is interpreted as an enabling address (provided there is an address match) or as input data (if one of the 8X330 registers has already been selected).
23	SC	Select command	When this signal is active (high), the information output on pins $\overline{IV0}$-$\overline{IV7}$ of the 8X300 is interpreted as an address input by the 8X330.
24	WC	Write command	When this signal is active (high), the information output on pins $\overline{IV0}$-$\overline{IV7}$ of the 8X300 is interpreted as input data by the 8X330.
25-32	$\overline{IV0}$-$\overline{IV7}$	Input/output lines	Eight three-state input/output lines that provide bidirectional data transfers between the 8X300 and the enabled I/O device; $\overline{IV7}$ is the *Least Significant Bit.*
33	$\overline{PDN}$	Pump down output	Open-collector output of on-chip phase detector which indicates (by a negative-going, quantized, pulse-width modulated signal) that internal CCO frequency is too high.
34	$\overline{PUP}$	Pump up output	Open-collector output of on-chip phase detector which indicates (by a negative-going, quantized, pulse-width modulated signal) that internal CCO frequency is too low.
35	CCO	Frequency Control Input for Current-Controlled Oscillator	Variable input current from external low-pass filter that controls the frequency of the oscillator.
36-37	C1, C2	Capacitor input terminals	Inputs for capacitor that determines center frequency of the current-controlled oscillator.
38	VCR	Regulated supply voltage	DC voltage input from emitter of external series-pass transistor; this voltage powers internal logic of chip.
39	VR	Reference voltage	Reference voltage output to base of series-pass transistor; this reference controls VCR.
40	V_{CC}	Supply voltage	+5 volt power.

SYSTEM INTERFACE

A typical floppy disk controller using an 8X300 microcontroller and the 8X330 is shown in Figure 2. The non-shaded portion of this particular configuration can service the command, status, and input/output requirements of two double-sided disk drives and, under software supervision, the system can read/write single-density (FM) or double-density (MFM/M²FM) codes. Interface requirements are simple—on one hand, consisting of the 8X300 microcontroller and, on the other, the two disk drives. All 8X330 control and data registers directly linked to the microprocessor interface (Figure 1) are addressable and appear to the 8X300 as simple I/O ports; a 13-bit address bus and a 16-bit instruction bus provide communications between the 8X300 and up to 8K of microprogram storage.

The disk-drive interface consists of seven (7) output control lines (DC1-DC7), five (5) input status lines (DS1-DS5), a write gate ($\overline{WG}$), a data- write output ($\overline{DW}$), and a data-read input ($\overline{DR}$). The twelve command/status lines are not dedicated; thus, the user can assign system functions to best suit a given application. As shown in Figure 2, all control lines except $\overline{WG}$ are buffered to accommodate a reasonable distance between the controller and the disk media; the Write Gate, being a 40-milliampere output, requires no buffering.

As shown by the shaded part of Figure 2, the control and status lines can be expanded with peripheral hardware—the 8T32 (in this example) being only one method of implementation. Using this particular technique, one I/O port is totally dedicated to output control, whereas, the other port is totally dedicated to input status. With additional hardware and supporting software, the disk-drive system can be expanded without limit; however, from a point of being practical, five or six drives is sufficient for most applications. By using the programmable features of the 8X330, the user can emphasize and prioritize those system parameters that are most important—economics, reliability and/or speed.

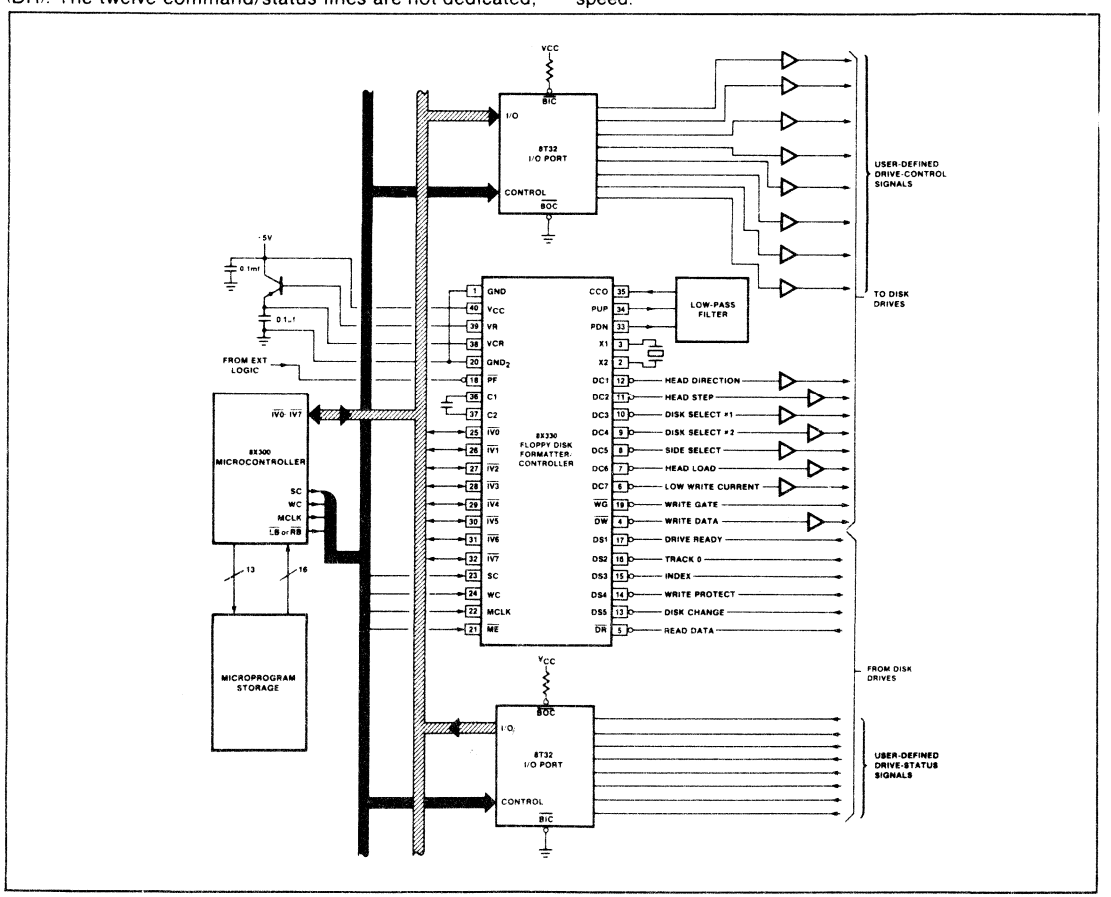


Figure 2. Typical Interface Using an 8X300 Microcontroller

FUNCTIONAL OPERATION

As shown in Figure 2, the interface between the 8X300 and 8X330 consists of twelve (12) lines—$\overline{IV0}$-$\overline{IV7}$, SC, WC, MCLK, and $\overline{ME}$; the Master Enable ($\overline{ME}$) input (pin 21) is driven from either the $\overline{LB}$ (Left Bank) or $\overline{RB}$ (Right Bank) output of the 8X300. An expanded view of this interface is shown in Figure 3 and, as indicated, the 8X330 appears as a number of addressable registers (110_8-127_8 and 132_8-137_8) under input/output control of the 8X300. These registers are used for general-purpose storage, data-transfer operations, disk commands, disk status, and various control functions. Design-oriented information for these registers and other data-processing/logic functions of the 8X330 are described in the paragraphs that follow; in all of these registers, bit 0 is the Most Significant Bit (MSB).

NOTE
When power is first applied to the 8X330, the Disk Command lines (DC1-DC7), the Write Gate ($\overline{WG}$) output, and contents of Command/Status Register #2 (CSR #2) are set to 1 (high). The wakeup state of all other bits is undefined.

General-Purpose Register File

This general-purpose (scratch pad) memory is directly accessible by the 8X300 and is used to store system variables such as track address, sector address and other necessary parameters. The sixteen 8-bit registers (110_8-127_8) provide sufficient on-chip memory to accommodate a minimum of two disk drives; the maximum number of drives that this non-dedicated memory file can support depends on several factors—system configuration, reliability requirements, economic constraints, and so on. Because of the on-chip file, all other system memory can be dedicated to the purpose of handling data to-and-from the disk media.

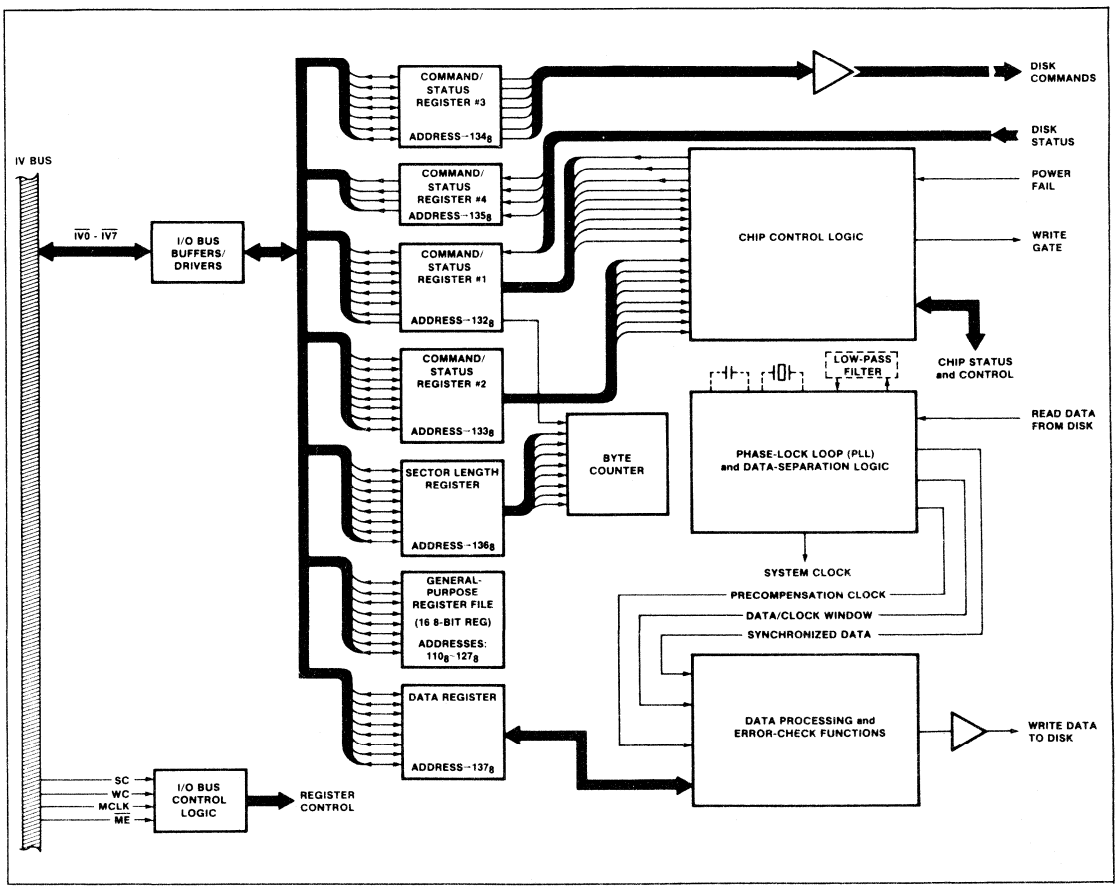


Figure 3. An Expanded View of the 8X330/8X300 Interface

Command/Status Register #1 (CSR 1/Address 132_8)

The disk status (read) or disk-command (write) contents of this register are interpreted as follows; unless otherwise indicated, all bits of CSR 1 are read/write from the I/O bus.

Bit 0 (Write Gate Enable)
Enables write gate output ($\overline{WG}$/pin 19) to disk drive(s)—the write gate ($\overline{WG}$) cannot be enabled unless the PF input pin (18) is high. When the WGE bit is set to 0, the $\overline{WG}$ ouput pin is low (enabled); when WGE is set to 1, the $\overline{WG}$ output is high (disabled). If the PF input goes low while the $\overline{WG}$ output is low, the $\overline{WG}$ output will go high and the Write Gate Enable bit is reset to 1.

Bit 1 (CRC Enable)
When set to 1, permits internal CRC register to compute remainders on the data stream in either read or write modes of operation. When set to 0, the CRC register becomes the source of data. A change in the CRC Enable bit does not become effective until the "next BYTRA flag appears" following the command bit change—refer to description of CSR 1/Bit 6.

Bit 2 (Data Register Control)
When set to 0, contents of data register consists of interleaved data-and-clock bits; starting from the MSB ($\overline{IV0}$) position, register contents are: Clock 1, Data 1, Clock 2, Data 2, Clock 3, Data 3, Clock 4, and Data 4. When writing an address mark, the appropriate data/clock pattern is loaded into the data register by the 8X300. Since each byte of data from the processor becomes an interleaved pattern (4-bits of data and 4-bits of clock) in the 8X330 data register, two bytes from the processor are required to write each full byte of address mark to the drive—eight bit cells with each cell containing a possible data and/or clock transition, or a total of 16 bit positions. When writing address marks, the normal on-chip clock insertion circuitry of the 8X330 is inhibited; thus, the user is free to define any clock/data pattern for the address mark.

When reading address marks, the data register is loaded with data and clock representing four bit cells from the disk media. The information in the data register can then be compared with the expected address mark by the 8X300 on a nibble-by-nibble basis. When the DRC bit is set to 1, the data register contains separated data (no clocks). A state change in this bit does not become effective until the "next BYTRA flag appears" following the state-change command.

Bit 3 (Sync Enable)
The Sync Enable bit allows the on-chip data separator to obtain bit and byte synchronization; this bit also controls initialization of the CRC Register. With the 8X330 in Read mode and with Bit 3 set to 0, bit synchronization occurs. The Preamble field is assumed to be all "zeroes" or all "ones" as determined by the Preamble Select bit (CSR 2/Bit 4).

When the proper number of preamble bytes, as determined by the disk-control program, have been found, the Sync Enable bit should be changed, under program direction, to a 1. This puts the 8X330 in the Address-Mark search mode. Accordingly, all bits of the CRC Register are preset to 1, the BYte TRAnsfer flag is inhibited, and the 8X330 examines the data stream for an Address Mark. The Address Mark is detected by observing the data and clock bits to find a change in the normal Preamble pattern. Byte synchronization is achieved by assuming that the change occurred in the bit cell determined by two Bit Select bits (CSR 2/Bits 2 and 3).

When the pattern change is found indicating the start of an Address Mark, the 8X330 starts CRC computation and synchronizes BYTRA to the byte boundaries. Note that the 8X330 presumes an Address Mark by finding a change in the preamble pattern; however, it is up to the 8X300 to read the Address Mark and to establish its validity or non-validity.

In write mode, setting the Sync Enable bit to 0 presets all bits of the CRC Register to 1. Setting the Sync Enable bit to 1 allows CRC computation to begin at the next byte boundary.

Bit 4 (Load Counter)
When set to 1, transfers 8-bits of data from Sector Length register and 1-bit (MSB) of data from Byte Counter (refer to next description) to 9-bit Byte Counter. Loading of the 9-bit Byte Counter is effective one bit-cell time after the Load Counter bit is set to 1. In both the read and write modes of operation, the Byte Counter is incremented by BYTRA. The Load Counter bit is self-clearing and always returns a 0 when read.

NOTE
The Load Counter bit must be set one or more instruction cycles *after* setting the Byte Counter MSB, that is, bits 4 and 5 of CSR 1 cannot be set during th same instruction cycle.

Bit 5 (Byte Counter MSB)
This bit is used to set and monitor the state of the ninth (MSB) bit in the Byte Counter; reading this bit always returns the current state of MSB in the Byte Counter. The MSB of the Byte Counter is set to the value of CSR 1/Bit 5 when the Load Counter bit (CSR 1/Bit 4) is asserted—refer to preceding description.

NOTE
The Byte Counter MSB must be set one or more instruction cycles *before* the Load Counter bits—bits 4 and 5 of CSR 1 cannot be set during the same instruction cycle.

Bit 6 (BYTRA)
During a disk read operation, the BYte TRAnsfer flag is automatically set to 0 when 8-bits of information are transferred from the Data Shift Register to the Data Register —see Figure 1. During a disk write operation, BYTRA is automatically set to 0 when 8-bits are transferred from the Data Register to the Data Shift Register. BYTRA (a read-

only bit) is reset to a 1 when the Data Register (address 137_8) is selected by the user's program. During read/write operations, the 1-to-0 transition of the BYTRA flag increments the Byte Counter to keep count of bytes read or bytes written. All read-only bits of the 8X330 are designed to remain stable during the monitor period; thus, to read a status change of BYTRA, Disk-Status bit, the Byte Counter MSB, or other read-only bit requires a two-instruction loop similar to:

```
TEST    SEL   CSR 1
        NZT   BYTRA,   TEST
```

Bit 7 (Disk Status 1)
Reflects state (0 or 1) of input DS1 (pin 17); this is a user-definable read-only bit.

NOTE
A high input on any one of the Disk Status lines of the 8X330 is read by the 8X300 program as a logical 1 and a low input on the status lines is read as a logical 0.

Command/Status Register #2 (CSR 2/Address 133_8)

The disk status (read) or disk-command (write) contents of this register are interpreted as follows:

Bit 0 (Precompensation Enable)
This command bit determines whether or not precompensation is applied to the data stream being written onto the disk. When set to 0, precompensation is inhibited. When set to 1 and with double-density encoding, write precompensation is applied to the following data/clock bit patterns:

Precomp Time	Data/Clock Pattern (in Data Shift Reg)			Bit Being Written	Bits Already Written to Disk		
2T (Late)	0	1	**0**	1	0	0	0
2T (Late)	0	1	**0**	1	0	0	1
2T (Early)	1	0	**0**	1	0	1	0
2T (Early)	0	0	**0**	1	0	1	0

where, $T = \frac{1}{\text{crystal frequency}}$ if bit 7 of CSR 2 ($1/2F$) = 1

$T = \frac{2}{\text{crystal frequency}}$ if bit 7 of CSR 2 ($1/2F$) = 0

Bit 1 (Read Mode)
When set to 0, the 8X330 reads data from the disk and transfers it to the Data Register; when set to 1, data from the Data Register is transferred to the disk, provided the Write Gate Enable bit (CSR1/Bit 0) is set to 1. With WGE set to 0 and the Read Mode bit set to 1, the current-controlled oscillator is forced to lock onto the crystal oscillator; this technique is used during a data-read operation to ensure rapid acquisition of the disk data.

Bits 2,3 (Bit Selects 1 and 0)
Together with the Sync Enable (CSR 1/Bit 3), these two bits allow the user to establish byte boundaries for the data stream; this is done in the following way. After bit synchronization is established, and the preamble pattern is verified, the 8X330 looks for a change in the normal preamble pattern. As shown in the following truth table, Bit Select 1 (Bit 2) and Bit Select 0 (Bit 3) identifies the bit cell within the first nibble of the first Address-Mark byte in which the first deviation from the normal preamble is expected. BYTRA is always referenced to bit cell 0.

BS 0	BS 1	Bit Cell
0	0	0
0	1	1
1	0	2
1	1	3

Bit 4 (Preamble Select)
This bit is used only for bit sychronization—refer to CSR 1/Bit 3. With Bit 1 of CSR 2 set to 0 (Read Mode) and the Preamble Select bit set to 0, the preamble field is assumed to be all zeroes; with the Preamble Select it set to 1, the preamble field is assumed to be all ones. In either case, preamble validity is determined by the 8X300.

Bits 5,6 (E1 and E2)
Together, E1 and E2 select the encoding scheme used to write data on the disk—refer to truth table that follows.

E1	E2	Encoding Scheme
1	X	FM
0	0	MFM
0	1	M²FM

x = don't care

Bit 7 (1/2F)
This bit allows the data transfer rate to be changed without modification of the frequency-selective components in the data-separation logic; thus, differences in data transfer rates between standard-and-mini floppies can be accommodated via software—no component or other hardware changes. Assuming an 8 MHz crystal and with the 1/2F bit set to 1, the data transfer rate is 250K-bits per second in the single-density (FM) mode and 500K-bits per second in the double-density (MFM/M²FM) mode. When set to 0, the transfer rates are halved—125K-bits and 250K-bits, respectively. When using frequencies other than 8 MHz, the data transfer rate is determined as follows:

Bit 7 ($1/2F$)	Single-Density (FM)	Double-Density (MFM/M²FM)
0	$\frac{\text{xtal freq}}{64}$	$\frac{\text{xtal freq}}{32}$
1	$\frac{\text{xtal freq}}{32}$	$\frac{\text{xtal freq}}{16}$

Command/Status Register #3 (CSR 3/Address 134_8)

This register contains seven bits (Bit 0 through Bit 6) which determines the state of the disk-command outputs; writing to Bit 7 has no effect and reading Bit 7 always returns a zero. When a logical "1" is specified by the 8X300 program for a given disk-command line, a high will appear at the output of the 8X330 for that particular command line. Each bit and the output pin it controls are summarized below.

Bit (CSR 3)	Control Function	Pkg Pin No.
0	DC1 Output	12
1	DC2 Output	11
2	DC3 Output	10
3	DC4 Output	9
4	DC5 Output	8
5	DC6 Output	7
6	DC7 Output	6

Command/Status Register #4 (CSR 4/Address 135_8)

This register contains four bits (Bit 0 through Bit 3) which reflect the state of the disk-status inputs to the 8X330; reading all other bits (4 through 7) always returns a zero. These read-only bits and the reflected status they represent are as follows; the information specified by notation for Bit 7/CSR 1 is applicable to these input lines.

Bit (CSR 4)	Control Function	Pkg Pin No.
0	DS2 Input	16
1	DS3 Input	15
2	DS4 Input	14
3	DS5 Input	13

Sector Length Register—Address 136_8

This register contains the load value for the lower eight (LSBs) bits of the Byte Counter. Data is transferred from the Sector Length Register to the Byte Counter under control of Load Counter Bit in CSR 1. When the contents of this register are transferred to another location via a read or write commands, the original holding of data is not lost; thus, if the same data is to be used more than once, a repetitive read or write can be implemented without reloading the register.

Data Register—Address 137_8

Together with the Data Shift Register, the Data Register is used for bidirectional transfer of data between the 8X330 and the I/O bus. All transfers to-and-from this register are made in conjunction with Bit 6 (BYTRA—Byte Transfer Flag) of CSR 1. When the Data Register Control bit (CSR 1/Bit 2) is set to 0, the content of this register is interleaved with four bits of data and four bits of clock. When data is transferred from the Data Register to the Data Shift Register, the original content of the Data Register is not lost.

Phase Lock Loop (PLL) and Data Separation Logic

An expanded view of the phase-lock loop and the data-separation logic is shown in Figure 4. Basically, the PLL consists of two counters, a phase detector, and a feedback loop containing a low-pass filter (off-chip) that controls a phase-locked oscillator (CCO). In simplified form, the data-separation logic consists of data flip-flops (pulse synchronizer) and other circuits required to separate data and clock transitions. In the read mode, the output of the phase-locked oscillator (CCO) is applied to the clock inputs of counter #1, counter #2, and the pulse synchronization circuits. Essentially, the frequencies of the two counters are identical (phase relationships may or may not be identical); to maintain proper frequencies and to continuously correct for any phase deviations, the following actions occur.

Preset values which represent, respectively, nominal mid-points of the clock and data windows are present at counter #2 and, when an output appears at the pulse synchronizer, these preset values are entered. The count sequence for both counters is from "0 to F"; hence, the phase difference between Carry 1 (counter #1) and Carry 2 (counter #2) actually corresponds to any phase deviation between the CCO and the synchronized data from the disk. The phase detector measures the phase difference between the two carry inputs and produces a series of quantized pulses whose widths are proportional to the phase error at the end of each counting cycle. After integration by the low-pass filter, a current proportional to the phase error is applied to the current-controlled oscillator. Accordingly, the CCO is driven in a direction (pump-up or pump-down) to correct any phase difference between the synchronized disk data and the feedback-controlled clock. Phase detector characteristics for both single-and-double density formats are shown in Figures 5 and 6.

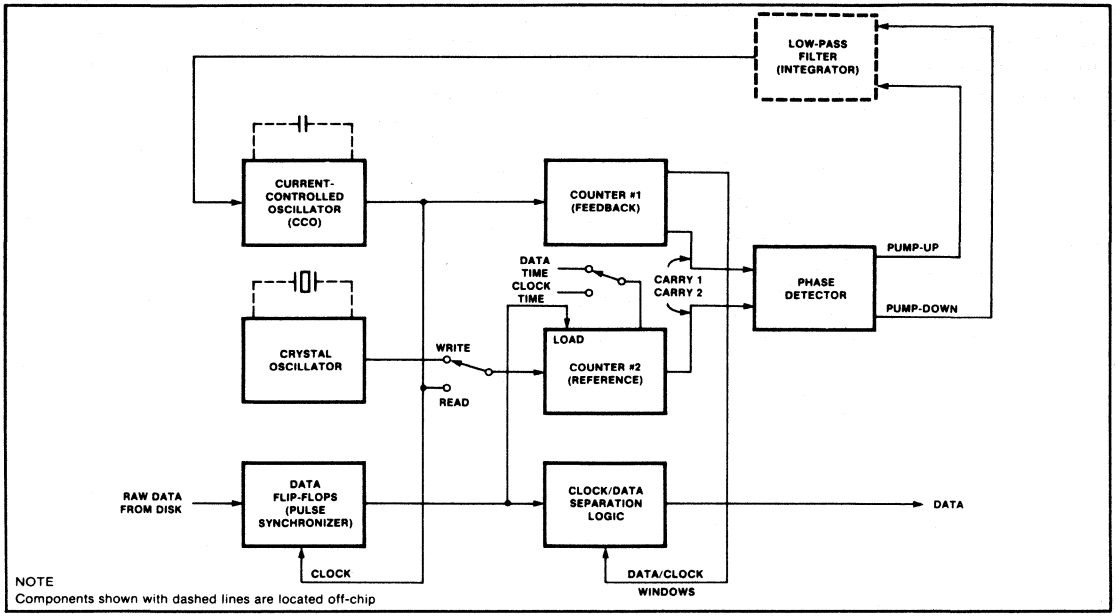


Figure 4. Simplified Block of Phase-Lock Loop

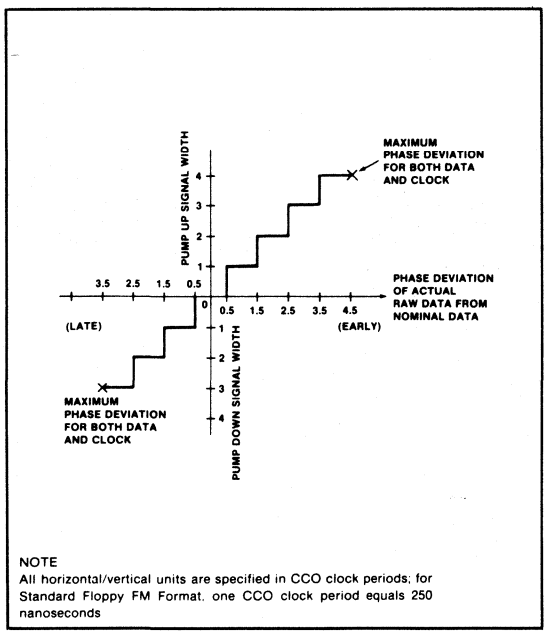


Figure 5. Phase Detector Characteristic for Single-Density (FM) Format

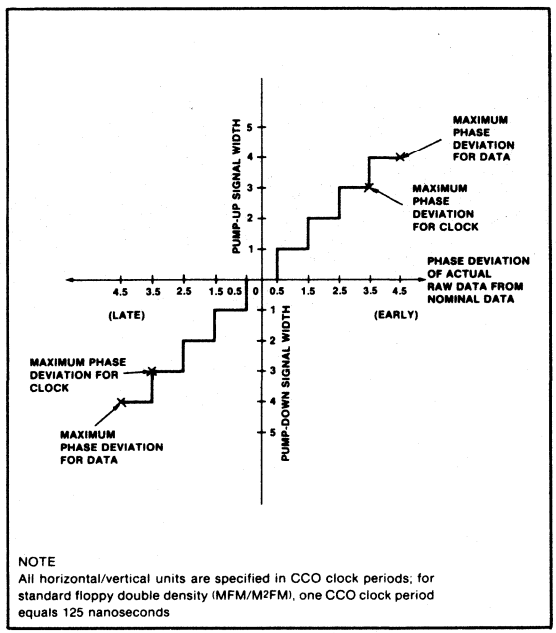


Figure 6. Phase Detector Characteristic for Double-Density (MFM/M²FM) Format

Data Processing and Error-Check Functions

These functions of the 8X330 are summarized in Figures 7 and 8. The read/write operations are software-controlled by previously-described bits of command/status registers CSR1 and CSR2. For the sake of simplicity, control lines and much of the control logic associated with the data processing and error-check functions are omitted in the read/write diagrams.

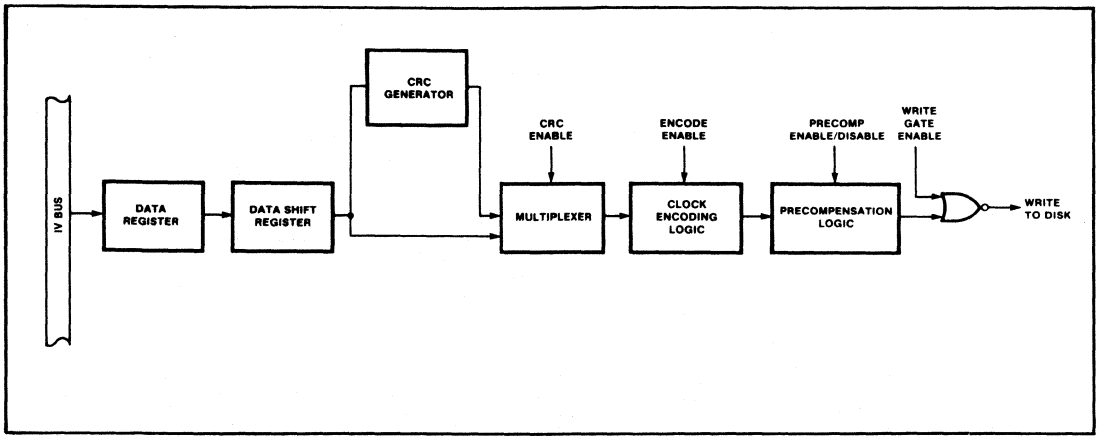


Figure 7. Simplified Block of Data Processing and Error Check Functions—Write Mode

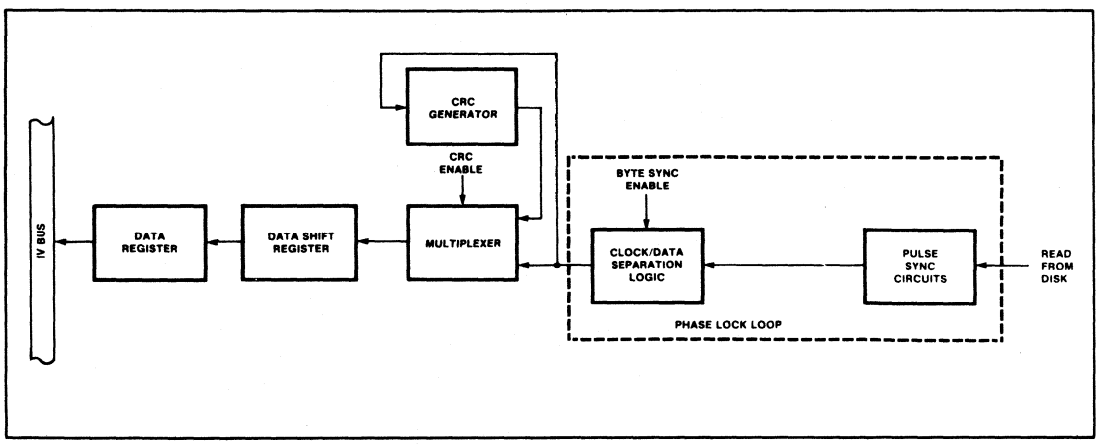


Figure 8. Simplified Block of Data Processing and Error Check Functions—Read Mode

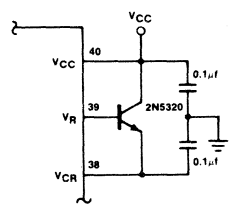
DC CHARACTERISTICS V_{CC} = 5V (±5%), T_A = 0°C to 70°C

PARAMETER		TEST CONDITIONS	LIMITS			UNITS	COMMENTS
			Min	Typ	Max		
V_{IH}	High level input voltage		2.0		V_{CC}	V	For all inputs except X1, X2, C1, C2, CCO, and V_{CR}
V_{IL}	Low level input voltage		−1.0		0.8	V	
V_{CC}	Supply voltage		4.75	5.0	5.25	V	5V (±5%)
V_{CR}	Regulator voltage	V_{CC} = 5V		3.1		V	From series-pass transistor
V_{CL}	Input clamp voltage	V_{CC} = Min I_{IN} = −5mA	−1.0			V	Inputs X1, X2, C1, C2, and CC0 do not have internal clamp diodes.
V_{OH}	High level output voltage	V_{CC} = Min; I_{OH} = −0.4mA	2.7			V	DC1 through DC7 (Pins 6-12) & $\overline{DW}$ (Pin 4)
		V_{CC} = Min; I_{OH} = −3mA	2.4			V	$\overline{IV0}$-$\overline{IV7}$ (Pins 25-32)
V_{OL}	Low level output voltage	V_{CC} = Min; I_{OL} = 8mA			0.5	V	DC1 through DC7 (Pins 6-12); $\overline{PUP}$, $\overline{PDN}$ (Pins 33, 34); $\overline{DW}$ (Pin 4)
		V_{CC} = Min; I_{OL} = 16mA			0.55	V	$\overline{IV0}$-$\overline{IV7}$ (Pins 25-32)
		V_{CC} = Min; I_{OL} = 40mA			0.55	V	$\overline{WG}$ (Pin 19)
I_{CEX}	Open-collector leakage current with output set to 1.	V_{CC} = Min; V_{OUT} = V_{CC}			100	μA	$\overline{WG}$ (Pin 19); $\overline{PUP}$ (Pin 34); $\overline{PDN}$ (Pin 33)
I_{IH}	High level input current	V_{CC} = Max; V_{IN} = 2.7V			20	μA	DS1-DS5 (Pins 13-17); $\overline{PF}$ (Pin 18); $\overline{DR}$ (Pin 5)
					40	μA	$\overline{ME}$ (Pin 21); MCLK (Pin 22); SC (Pin 23); WC (Pin 24)
		V_{CC} = Max; V_{IN} = 5.25V; CCO (Pin 35) input current = 0mA			4	mA	With C1 (Pin 36) under test, C2 (Pin 37) is open and, vice-versa.
		V_{CC} = Max; V_{IN} = 5.25V CCO (Pin 35) input current = 1mA			2	mA	
		V_{CC} = Max; V_{IN} = 0.6V			4	mA	With X2 (Pin 2) under test, X1 (Pin 3) is open and, vice-versa.
		V_{CC} = Max; V_{IN} = 4.5V			50	μA	$\overline{IV0}$ - $\overline{IV7}$ (pins 25-32)
V_{CCO}	Input voltage for current-controlled oscillator	V_{CC} = 5V; T_A = 25°C CCO input current (Pin 35) = 300 μA		750		mV	

DC CHARACTERISTICS (Cont'd) V_{CC} = 5V (±5%), T_A = 0°C to 70°C

PARAMETER		TEST CONDITIONS	LIMITS			UNITS	COMMENTS
			Min	Typ	Max		
I_{IL}	Low level input current	V_{CC} = Max; V_{IN} = 0.4V			−400	µA	DS1-DS5 (Pins 13-17); $\overline{PF}$ (Pin 18); $\overline{DR}$ (Pin 5)
					−800	µA	$\overline{ME}$ (Pin 21); MCLK (pin 22); SC (Pin 23); WC (Pin 24)
					−4	mA	X1 (Pin 2), X2 (Pin 3), with X1 under test, X2 is open and, vice-versa.
		V_{CC} = Max V_{IN} = 0.5V			−550	µA	$\overline{IV0}$-$\overline{IV7}$ (Pins 25-32)
I_{OS}	Output short-circuit current	V_{CC} = Max; Output = "1"; V_{OUT} = "0".	−15		−100	mA	DC1-DC7 (Pins 6-12) & $\overline{DW}$ (Pin 4)
		(NOTE At any time, no more than one output should be connected to ground)	−30		−140	mA	$\overline{IV0}$-$\overline{IV7}$ (Pins 25-32)
I_{CC}	(Pin 40)	V_{CC} = Max			200	mA	
I_{CR}		V_{CC} = Max			250	mA	
I_{REG}	(Pin 39)	V_{CC} = 5V; V_{CR} = 0V & V_R = 2V	−16		−27	mA	

NOTES
1. Operating temperature ranges are guaranteed after thermal equilibrium has been reached.
2. All voltages measured with respect to ground terminal.
3. Unless otherwise specified, each test requires that V_{CR} be supplied through a series-pass transistor as shown in the accompanying drawing.



AC CHARACTERISTICS V_{CC} = 5V (±5%), T_A = 0°C to 70°C

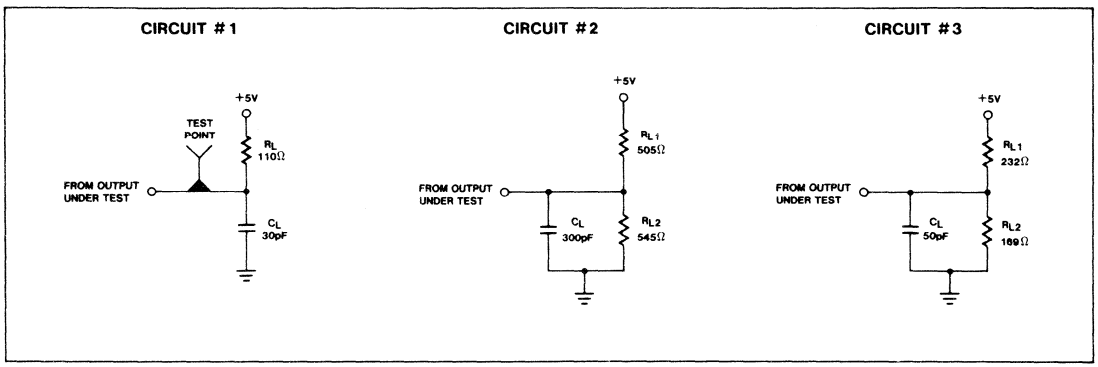
MNEMONIC	REFERENCE	INPUT	OUTPUT	Min	Typ	Max	COMMENTS
t_{PD}		$\overline{PF}$	$\overline{WG}$			60ns	Refer to Note 3 and Test Loading Circuit #1.
t_{PD}		MCLK	$\overline{WG}$			100ns	
t_{PD}		MCLK	$\overline{WG}$			100ns	
t_{PD}		MCLK	DC1-7			70ns	Refer to Note 3 and Test Loading Circuit #2.
t_{PD}		MCLK	DC1-7			70ns	
t_{pw}		$\overline{PF}$		50ns			
t_{pw}		$\overline{DR}$		50ns			
t_{pw}			$\overline{DW}$				Note 1
t_{SETUP}	SC	Input on DS1-5		55ns			Note 2
t_{SETUP}	WC	Input on DS1-5		55ns			Note 2
t_{SETUP}	$\overline{ME}$	Input on DS1-5		55ns			Note 2
t_{HOLD}	SC	Input on DS1-5		0ns			Note 2
t_{HOLD}	WC	Input on DS1-5		0ns			Note 2
t_{HOLD}	$\overline{ME}$	Input on DS1-5		0ns			Note 2
t_{OE}—$\overline{ME}$, SC & WC		$\overline{ME}$, SC, WC	I/O bus			25ns	Refer to Test Loading Circuit #3.
t_{OD}—$\overline{ME}$, SC & WC		$\overline{ME}$, SC, WC	I/O bus (three-state)			30ns	
t_w (MCLK pulse width)		MCLK		45ns			
t_{SD} (data setup time)	MCLK	I/O bus		50ns			
t_{SD} ($\overline{ME}$ setup time)	MCLK	$\overline{ME}$		45ns			
t_{SD} (SC setup time)	MCLK	SC		45ns			
t_{SD} (WC setup time)	MCLK	WC		45ns			
t_{HD} (data hold time)	MCLK	I/O bus		0ns			

AC CHARACTERISTICS (Cont'd) V_{CC} = 5V (±5%), T_A = 0°C to 70°C

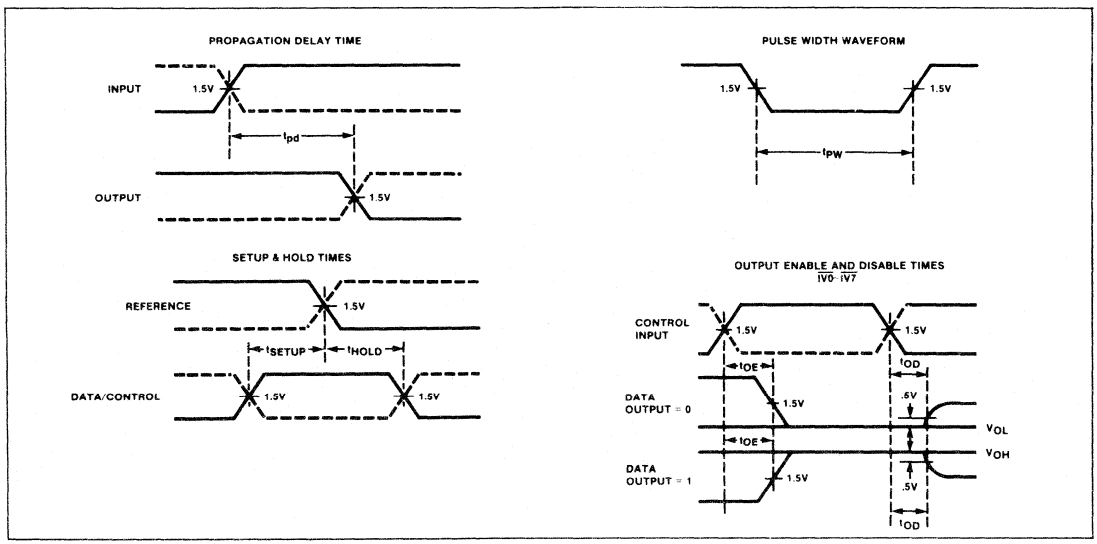
MNEMONIC	REFERENCE	INPUT	OUTPUT	Min	Typ	Max	COMMENTS
t_{HD} ($\overline{ME}$ hold time)	MCLK	$\overline{ME}$		0ns			
t_{HD} (SC hold time)	MCLK	SC		0ns			
t_{HD} (WC hold time)	MCLK	WC		0ns			

NOTES
1. Write pulse width = $2/F_{XTAL}$, that is, for 8MHz crystal, t_{pw} = 250nsec (typical)
2. Changes on DS1-5 are not stored in read mode ($\overline{ME}$ = 0, SC = 0, and WC = 0)
3. During the period when MCLK is high, measurement is made with $\overline{ME}$ = Low, SC = Low, and WC = High.

TEST LOADING CIRCUIT



TIMING DIAGRAM



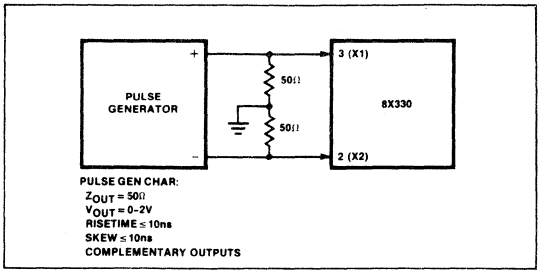
CLOCK REQUIREMENTS

Crystal Oscillator. The on-chip crystal oscillator circuit is designed for operation using an external series-resonant quartz crystal; alternately the crystal oscillator can be driven with complementary outputs of a pulse generator or interfaced to a master clock source via TTL logic—see accompanying circuits. When a crystal is used, the on-chip oscillator operates at the resonant frequency (f.) of the crystal; the crystal connects to the 8X330 via pins 3 (X1) and 2 (X2). The lead lengths of the crystal should be approximately equal and as short as possible; also, avoid close proximity to all potential noise sources. The crystal should be hermetically sealed (HC type can) and have the following electrical characteristics:

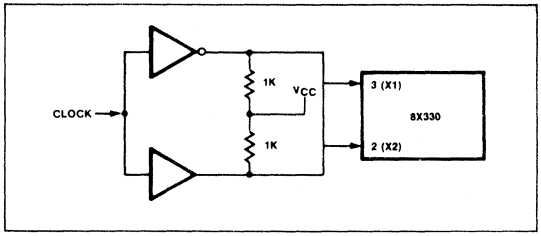
Type: Fundamental mode, series resonant
Impedance at Fundamental: 35-ohms maximum
Impedance at Harmonics and Spurs: 50-ohms minimum

When the crystal oscillator is externally-driven, typical waveforms are as follows:

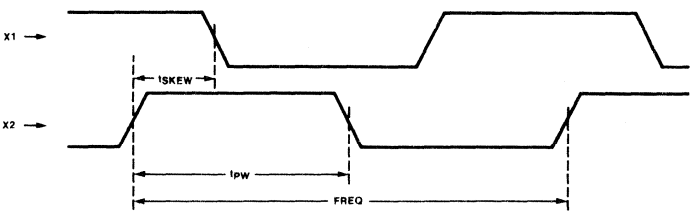
CLOCKING XTAL OSC WITH PULSE GEN



CLOCKING XTAL OSC WITH OPEN-COLLECTOR TTL



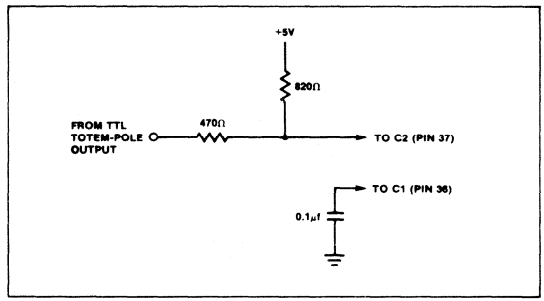
TYPICAL WAVE FORM



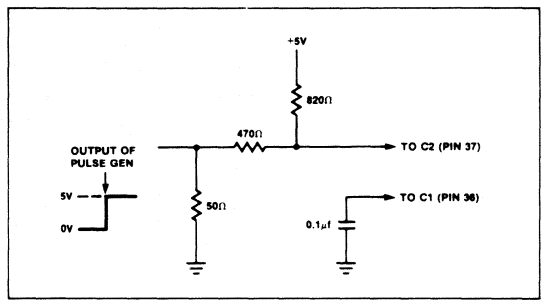
FREQUENCY = 8MHz (TYP)/16MHz (MAX)
t_{PW} = 50% (TYP)
t_{SKEW} ≤ 10ns

Current-Controlled Oscillator (CCO).
A non-polarized ceramic or mica capacitor is recommended for the current-controlled oscillator. The capacitor connects to the 8X330 via pins 37 (C2) and 36 (C1); lead lengths of the capacitor should be approximately the same and as short as possible. When the input current to the CCO is near zero (maximum frequency), the capacitor value should be chosen so that the high-limit rest frequency of the oscillator does not exceed 24 MHz. If the rest frequency is higher than 24 MHz, synchronization of the CCO with the crystal oscillator just prior to the read operation, may be impeded. The curves in Figure 9 (current-versus-frequency) and Figure 8 (capacitance-versus-frequency) show how these design parameters affect operation of the CCO over a temperature range of 0° C to 70° C. A suitable test circuit for verification/validation of the current-controlled oscillator is also shown in Figure 10. Like the crystal oscillator, the CCO can be driven with the TTL output of a pulse generator or interfaced to a master clock via TTL logic—see accompanying diagrams.

CLOCKING WITH OPEN-COLLECTOR TTL



CLOCKING WITH PULSE GENERATOR



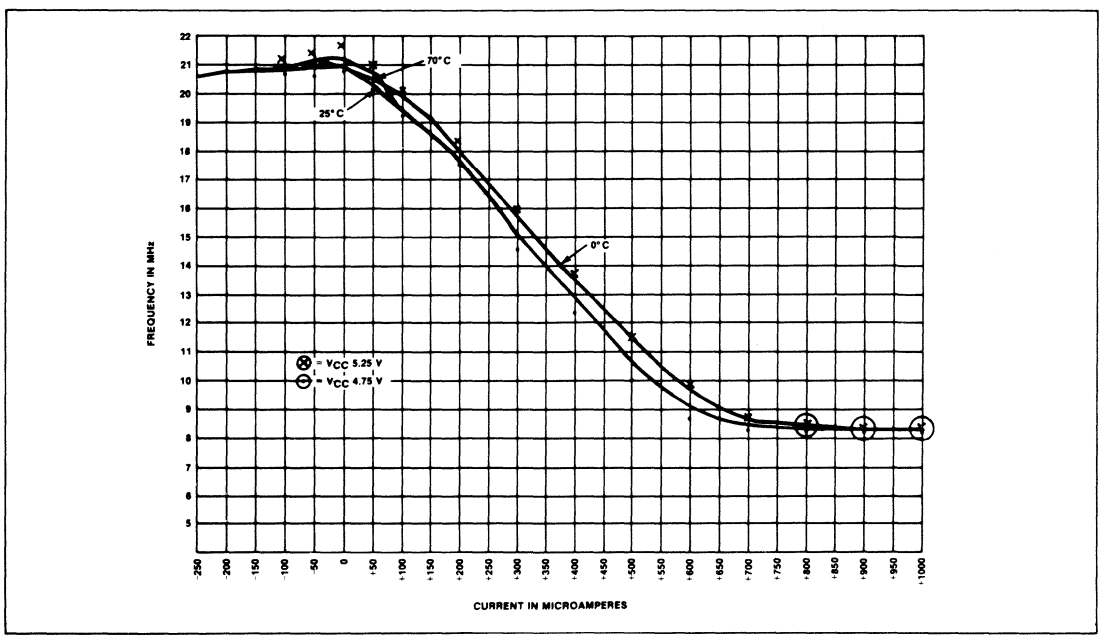


Figure 9. Current-versus-Frequency with: V_{CC} = 5V and Capacitance = 25 Picofarads

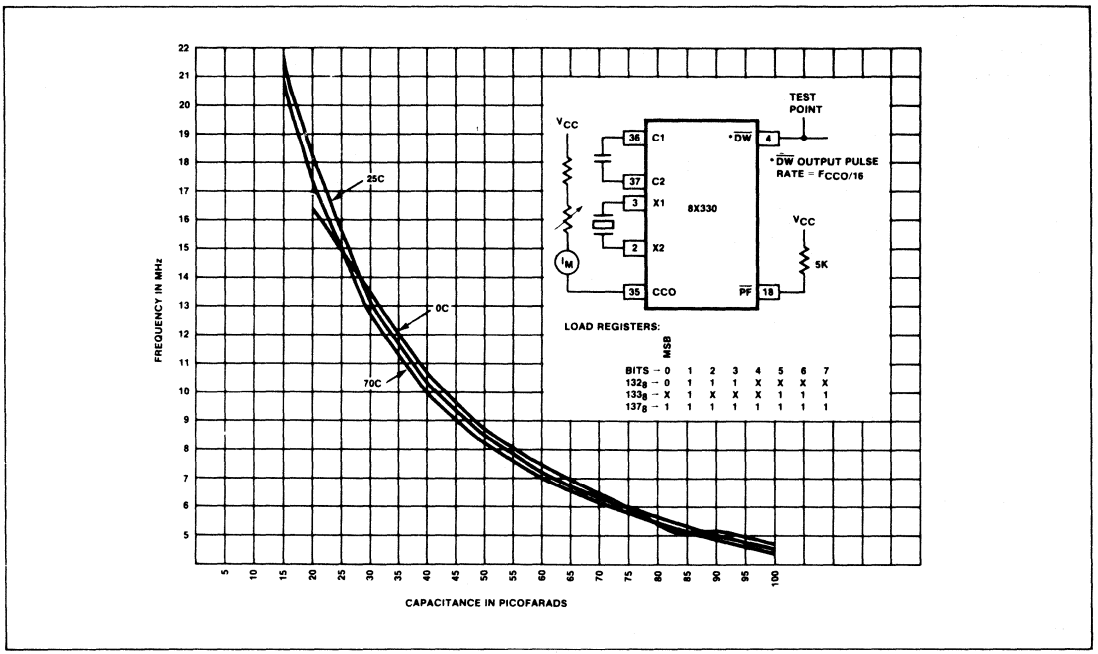


Figure 10. Capacitance-versus-Frequency with: V_{CC} = 5V, VCR = 2.5V, and I = 300μA

VOLTAGE REGULATOR

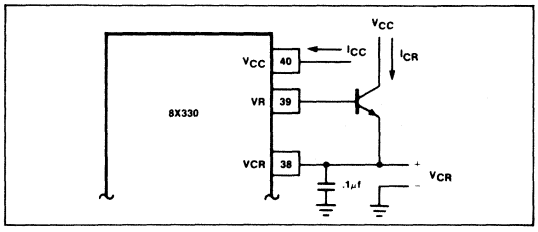
All internal logic of the 8X330 is powered by an on-chip voltage regulator that requires an external series-pass transistor. Electrical specifications for the off-chip power transistor and a typical hook-up are shown in accompanying diagrams. To minimize lead inductance, the transistor should be as close as possible to the 8X330 package and the emitter should be ac-grounded via a 0.1-microfarad capacitor.

ELECTRICAL SPECIFICATIONS

*PARAMETER	CONDITIONS	LIMITS
h_{fe}	V_{CE} = 2V	>50
V_{BEON}	V_{CE} = 5V/I_C = 500mA	<1V
V_{CESAT}	I_C = 500 mA/I_B = 50 mA	<0.5V
BV_{CEO}		>8V
f_t		>30 MHz

*Medium power NPN silicon (0°<T_A<70°C) recommended parts: 2N5320, 2N5337

TYPICAL HOOK-UP



2048-BIT BIPOLAR RAM (256×8)

Originally published by Signetics January 1984

FEATURES

- **On-chip address latches**
- **3-state outputs**
- **Schottky clamped TTL**
- **Internal control logic for 8X300 system**
- **Directly interfaces with the 8X300 bipolar microprocessor with no external logic**
- **May be used on left or right bank**

APPLICATIONS

- **8X300 or 8X305 working storage**

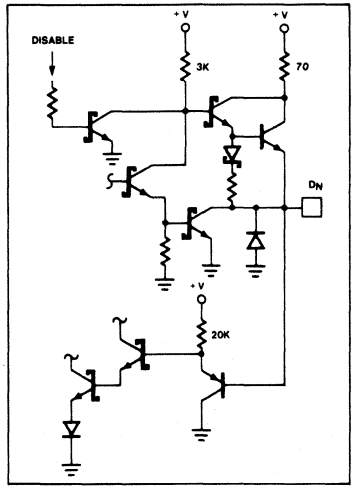
DESCRIPTION

The 8X350 bipolar RAM is designed principally as a working storage element in an 8X300 based system. Internal circuitry is provided for direct use in 8X300 applications. When used with the 8X300, the RAM address and data buses are tied together and connected to the IV bus of the system.

The data inputs and outputs share a common I/O bus with 3-state outputs.

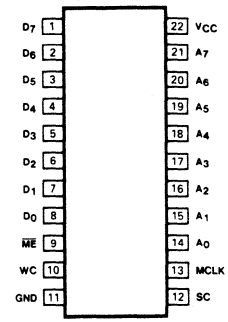
The 8X350 is available in commercial and military temperature ranges. For the commercial temperature range (0°C to +75°C) specify N8X350-F, and for the miltiary temperature range (−55°C to +125°C) specify S8X350-F.

TYPICAL I/O STRUCTURE



PIN CONFIGURATION

F PACKAGE



TOP VIEW

F = Cerdip

ORDER NUMBERS

8X350N, 8X350F
S8X350F/883B

ABSOLUTE MAXIMUM RATINGS

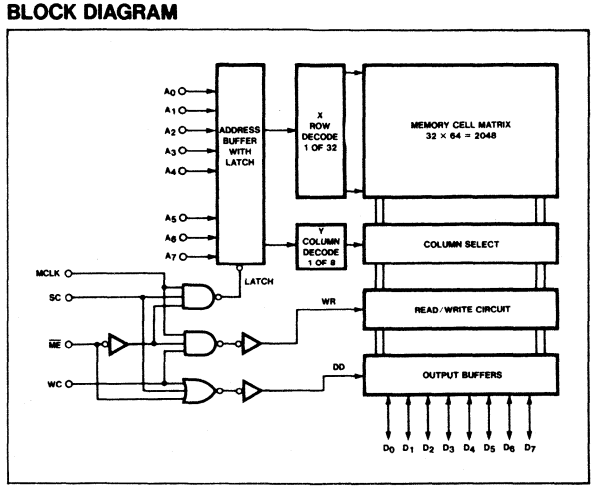
PARAMETER		RATING	UNIT
V_{CC}	Supply voltage	+7	Vdc
V_{IN}	Input voltage	+5.5	Vdc
	Output voltage		Vdc
V_{OH}	High	+5.5	
V_O	Off-state	+5.5	
	Temperature range		°C
T_A	Operating		
	Commercial	0 to +75	
	Military	−55 to +125	
T_{STG}	Storage	−65 to +150	

TRUTH TABLE

Note X = Don't care

MODE	$\overline{ME}$	SC	WC	MCLK	BUSSED DATA/ADDRESS LINES
Hold address Disable data out	1	X	X	X	High Z data out
Input new address	0	1	0	1	Address High Z
Hold address Disable data out	0	1	0	0	High Z data out
Hold address Write data	0	0	1	1	Data in
Hold address Disable data out	0	0	1	0	High Z data out
Hold address Read data	0	0	0	X	Data out
Undefined state[12]	0	1	1	1	—
Hold address[12] Disable data out	0	1	1	0	High Z data out

BLOCK DIAGRAM



DC ELECTRICAL CHARACTERISTICS[2]

N8X350: $0°C \leq T_A \leq +75°C$, $4.75V \leq V_{CC} \leq 5.25V$
S8X350: $-55°C \leq T_A \leq +125°C$, $4.75V \leq V_{CC} \leq 5.25V$

PARAMETER		TEST CONDITIONS	N8X350			S8X350			UNIT
			Min	Typ	Max	Min	Typ	Max	
	Input voltage								V
V_{IL}	Low[1]	V_{CC} = Min			.85			.80	
V_{IH}	High[1]	V_{CC} = Max	2.0			2.0			
V_{IC}	Clamp[1,3]	V_{CC} = Min, I_{IN} = −12mA			−1.2			−1.2	
	Output voltage	V_{CC} = Min							V
V_{OL}	Low[1,4]	I_{OL} = 9.6mA			0.5			.5	
V_{OH}	High[1,5]	I_{OH} = −2mA	2.4			2.4			
	Input current								μA
I_{IL}	Low	V_{IN} = 0.45V			−100			−150	
I_{IH}	High	V_{IN} = 5.5V			25			50	
	Output current								μA
$I_{O(OFF)}$	High Z state	$\overline{ME}$ = High, V_{OUT} = 5.5 V			40			60	μA
		$\overline{ME}$ = High, V_{OUT} = 0.5 V			−100			−100	
I_{OS}	Short circuit[3,6]	SC = WC, $\overline{ME}$ = Low, V_{OUT} = 0V, Stored High	−20		−70	−15		−85	mA
I_{CC}	V_{CC} supply current[7]	V_{CC} = Max			185			200	mA
	Capacitance	$\overline{ME}$ = High, V_{CC} = 5.0V							pF
C_{IN}	Input	V_{IN} = 2.0V		5			5		
C_{OUT}	Output	V_{OUT} = 2.0V		8			8		

AC ELECTRICAL CHARACTERISTICS[2,9]

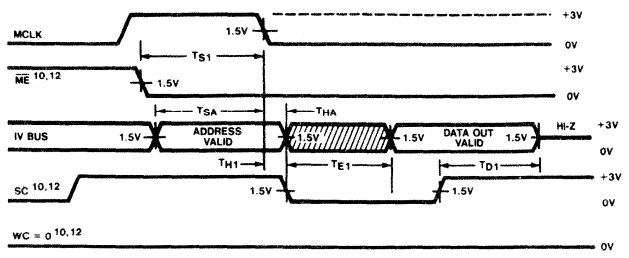
N8X350: $0°C \leq T_A \leq +75°C$, $4.75V \leq V_{CC} \leq 5.25V$ $R_1 = 470\Omega$, $R_2 = 1k\Omega$, $C_L = 30pF$
S8X350: $-55°C \leq T_A \leq +125°C$, $4.75V \leq V_{CC} \leq 5.25V$

PARAMETER		TO	FROM	N8X350			S8X350			UNIT
				Min	Typ	Max	Min	Typ	Max	
	Enable time									ns
T_{E1}	Output	Data out	SC−			35			40	
T_{E2}	Output	Data out	$\overline{ME}$−			35			40	
	Disable time									ns
T_{D1}	Output	Data out	SC+			35			40	
T_{D2}	Output	Data out	$\overline{ME}$+			35			40	
	Pulse width									ns
T_W	Master clock[8]			40			50			
	Setup and hold time									ns
T_{SA}	Setup time	MCLK−	Address	30			40			
T_{HA}	Hold time	Address	MCLK−	5			10			
T_{SD}	Setup time	MCLK−	Data in	35			45			
T_{HD}	Hold time	Data in	MCLK−	5			10			
T_{S3}	Setup time	MCLK−	$\overline{ME}$−	40			50			
T_{H3}	Hold time	$\overline{ME}$+	MCLK−	5			5			
T_{S1}	Setup time	MCLK−	$\overline{ME}$−	30			40			
T_{H2}	Hold time	$\overline{ME}$−	MCLK−	5			5			
T_{S2}	Setup time	$\overline{ME}$−	SC−,WC−	0			5			
T_{H1}	Hold time	SC−	MCLK−	5			5			
T_{H4}	Hold time	WC−	MCLK−	5			5			

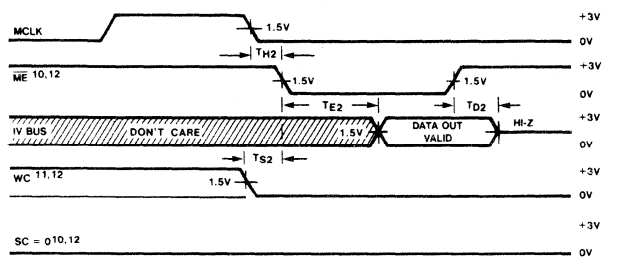
Notes on following page.

TIMING DIAGRAMS

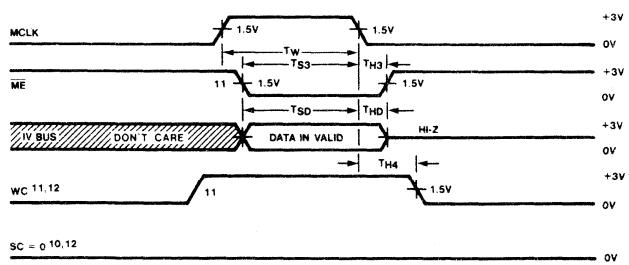
TRANSPARENT ADDRESS READ CYCLE



LATCHED ADDRESS READ CYCLE



LATCHED ADDRESS WRITE CYCLE



NOTES

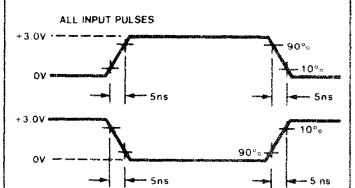
1. All voltage values are with respect to network ground terminal.
2. The operating ambient temperature ranges are guaranteed with transverse air flow exceeding 400 linear feet per minute and a 2-minute warm-up.
   Typical thermal resistance values of the package at maximum temperature are:
   Θ_{JA} junction to ambient at 400fpm air flow - 50°C/watt
   Θ_{JA} junction to ambient - still air - 90°C/watt
   Θ_{JA} junction to case - 20°C/watt
3. Test each pin one at a time.
4. Measured with a logic low stored Output sink current is supplied through a resistor to V_{CC}.
5. Measured with a logic high stored.
6. Duration of the short circuit should not exceed 1 second.
7. I_{CC} is measured with the write enable and memory enable inputs grounded, all other inputs at 4.5V and the output open.
8. Minimum required to guarantee a Write into the slowest bit.
9. Applied to the 8X300 based system with the data and address pins tied to the IV Bus.
10. SC + $\overline{ME}$ = 1 to avoid bus conflict.
11. WC + $\overline{ME}$ = 1 to avoid bus conflict.
12. The SC and WC outputs from the 8X300 are never at 1 simultaneously.

TIMING DEFINITIONS

T_{S1}	Required delay between beginning of Master Enable low and falling edge of Master Clock.
T_{SA}	Required delay between beginning of valid address and falling edge of Master Clock.
T_{HA}	Required delay between falling edge of Master Clock and end of valid Address.
T_{H1}	Required delay between falling edge of Master Clock and when Select Command becomes low.
T_{E1}	Delay between beginning of Select Command low and beginning of valid data output on the IV Bus.
T_{D1}	Delay between when select Command becomes high and end of valid data output on the IV Bus.
T_{H2}	Required delay between falling edge of Master Clock and when Master Enable becomes low.
T_{E2}	Delay between when Master Enable becomes low and beginning of valid data output on the IV Bus.
T_{D2}	Delay between when Master Enable becomes high and end of valid data output on the IV Bus.
T_{S2}	Required delay between when Select Command or Write Command becomes low and when Master Enable becomes low.
T_W	Minimum width of the Master Clock pulse.
T_{S3}	Required delay between when Master Enable becomes low and falling edge of Master Clock.
T_{H3}	Required delay between falling edge of Master Clock and when Master Enable becomes high.
T_{SD}	Required delay between beginning of valid data input on the IV Bus and falling edge of Master Clock.
T_{HD}	Required delay between falling edge of Master Clock and end of valid data input on the IV Bus.
T_{H4}	Required delay between falling edge of Master Clock and when Write Command becomes low.

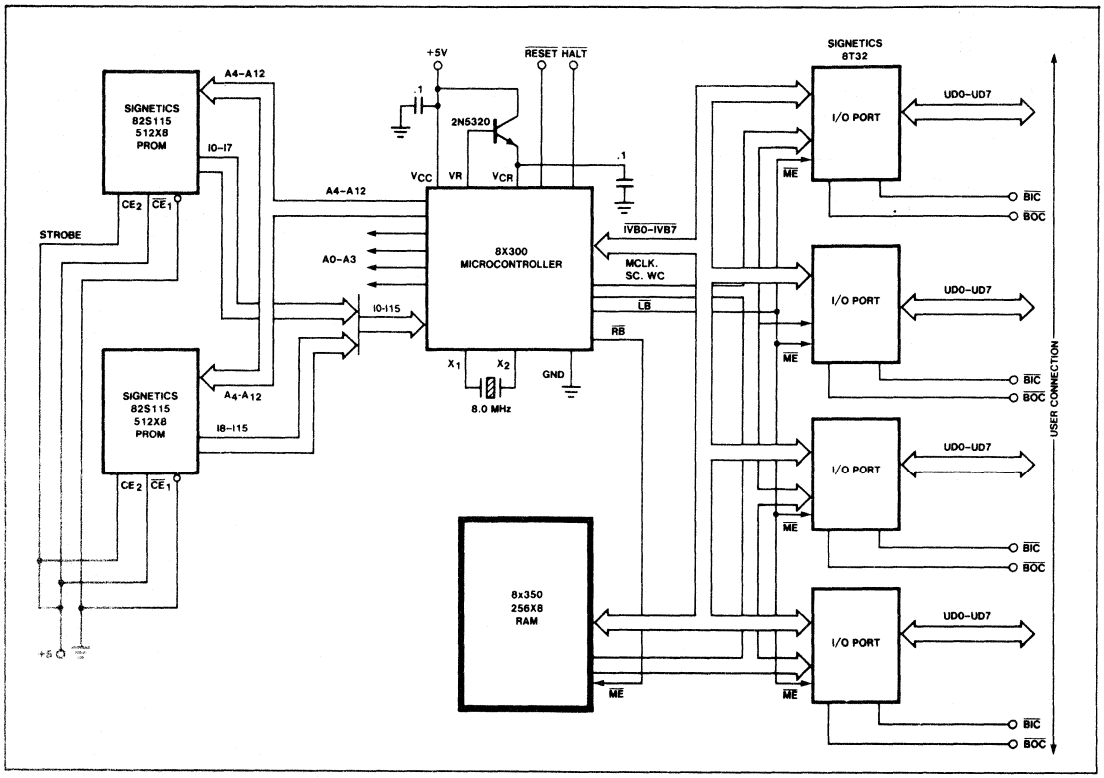
VOLTAGE WAVEFORM

INPUT PULSES

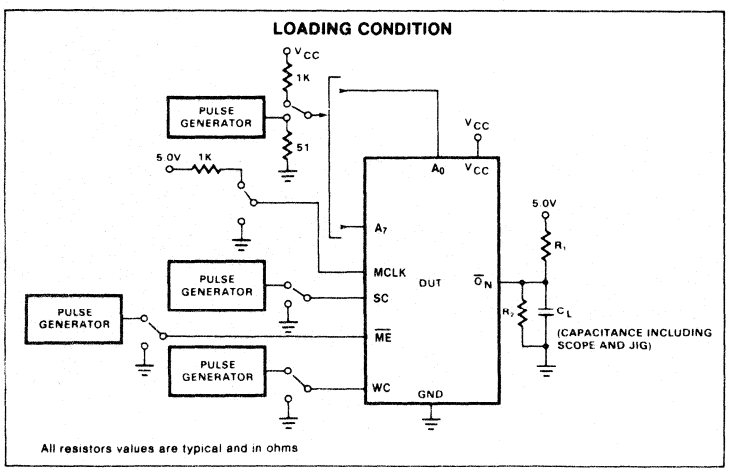


Measurements: All circuit delays are measured at the +1.5V level of inputs and output.

TYPICAL 8X350 APPLICATION



TEST LOAD CIRCUIT



Bipolar Ram (32X8)

Originally published by Signetics January 1985

Product Specification

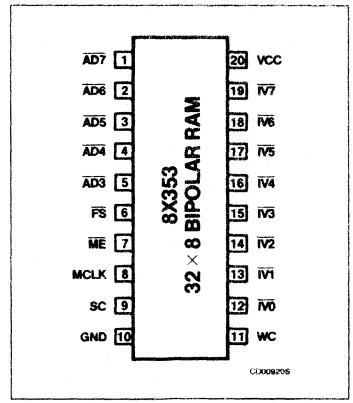
DESCRIPTION

The 8X353 is a 32-byte RAM designed principally as a working storage element in 8X305-based systems. The 8X353 is ideal for applications requiring a relatively small amount of data storage and maximum I/O flexibility. Since the 8X353 takes up only 32 of 256 locations on a controller bank, this device allows single cycle, bank-to-bank data transfer and other implementations in which the user does not wish to dedicate an entire I/O bank to data storage. Contributing to the versatility of the 8X353 is a fast select feature which, when supplemented with extended micro-code, allows this device to be selected externally of the IV bus.

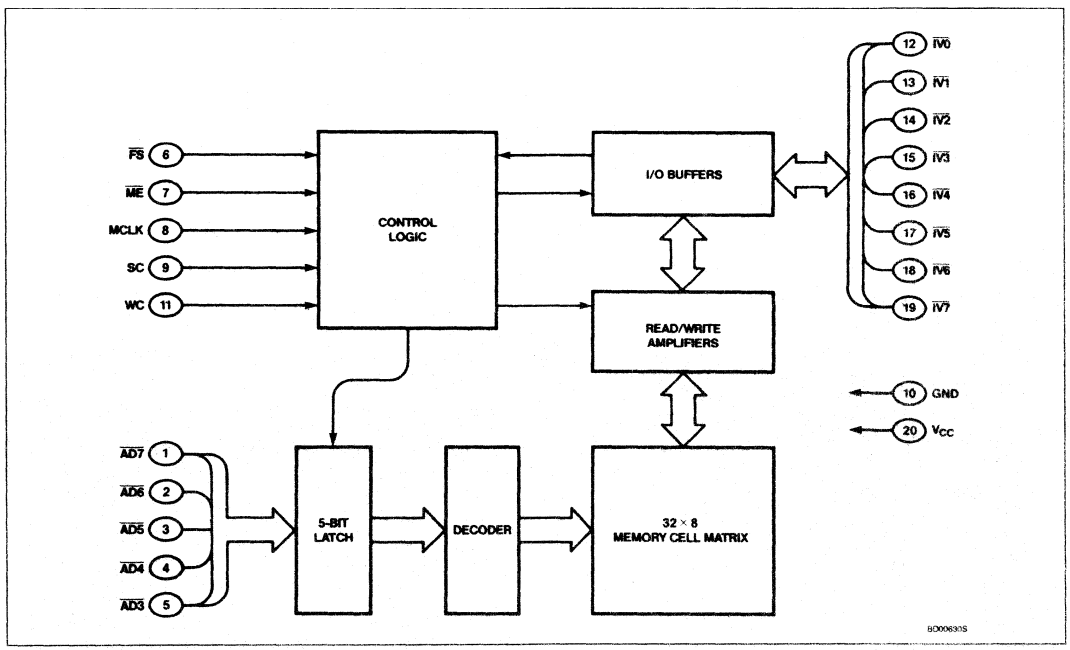
FEATURES

- **32 bytes of storage**
- **32 port addresses ($00_8 - 37_8$)**
- **224 bank addresses available for other 8X305 peripherals**
- **Fast Select feature for use with extended micro-code**
- **On-chip address decoding**
- **Separate address input pins**
- **Direct interface**
- **Single 5V power supply**
- **0.3 inch, Slim Line package**
- **0.3 inch, CERDIP package**

PIN CONFIGURATION

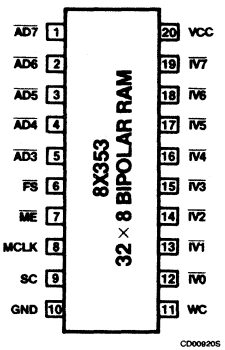


BLOCK DIAGRAM



Product Specification

PIN DESCRIPTION



PIN NO.	IDENTIFIER	FUNCTION
1 – 5	$\overline{AD7}$ – $\overline{AD3}$	ADdress bus – Active low inputs that define which memory locations are to be accessed.
6	$\overline{FS}$	Fast Select – When active low, this input allows read and write operations without performing a full select cycle.
7	$\overline{ME}$	Master Enable – When low, this input indicates the microcontroller bank is active; otherwise the chip will not respond to any signal on the IV bus.
8	MCLK	Master CLocK – A standard 8X305 clock input used for timing reference and system synchronization.
9	SC	Select Command – When active high, this input indicates IV bus data is to be interpreted as an address.
10	GND	GrouND.
11	WC	Write Command – When active high, this signal indicates data is being input from the IV bus.
12 – 19	$\overline{IV0}$ – $\overline{IV7}$	Interface Vector Bus – Three-state, active low, bidirectional signals which communicate I/O data and addresses (chip select).
20	V_{CC}	Supply Voltage.

Product Specification

FUNCTIONAL OPERATION

The 8X353 is capable of performing two functions — write and read. These operations, and the control logic necessary for their execution, are described in the following text and summarized in Table 1. Typical timing relationships for write, read, and select cycles are shown in Timing Diagrams.

Select Logic

Standard Select Cycle — Before a read or write can occur, the 8X353 must first be selected by either a standard select cycle or by Fast Select operation (see below). A standard select cycle is initiated when SC and MCLK are high, and $\overline{ME}$ and WC are low. The three most significant bits on the $\overline{IV}$ bus are then compared with fixed internal values (110_2 active low), and if identical, an internal select latch is activated (high) and the five bits of the memory address (pins AD7 – AD3) are loaded into an address latch.

As shown in Table 1, the 8X353 remains in the selected state until another select cycle is initiated, at which point the select latch and addresses are updated.

In situations where more than one RAM is required, the fixed internal values of IV0, IV1, and IV2 can be used as chip addresses. By scrambling these pins, up to three RAMs can be selected from a single IV bus.

Write/Read Operation

Write and read operations are possible only after the 8X353 has been selected by either a standard select cycle or by Fast Select operation. As shown in Table 1, data I/O is controlled by $\overline{ME}$, SC, WC, MCLK, FS, and the current state of the select latch.

After a data address has been latched, data is written to the specified memory location when MCLK and WC are high and $\overline{ME}$ is low; a read occurs when MCLK, WC, SC, and $\overline{ME}$ are low. In both cases, data is transmitted via the IV bus.

Table 1. Summary Of 8X353 Operation

MODE	CONDITIONS						RESULTS				
	$\overline{ME}$	SC	WC	MCLK	$\overline{FS}$	Select Latch	$\overline{IV}$ Bus	$\overline{AD}$ Bus	Data	Address Latch	Select Latch
A_{UF}	L	H	L	H	X	X	Input (chip address)	Input[2]	Keep	Update[2]	Update[2]
D_{IS}	L	L	H	H	H	H	Input	Ignore	Update	Keep	Keep
D_{OS}	L	L	L	L	H	H	Output	Ignore	Keep	Keep	Keep
A_{UF}	L	S	L	H	L	X	Ignore	Input	Keep	Update	Keep
A_{UF}	H	X	X	H	L	X	Ignore	Input	Keep	Update	Keep
D_{IF}	L	L	H	H	L	X	Input	Ignore	Update	Keep	Keep
D_{OF}	L	L	L	L	L	X	Output	Ignore	Keep	Keep	Keep
$\overline{ME}$ High	H	X	X	X	H	X	Ignore	Ignore	Keep	Keep	Keep
$\overline{ME}$ High	H	X	X	L	L	X	Ignore	Ignore	Keep	Keep	Keep
D_{OS} w/o MCLK Low	L	L	L	H	H	X	Ignore	Ignore	Keep	Keep	Keep
D_{OS} w/o select latch High	L	L	L	L	H	L	Ignore	Ignore	Keep	Keep	Keep
D_{IS} w/o MCLK High	L	L	H	L	X	X	Ignore	Ignore	Keep	Keep	Keep
D_{IS} w/o select latch High	L	L	H	H	H	L	Ignore	Ignore	Keep	Keep	Keep
A_{US} w/o MCLK High	L	H	L	L	X	X	Ignore	Ignore	Keep	Keep	Keep
Not defined	X	H	H	X	X	X			Not defined		

NOTES:

1. A_{US} = address update with standard select
   D_{IS} = data input with standard select
   D_{OS} = data output with standard select
   A_{UF} = address update with fast select
   D_{IF} = data input with fast select
   D_{OF} = data output with fast select
2. Depending on IV bus data
3. Fast select Logic ($\overline{FS}$) given but is not used for 8X305 operation.
   X = Don't care

ABSOLUTE MAXIMUM RATINGS

PARAMETER		RATING	UNIT
V_{CC}	Power supply voltage	+7	V dc
V_{IN}	Input voltage	+5.5	V dc
V_O	Off-state voltage	+5.5	V dc
T_{STG}	Storage temperature range	−65 to +150	C

DC ELECTRICAL CHARACTERISTICS $V_{CC} = 5.0 \pm 5\%$, $0°C \leq T_A \leq 70°C$

PARAMETER		TEST CONDITIONS[1]	LIMITS			UNIT
			Min	Typ	Max	
V_{IH}	High level input voltage		2			V
V_{IL}	Low level input voltage				0.8	V
V_{OH}	High level output voltage	IV pins: V_{CC} = Min, I_{OH} = −3.2mA	2.4			V
V_{OL}	Low level output voltage	IV pins : V_{CC} = Min, I_{OL} = 16mA			0.55	V
V_{IC}	Input clamp voltage	V_{CC} = Min, I_{IN} = −10mA			−1.0	V
I_{IH}	High level input (or leakage) current	Address pins and control pins: V_{CC} = Max, V_{IN} = 4.5V			100	μA
I_{IL}	Low level input current	Address pins and control pins: V_{CC} = Max, V_{IN} = .5V			−100	μA
I_{OS}	Short circuit output current	IV pins: V_{CC} = Max[2]	−30		−140	mA
I_{OZH}	High-Z state output current – high level	IV pins: V_{CC} = Max, V_0 = 2.7V			100	μA
I_{OZL}	High-Z state output current – low level	IV pins: V_{CC} = Max, V_0 = .5V			−200	μA
I_{CC}	Supply current	All inputs: V_{CC} = Max, V_{IN} = 0V			200	mA

NOTES:
1. All voltages measured with respect to Ground terminal.
2. At any time, no more than one output should be connected to ground.

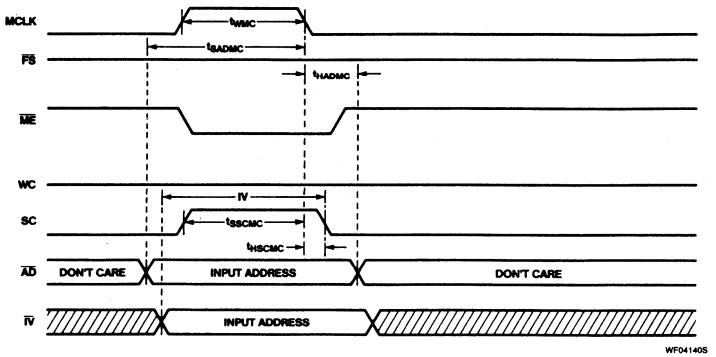
Product Specification

AC ELECTRICAL CHARACTERISTICS $V_{CC} = 5.0 \pm 5\%$, $0°C \leq T_A \leq 70°C$

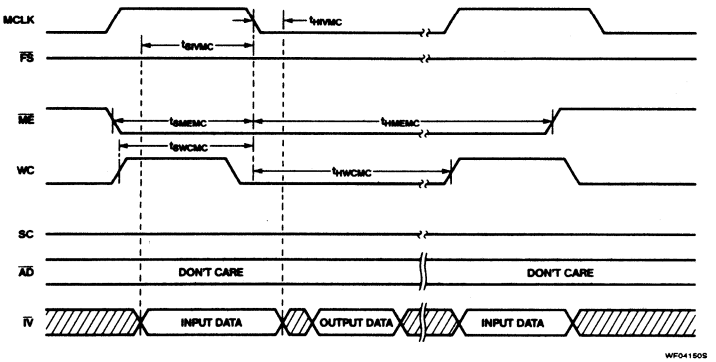
PARAMETER	REFERENCES		TEST CONDITIONS[1,2,3]	LIMITS			UNIT
	From	To		Min	Typ	Max	
Pulse width							
t_{WMC} MCLK pulse width	↑MCLK	↓MCLK	FS = High;$\overline{ME}$ = Low	35			ns
Set-up times							
t_{SIVMC} IV set-up time	$\overline{IV}$	↓MCLK	FS = High;$\overline{ME}$ = Low	40			ns
t_{SADMC} Address set-up time	$\overline{AD}$	↓MCLK	FS, SC = High; WC,$\overline{ME}$ = Low	40			ns
t_{SWCMC} Write command set-up time	↑WC	↓MCLK	FS = High; SC,$\overline{ME}$ = Low	60			ns
t_{SSCMC} Select command set-up time	↑SC	↓MCLK	FS = High; WC,$\overline{ME}$ = Low	55			ns
t_{SMEMC} Master enable set-up time	↓$\overline{ME}$	↓MCLK	FS = High	55			ns
Hold times							
t_{HIVMC} IV hold time	↓MCLK	$\overline{IV}$	FS = High;$\overline{ME}$ = Low	5			ns
t_{HADMC} Address hold time	↓MCLK	$\overline{AD}$	FS, SC = High; WC,$\overline{ME}$ = Low	5			ns
t_{HWCMC} Write command hold time	↓MCLK	↓WC	FS = High; SC,$\overline{ME}$ = Low	3			ns
t_{HSCMC} Select command hold time	↓MCLK	↓SC	FS = High; WC,$\overline{ME}$ = Low	0			ns
t_{HMEMC} Master enable hold time	↓MCLK	↑$\overline{ME}$	FS = High; SC, WC = Low	5			ns
Output enable times							
t_{EDOMC} IV output enable time	↓MCLK	$\overline{IV}$	FS = High; WC,$\overline{ME}$, SC = Low			30	ns
t_{EDOWC} IV output enable time	↓WC	$\overline{IV}$	FS = High; MCLK,$\overline{ME}$, SC = Low			30	ns
t_{EDOSC} IV output enable time	↓SC	$\overline{IV}$	FS = High; MCLK, WC,$\overline{ME}$ = Low			30	ns
t_{EDOME} IV output enable time	↓$\overline{ME}$	$\overline{IV}$	FS = High; MCLK, WC, SC = Low			30	ns
Output disable times							
t_{DDOMC} IV output disable time	↑MCLK	$\overline{IV}$	FS = High; WC,$\overline{ME}$, SC = Low[4]			25	ns
t_{DDOWC} IV output disable time	↑WC	$\overline{IV}$	FS = High; MCLK,$\overline{ME}$ = Low[4]			25	ns
t_{DDOSC} IV output disable time	↑SC	$\overline{IV}$	FS = High; MCLK, WC,$\overline{ME}$ = Low[4]			25	ns
t_{DDOME} IV output disable time	↑$\overline{ME}$	$\overline{IV}$	FS = High; MCLK, WC = Low[4]			25	ns

NOTES:
1. Loading: see Test Loading Circuits
2. All inputs are driven between 0 and 3.0 volts.
3. Except for output disable times, all timing parameters are measured at 1.5 volts.
4. These parameters are measured with a capacitive loading of 50pF and represent the output driver turn-off time. Disable times measured into a capacitive loading of 300pF will be a maximum of 40ns.

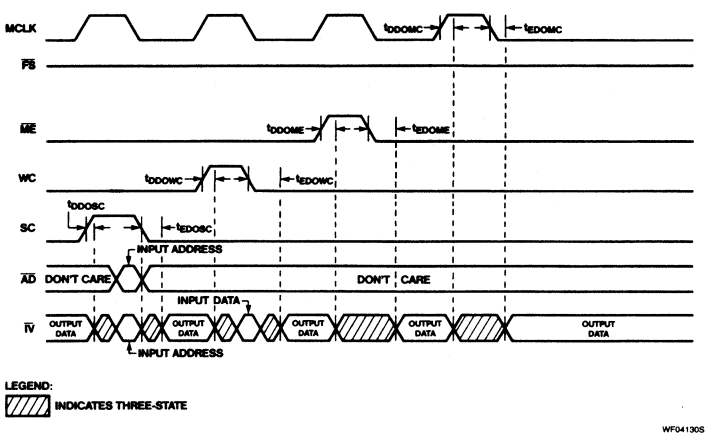
TIMING DIAGRAM



a. Standard Select Cycle Timing

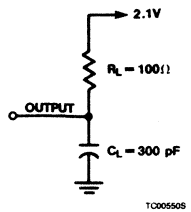


b. Write/Read Cycle Timing



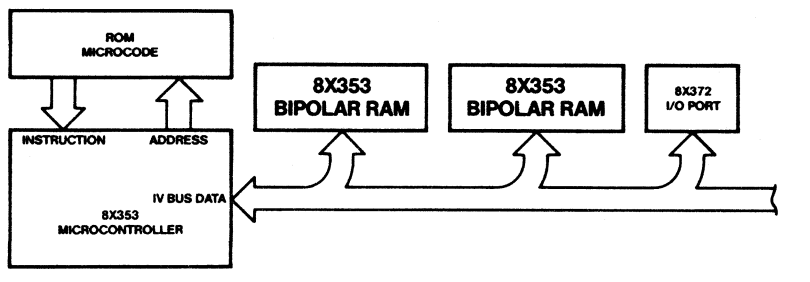
c. Output Enable/Disable Timing

TEST LOADING CIRCUIT

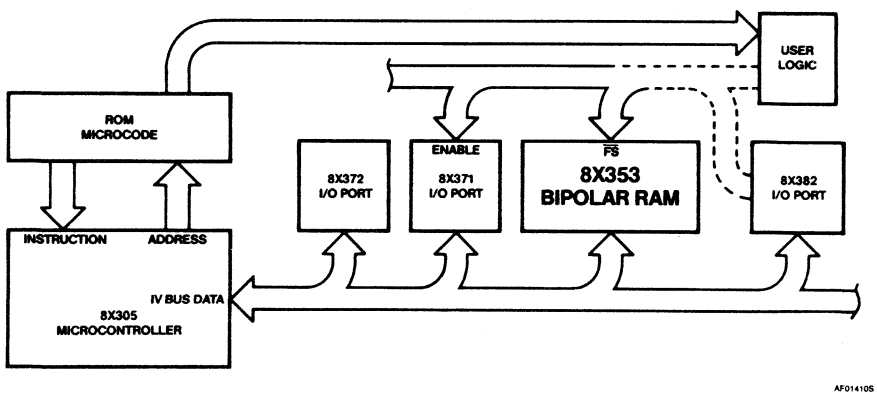


NOTE:
C_L = 50pF for output disable times.

APPLICATIONS DIAGRAMS

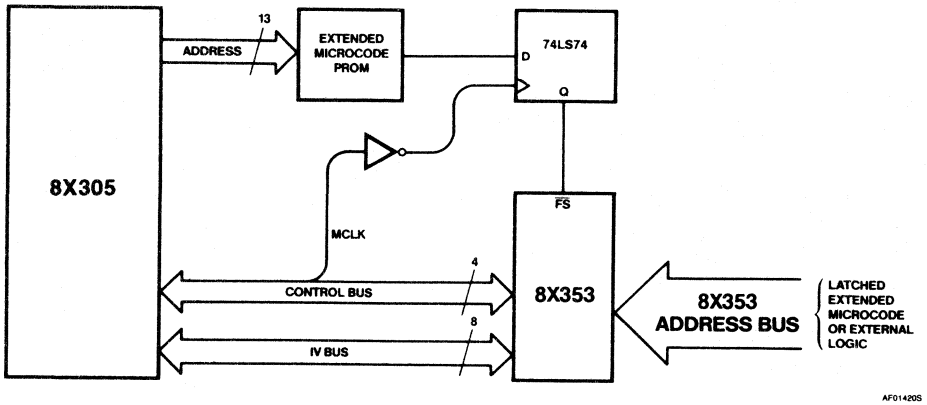


a. Implementation of the 8X353 Using $\overline{IV}$ Bus Select



b. Implementation of the 8X353 Using FAST Select and External Logic

APPLICATIONS DIAGRAM (Continued)



NOTES:
1. The control bus consists of MCLK, SC, WC and $\overline{ME}$.
2. The 8X353 can be selected using the $\overline{FS}$ pin instead of a normal select cycle. Since the flip-flop is clocked by the falling edge of MCLK, the data input to the flip-flop must have sufficient set-up time.

Example Use of the 8X353 Fast Select Mode.

ORDERING INFORMATION

N8X353N (Plastic)
N8X353F (CERDIP)

Originally published by Signetics July 1985

8X355
LIFO Stack Memory (32X8)

Product Specification

DESCRIPTION

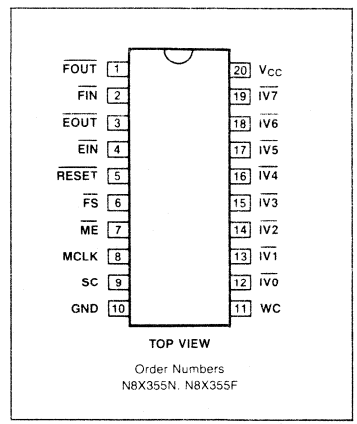
The 8X355 Last In/First Out (LIFO) stack memory is designed for use with the 8X305 Microcontroller; however, it can be easily adapted to other microprocessor-based systems. The 32-byte storage capacity of the chip can be expanded by cascading to form 8-bit wide memories of any required depth. Typical applications are shown at the rear of this data sheet.

The 8X355 can be addressed by using standard $\overline{IV}$ bus control logic or, in high-performance systems, the instruction word of the Microcontroller can be extended and the 8X355 can be addressed in the fast-select mode. In the latter case, an address select cycle prior to each data access is omitted. When used with the 8X310 Interrupt Control Coprocessor, this enhanced throughput feature is very convenient for saving 8X305 internal registers during interrupt servicing. For either single or stack configurations, the 8X355 uses a fixed enabling address (130_8), thus, once enabled, the memory can accept an uninterrupted stream of data.

FEATURES

- **32 bytes of storage**
- **Cascadable LIFO operation**
- **Dedicated port address**
- **Fast select feature for use with extended microcode**
- **Three-state TTL outputs**
- **Single 5-volt power supply**
- **0.3-inch, Slim Line package**

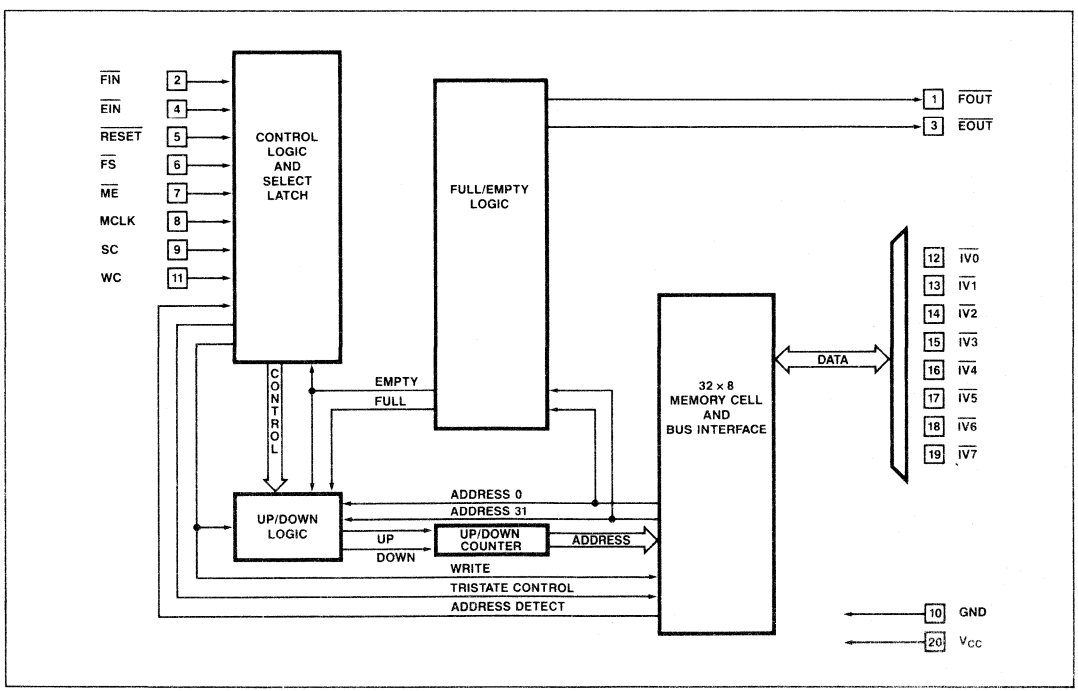
PIN CONFIGURATION



TOP VIEW

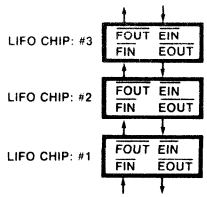
Order Numbers
N8X355N, N8X355F

BLOCK DIAGRAM



Product Specification

CASCADED CONFIGURATION



NOTE: For simplicity, this data sheet refers to the cascaded configuration shown above in which data is stacked from bottom to top; however, this configuration can be inverted (data stacked from top to bottom) without changing the overall logic of the 8X355.

PIN NO.	IDENTIFIER	FUNCTION
1	$\overline{FOUT}$	**Full Output:** When active (low), this output indicates that the 8X355 is full and no further writes can be accepted until at least one read operation is executed. On power up, $\overline{FOUT}$ is high and remains in this inactive state until the last memory location is written to. In a cascaded configuration, $\overline{FOUT}$ is tied to $\overline{FIN}$ of the next higher LIFO chip in the stack.
2	$\overline{FIN}$	**Full Input:** In cascaded configurations, an active (low) $\overline{FIN}$ signal indicates to a given LIFO chip that the next lower chip is full. When $\overline{FIN}$ is inactive, the 8X355 cannot be accessed except for select and deselect cycles. In non-cascaded configurations, $\overline{FIN}$ is usually forced low to enable reads and writes.
3	$\overline{EOUT}$	**Empty Output:** When active (low), this output indicates that the 8X355 is empty, and further read operations cannot be accepted until at least one write operation is executed. $\overline{EOUT}$ is asserted in three situations: on power up, when $\overline{FIN}$ is inhibited, and when a read is performed with the stack pointer set at 0. In a cascaded configuration, $\overline{EOUT}$ is tied to $\overline{EIN}$ of the next lower LIFO chip of the stack.
4	$\overline{EIN}$	**Empty In:** In cascaded configurations, an active (low) $\overline{EIN}$ signal indicates to a given LIFO chip that the next higher chip is empty. When $\overline{EIN}$ is inactive (high), the 8X355 will not respond to read operations; write operations are inhibited by $\overline{FOUT}$ (which is active), while selects and deselects are allowed. In non-cascaded configurations, $\overline{EIN}$ is usually forced low to enable reads and writes.
5	$\overline{RESET}$	**Reset:** When active (low), this input returns the 8X355 to empty, deselected status. In cascaded configurations, RESETs are normally tied together.
6	$\overline{FS}$	**Fast Select:** When active (low), this input temporarily allows read and write operations without performing a full select cycle.
7	$\overline{ME}$	**Master Enable:** When active (low), this input indicates that the controller bank is active; otherwise, the 8X355 will not respond to any signal on the $\overline{IV}$ bus.
8	MCLK	**Master Clock:** A standard 8X355 clock input used for timing reference and system synchronization.
9	SC	**Select Command:** When active (high), this input indicates that data on the $\overline{IV}$ bus is to be interpreted as an address.
10	GND	**Ground**
11	WC	**Write Command:** When active (high), this signal indicates that data is being input from the $\overline{IV}$ bus.
12 to 19	$\overline{IV0}$ to $\overline{IV7}$	**Interface Vector Bus:** Active low, bidirectional, three-state signals which communicate I/O data and addresses.
20	V_{CC}	**Supply Voltage**

Product Specification

FUNCTIONAL OPERATION

The 8X355 performs two fundamental operations: write (push) and read (pop). These functions, along with the cascading capabilities and control logic of the device, are described below. Typical timing relationships for write, read, and select cycles are shown later.

Control Logic

Select Cycle — Before read or write operations can be executed, the 8X355 must be selected by either a standard select cycle or by Fast Select operation. With MCLK high, information entered on the $\overline{IV}$ bus is either an address or input data, depending on the states of SC and WC. An active (high) SC signal indicates that $\overline{IV}$ bus data is to be interpreted as an address; this address is then compared with that of the LIFO chip (130_8). If identical, the select latch becomes active (high) and the chip may be accessed for reads and writes. As shown in Table 1, the 8X355 remains in the selected state until another select cycle is initiated (MCLK = S – C = [high]), at which point the latch is updated.

Fast Select — A Fast Select ($\overline{FS}$) pin is provided which allows the 8X355 to be accessed whether or not a select cycle has been executed with the proper address. Since $\overline{FS}$ is ORed with the output of the select latch, this input has no effect on latch operation. Note that the 8X355 is selected only while $\overline{FS}$ is low, and does not remain selected when $\overline{FS}$ goes high unless a proper select cycle had been performed during the time $\overline{FS}$ was low. When coupled with external logic, this feature makes it possible to quickly push or pop data without using an extra cycle to select the stack peripheral.

Addressing — The 8X355 has a dedicated port address; however, the IV bus pins can be scrambled to obtain different addresses on a given bank. For example, a port designed with the address 130_8 (01011000_2) can be made to respond to the address 230_8 (10011000_2) simply by swapping the connections of the two most significant $\overline{IV}$ bus pins on a LIFO chip. Besides changing the enabling address, this pin swap has no effect on any other chip operation.

Reset — A reset pin is provided which empties and deselects the 8X355. When activated (low), the $\overline{RESET}$ signal sets the stack pointer to 0, asserts $\overline{EOUT}$ (low), negates $\overline{FOUT}$ (high), and negates select latch output.

Write/Read Operations

As shown in Table 1, the select latch, $\overline{IV}$ bus, and stack data are controlled by $\overline{ME}$, WC, SC, MCLK, $\overline{FS}$ and the current state of the select latch.

To perform a read, the 8X355 must be selected, and MCLK, SC, WC and $\overline{ME}$ must be low; in cascaded configurations, the 8X355 must also conform to status signal requirements (see Cascading Capabilities below). With these conditions satisfied, data is output on the $\overline{IV}$ bus and the stack pointer is decremented on the rising edge of MCLK.

A write is performed when MCLK and WC are high, $\overline{ME}$ and SC are low, and the 8X355 is selected. When these conditions (and any status signal conditions) are satisfied, the stack pointer is incremented on the rising edge of the MCLK, after which data is loaded onto the top of the stack.

In single configurations, $\overline{FIN}$ and $\overline{EIN}$ are usually forced low (grounded) to enable reads and writes. In cascaded configurations, the $\overline{FIN}$ signal of the lowest LIFO chip and the $\overline{EIN}$ signal of the highest LIFO chip are similarly grounded.

Cascading Capabilities

To facilitate cascading, the 8X355 provides pins to detect and send signals indicating full or empty status. As described in PACKAGE AND PIN DESIGNATIONS, active $\overline{FOUT}$ and $\overline{EOUT}$ signals indicate, respectively, full and empty status. $\overline{FIN}$ is then defined by $\overline{FOUT}$ of the LIFO chip below, and $\overline{EIN}$ is defined as $\overline{EOUT}$ of the LIFO chip above. Thus data is stacked continuously from bottom to top unless provisions are made with external logic and Fast Select.

Empty status signals are updated on the rising edge of MCLK, while full status signals are updated on the falling edge. As described in Table 2, these signals control read and write operations of the 8X355 — note that LIFO chip #2 in the given configuration is used as the point of reference.

Table 1. General Operations

CONDITIONS						RESULTS		
$\overline{ME}$	SC	WC	MCLK	$\overline{FS}$	Select Latch	$\overline{IV}$ Bus	Data	Select Latch
H	X	X	X	X	X	Ignore	Keep	Keep
L	L	L	L	L	X[1]	Output	Pop	Keep
L	L	L	L	H	L	Ignore	Keep	Keep
L	L	L	L	H	H[1]	Output	Pop	Keep
L	L	L	H	X	X	Not Defined		
L	L	H	L	X	X	Ignore	Keep	Keep
L	L	H	H	L	X[2]	Input	Push	Keep
L	L	H	H	H	L	Ignore	Keep	Keep
L	L	H	H	H	H[2]	Input	Push	Keep
L	H	L	L	X	X	Ignore	Keep	Keep
L	H	L	H	X	X	Input (Address)	Keep	Update
X	H	H	X	X	X	Not Defined		

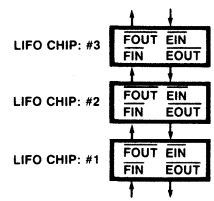
NOTES:
1. If not empty
2. If not full
X Don't Care

Product Specification

Table 2. Status Signal Operations in a Typical Cascaded Configuration

CONDITIONS				RESULTS				COMMENTS
$\overline{EIN}$	$\overline{FIN}$	$\overline{EOUT}$	$\overline{FOUT}$	$\overline{EOUT}$	$\overline{FOUT}$	Push	Pop	
X	X	L	L	Not Possible				Empty and full status
L	L	L	H	Update	H	Yes	No	Top of data stack at top of #1
L	L	H	L	H	Update	No	Yes	Top of data stack at top of #2
L	L	H	H	Update	Update	Yes	Yes	Top of data stack within #2
X	H	X	X	L	H	No	No	Top of data stack below #2
H	L	X	H	Not Defined				#2 not full and data in #3
H	L	H	L	H	L	No	No	Top of data stack above #2

NOTES:
Status signals and Push/Pop operations reference LIFO chip #2.
X = Don't Care



DC ELECTRICAL CHARACTERISTICS (V_{CC} = 5V ±5%/0°C ≤ T_A ≤ 70°C)

ABSOLUTE MAXIMUM RATINGS

SYMBOL	PARAMETER	RATING	UNIT
V_{CC}	Power supply voltage	+7.0	V_{dc}
V_{IN}	Input voltage	+5.5	V_{dc}
V_O	Off-state voltage	+5.5	V_{dc}
T_{STG}	Storage temperature range	−65 to 150	°C

DESIGN RATINGS/TEST CONDITIONS

SYMBOL	PARAMETER	TEST CONDITIONS[1]	LIMITS			UNIT
			Min	Typ	Max	
V_{IH}	High level input voltage		2			V
V_{IL}	Low level input voltage				0.8	V
V_{OH}	High level output voltage	$\overline{IV}$ pins: V_{CC} = Min/I_{OH} = −3.2mA	2.4			V
V_{OL}	Low level output voltage	$\overline{IV}$ pins: V_{CC} = Min/I_{OL} = 16mA $\overline{EOUT}$ & $\overline{FOUT}$: V_{CC} = Min/I_{OL} = 8mA			0.55 0.55	V V
V_{IC}	Input clamp voltage	V_{CC} = Min/I_{IN} = −10mA			−1	V
I_{IH}	High level input current	All inputs except $\overline{IV}$ pins: V_{CC} = Max/V_{IN} = 4.5V			100	μA
I_{IL}	Low level input current	All inputs except $\overline{IV}$ pins: V_{CC} = Max/V_{IN} = 0.5V			−100	μA
I_{OS}	Short circuit output current[2]	$\overline{IV}$ pins: V_{CC} = Max	−30		−140	mA
I_{CEX}	Open-collector leakage current	$\overline{EOUT}$ & $\overline{FOUT}$ = High; V_{CC} = Min/V_O = 5.5V			100	μA
I_{OZH}	High-Z state output current (high level)	$\overline{IV}$ pins: V_{CC} = Max/V_O = 2.7V			100	μA
I_{OZL}	High-Z state output current (low level)	$\overline{IV}$ pins: V_{CC} = Max/V_O = 0.5V			−100	μA
I_{CC}	Supply current	All inputs except $\overline{IV}$ pins: V_{CC} = Max/V_{IN} = 0V			200	mA

NOTES:
1. All voltages measured with respect to ground terminal.
2 At any one time, no more than one output should be connected to ground.

Product Specification

AC ELECTRICAL CHARACTERISTICS (V_{CC} = 5V ± 5%/0°C ≤ T_A ≤ 70°C)

SYMBOL	PARAMETER	REFERENCES		TEST CONDITIONS[1,2,3]	LIMITS		UNITS
		From	To		Min	Max	
Pulse Widths:							
t_{WMC}	MCLK pulse width	↑MCLK	↓MCLK	$\overline{FS}$ & $\overline{RESET}$ = High; $\overline{ME}$, $\overline{EIN}$ & $\overline{FIN}$ = Low	35		ns
t_{WRS}	Reset pulse width[4]	↓$\overline{RESET}$	↑$\overline{RESET}$	$\overline{FS}$ = High; $\overline{ME}$, $\overline{EIN}$ & $\overline{FIN}$ = Low	60		ns
Setup Times:							
t_{SIVMC}	$\overline{IV}$ setup time	$\overline{IV}$	↓MCLK	$\overline{FS}$ & $\overline{RESET}$ = High; $\overline{ME}$, $\overline{EIN}$ & $\overline{FIN}$ = Low	40		ns
t_{SWCMC}	Write command setup time	↑WC	↓MCLK	$\overline{FS}$ & $\overline{RESET}$ = High; SC, $\overline{ME}$, $\overline{EIN}$ & $\overline{FIN}$ = Low	60		ns
t_{SSCMC}	Select command setup time	↑SC	↓MCLK	$\overline{FS}$ & $\overline{RESET}$ = High; WC, $\overline{ME}$, $\overline{EIN}$ & $\overline{FIN}$ = Low	55		ns
t_{SMEMC}	Master enable setup time	↓$\overline{ME}$	↓MCLK	$\overline{FS}$ & $\overline{RESET}$ = High; $\overline{EIN}$ & $\overline{FIN}$ = Low	55		ns
Hold Times:							
t_{HIVMC}	$\overline{IV}$ hold time	↓MCLK	$\overline{IV}$	$\overline{FS}$ & $\overline{RESET}$ = High; $\overline{ME}$, $\overline{EIN}$ & $\overline{FIN}$ = Low	5		
t_{HMEMC}	Master enable hold time	↓MCLK	↑$\overline{ME}$	$\overline{FS}$ & $\overline{RESET}$ = High; $\overline{EIN}$ & $\overline{FIN}$ = Low	5		ns
t_{HSCMC}	Select command hold time	↓MCLK	↓SC	$\overline{FS}$ & $\overline{RESET}$ = High; WC, $\overline{ME}$, $\overline{EIN}$ & $\overline{FIN}$ = Low	0		
t_{HWCMC}	Write command hold time	↓MCLK	↓WC	$\overline{FS}$ & $\overline{RESET}$ = High; SC, $\overline{ME}$, $\overline{EIN}$ & $\overline{FIN}$ = Low	3		ns
Enable/Disable Times:							
t_{EIVWC}	$\overline{IV}$ output enable time	↓WC	$\overline{IV}$	$\overline{FS}$ & $\overline{RESET}$ = High; SC, $\overline{ME}$, MCLK, $\overline{EIN}$ & $\overline{FIN}$ = Low		30	ns
t_{EIVSC}	$\overline{IV}$ output enable time	↓SC	$\overline{IV}$	$\overline{FS}$ & $\overline{RESET}$ = High; WC, $\overline{ME}$, MCLK, $\overline{EIN}$ & $\overline{FIN}$ = Low		30	ns
t_{EIVME}	$\overline{IV}$ output enable time	↓$\overline{ME}$	$\overline{IV}$	$\overline{FS}$ & $\overline{RESET}$ = High; WC, SC, MCLK, $\overline{EIN}$ & $\overline{FIN}$ = Low		30	ns
t_{EIVFS}	$\overline{IV}$ output enable time	↓$\overline{FS}$	$\overline{IV}$	$\overline{RESET}$ = High, WC, SC, $\overline{ME}$, MCLK, $\overline{EIN}$ & $\overline{FIN}$ = Low		35	ns
t_{DIVWC}	$\overline{IV}$ output disable time	↑WC	$\overline{IV}$	$\overline{FS}$ & $\overline{RESET}$ = High; SC, $\overline{ME}$, MCLK, $\overline{EIN}$ & $\overline{FIN}$ = Low[5]		25	ns
t_{DIVSC}	$\overline{IV}$ output disable time	↑SC	$\overline{IV}$	$\overline{FS}$ & $\overline{RESET}$ = High; WC, $\overline{ME}$, MCLK, $\overline{EIN}$ & $\overline{FIN}$ = Low[5]		25	ns
t_{DIVME}	$\overline{IV}$ output disable time	↑$\overline{ME}$	$\overline{IV}$	$\overline{FS}$ & $\overline{RESET}$ = High; WC, SC, MCLK, $\overline{EIN}$ & $\overline{FIN}$ = Low[5]		25	ns
t_{DIVFS}	$\overline{IV}$ output disable time	↑$\overline{FS}$	$\overline{IV}$	$\overline{RESET}$ = High; WC, SC, $\overline{ME}$, MCLK, $\overline{EIN}$ & $\overline{FIN}$ = Low[5]		30	ns

TEST LOADING CIRCUITS

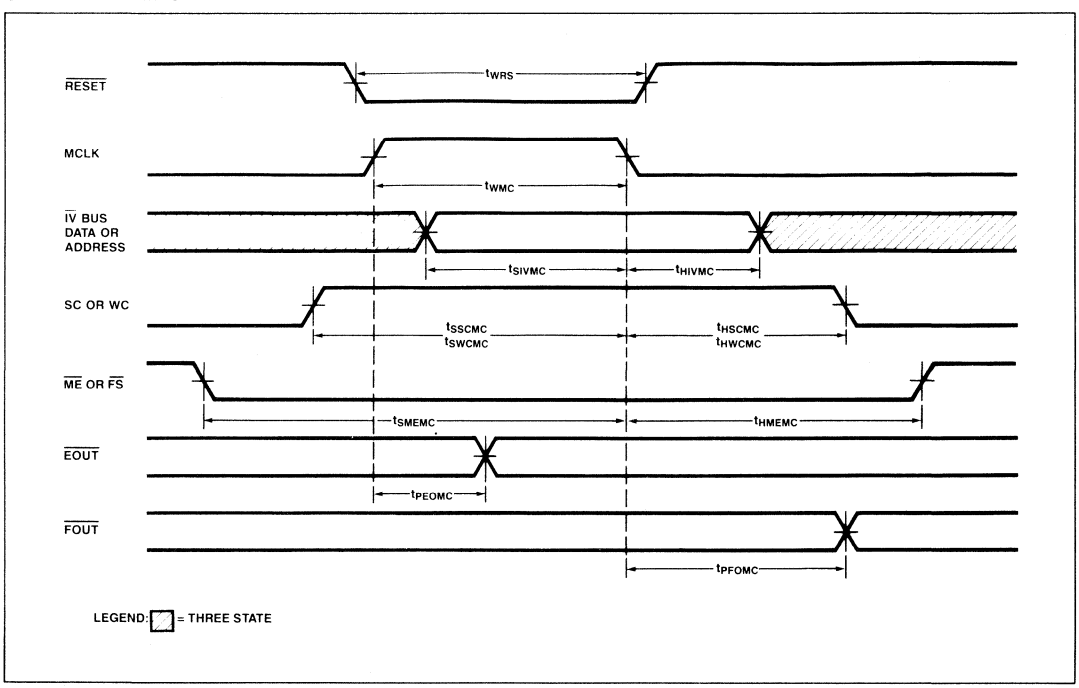
Product Specification

AC ELECTRICAL CHARACTERISTICS (V_{CC} = 5V ± 5%/0°C ≤ T_A ≤ 70°C) *(continued)*

SYMBOL	PARAMETER	REFERENCES		TEST CONDITIONS[1,2,3]	LIMITS		UNITS
		From	To		Min	Max	
Propagation Delays:							
t_{PEOMC}	Empty output propagation delay; both high-to-low and low-to-high transitions	↑MCLK	$\overline{EOUT}$	$\overline{FS}$ & $\overline{RESET}$ = High; WC, SC, $\overline{ME}$, $\overline{EIN}$ & $\overline{FIN}$ = Low		40	ns
t_{PFOMC}	Full output propagation delay; both high-to-low and low-to-high transitions	↓MCLK	$\overline{FOUT}$	$\overline{FS}$ & $\overline{RESET}$ = High; SC $\overline{ME}$, $\overline{EIN}$ & $\overline{FIN}$ = Low		45	ns
Stack Timing:							
t_{NOPOP}	Maximum spurious read (or pop) time[6]	↓SC or ↓WC	↑$\overline{ME}$ or ↑$\overline{FS}$	MCLK = Low		15	ns
t_{POP}	Minimum read (or pop) time[7]	Setup read condition (see truth table)	Stack popped	SC, WC & MCLK = Low; or $\overline{ME}$ or $\overline{FS}$ = Low	40		ns

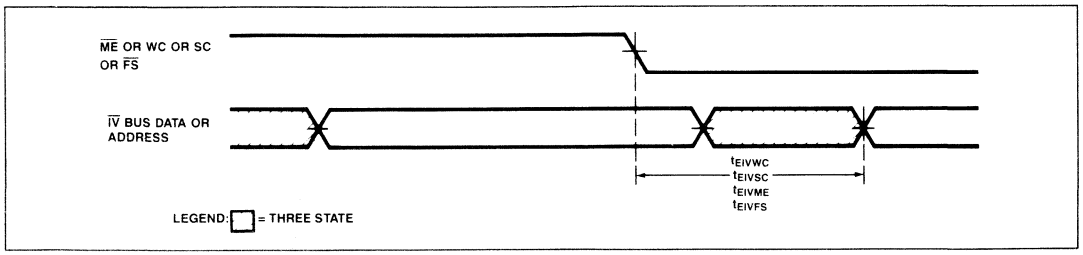
NOTES:
1. Loading — see TEST LOADING circuits.
2. All inputs are driven between 0.0 and 3.0 volts; input timing parameters are measured at 1.5 volts.
3. Except for output disable times, all output timing parameters are measured at 1.5 volts.
4. Resetting the part initializes $\overline{EOUT}$ and $\overline{FOUT}$, respectively, to low and high states and three-states the $\overline{IV}$ bus; all of these conditions occur within the minimum $\overline{RESET}$ pulse width.
5. These parameters are measured with a capacitive loading of 50 picofarads and represent the output driver turn-off time. Disable times measured into a capacitive loading of 300 picofarads do not exceed 40 nanoseconds, except for $\overline{FS}$ which is 45 nanoseconds.
6. This parameter represents the maximum time a Read (Pop) condition can be logically asserted, that is, as a decoding transient, without popping the stack.
7. This parameter represents the minimum time a Read (Pop) condition must be logically asserted to pop the stack.

OVERALL TIMING

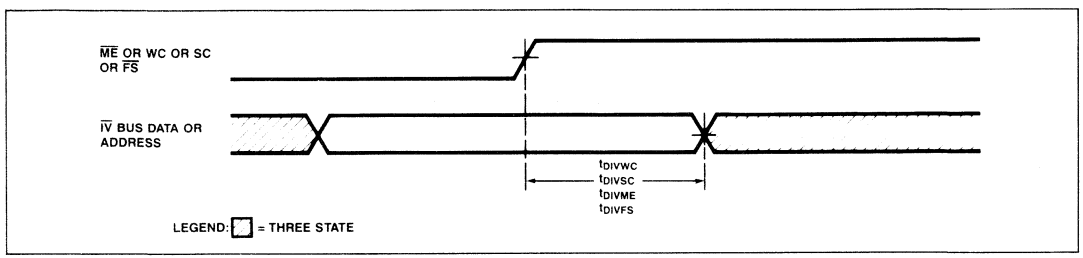


Product Specification

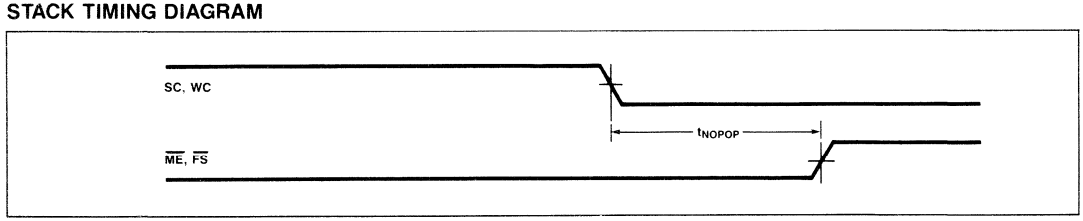
ENABLE TIMING



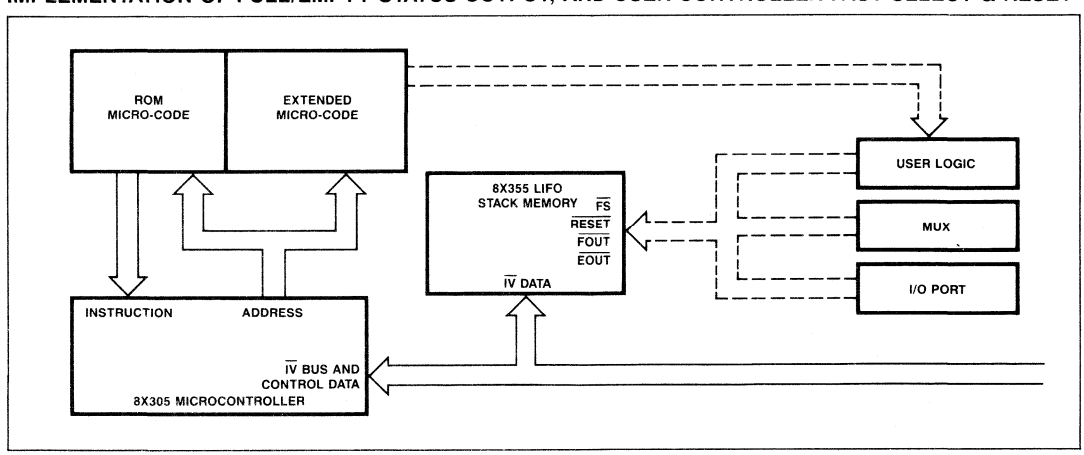
DISABLE TIMING



STACK TIMING DIAGRAM

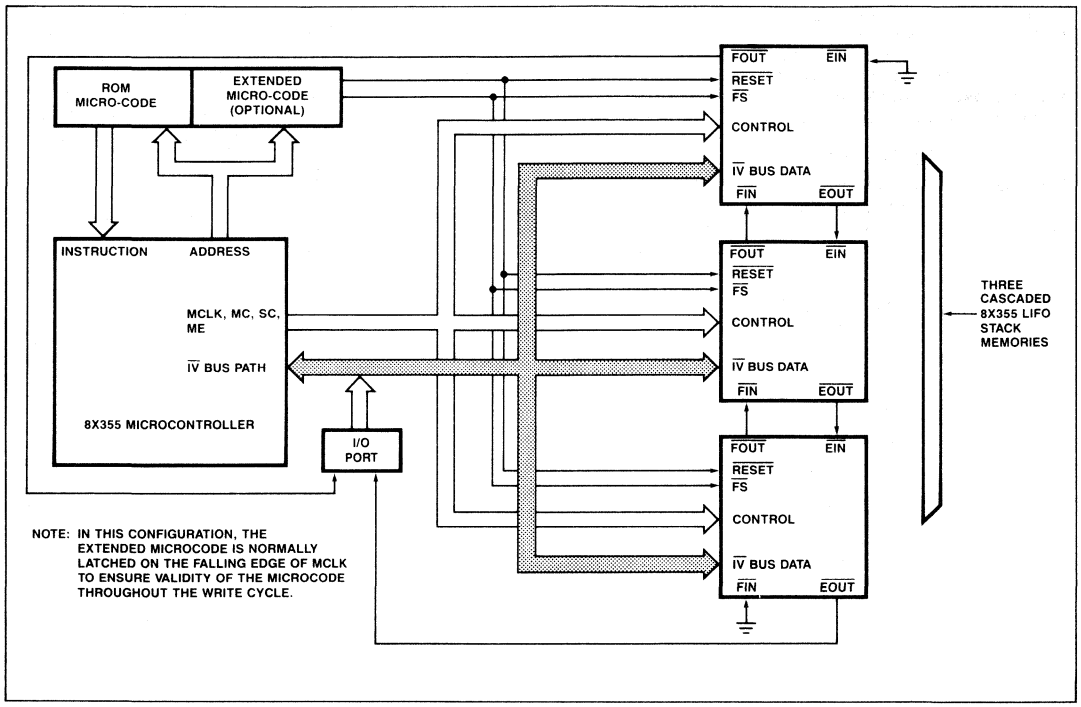


IMPLEMENTATION OF FULL/EMPTY STATUS OUTPUT, AND USER-CONTROLLER FAST SELECT & RESET



Product Specification

IMPLEMENTATION OF CASCADED 8X335's



MEMORY ADDRESS DIRECTOR

Originally published by Signetics January 1984

FEATURES

- **Address control for working storage**
- **16-bit addressing capability**
- **Byte and word addressing support**
- **Automatic increment and decrement**
- **11 Address and word-count registers**
- **Reduces number of 8X305 instructions required**

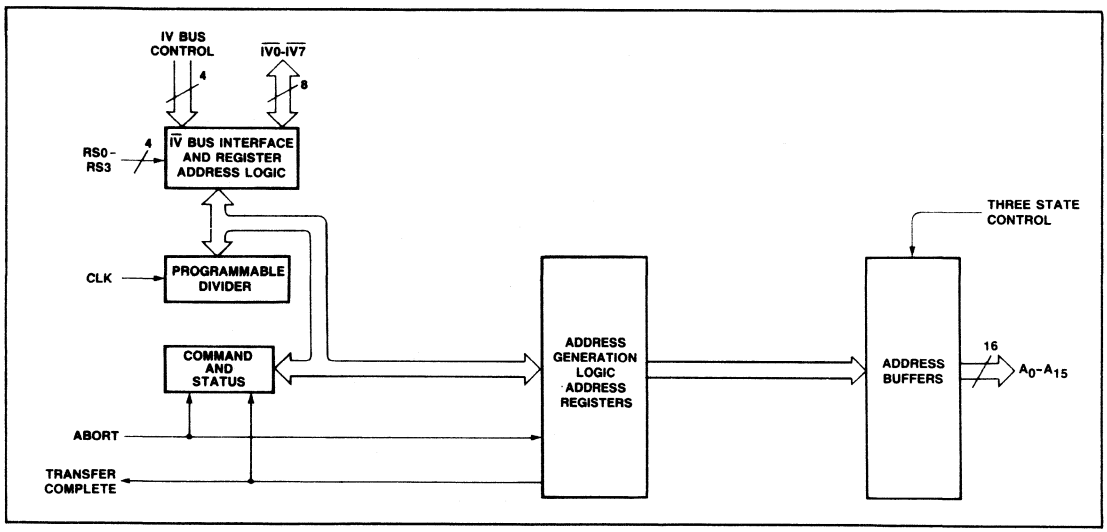
PRODUCT DESCRIPTION

The 8X360 Memory Address Director (MAD) is a high-performance member of the 8X300 Family that generates sequential memory addresses to facilitate the transfer of data to and from memory. The MAD provides a highly-efficient and cost-effective solution for DMA and other applications requiring large working-storage memories and high-speed data transfers. Once initialized with such information as starting address, ending address, byte count, address increment, address decrement, etc., the 8X360 performs all bookkeeping chores automatically and all address-management software is off-loaded from the processor. The 8X360 can be addressed by conventional means or by extended microcode; system status is available to the user via I/O pins.

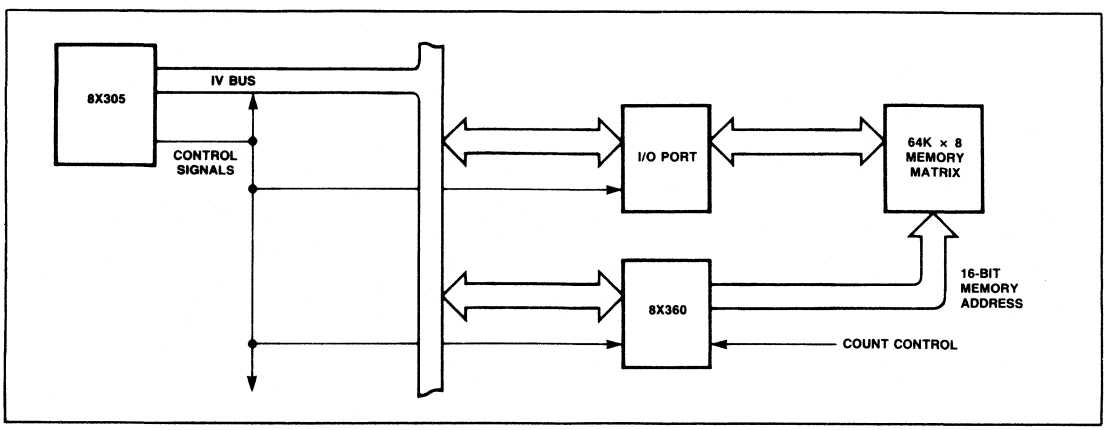
ORDERING INFORMATION

N8X360N, N8X360I, S8X360I

BLOCK DIAGRAM OF 8X360



8X360 APPLICATIONS



8-BIT LATCHED BIDIRECTIONAL I/O PORT

Originally published by Signetics January 1984

FEATURES

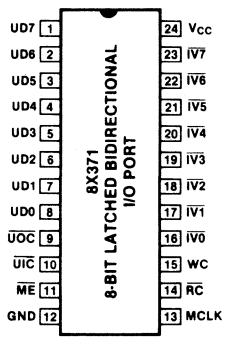
- **Two bidirectional 8-bit busses**
- **Independent bus operation (user-bus priority for data entry)**
- **User data input synchronous with respect to MCLK**
- **Three-state TTL outputs with high-drive capabilities**
- **Power-up to predetermined logic state**
- **Directly compatible with 8X305 (or 8X300) MicroControllers**
- **Single +5V supply**
- **0.4 inch 24-pin DIP**

PRODUCT DESCRIPTION

The 8X371 I/O Port is a bidirectional device designed for use as an interface element in systems that use TTL-compatible busses. Typically, the 8X371 is used with the 8X305 MicroController and its associated Interface Vector ($\overline{IV}$) bus; however, it can also be used with the 8X300 MicroController or an equivalent microprocessor. The 8X371 is functionally the same and pin-for-pin compatible with the older 8T31/8X31 but features improved performance and increased drive current. As shown in the logic diagram of Figure 1, the 8X371 consists of eight identical data latches—bits 0 through 7. The latches are accessed from either of two 8-bit busses—the MicroController ($\overline{IV}$ bus) and the user data (UD bus). Separate controls are provided for each bus and both busses operate independently, except when both attempt to input data at the same time; in such situations, the user bus always has priority. A Master Enable ($\overline{ME}$) input is available for additional control over the $\overline{IV}$ bus. The data latches are transparent, in that, while either bus is enabled for input, all input-data transitions are propagated to the other bus, if enabled for output.

8X371 PACKAGE and PIN DESIGNATIONS

N, I PACKAGE



TOP VIEW

ORDER NUMBERS

N8X371N, N8X371I
S8X371I/883B, S8X371I/883C

PIN. NO.	IDENTIFIER	FUNCTION
1-8	UD7-UD0	Three-state, bidirectional User Data (UD) bus; UD0 and corresponds to $\overline{IV0}$.
9	$\overline{UOC}$	User Output Control—active low input to enable data output to UD0-UD7.
10	$\overline{UIC}$	User Input Control—active low input to enable data input from UD0-UD7.
11	$\overline{ME}$	Master Enable—active low input to enable the $\overline{IV}$ bus for data input, or data output; UD-bus operations are unaffected.
12	GND	Ground.
13	MCLK	Master Clock—active high input from MicroController used to strobe data into data latches from the $\overline{IV}$ and UD buses.
14	$\overline{RC}$	Read Control—active low input to enable data output to $\overline{IV0}$-$\overline{IV7}$.
15	WC	Write Command—active high input from MicroController to enable the writing of data into the data latches from the $\overline{IV}$ bus provided $\overline{UIC}$ is not low.
16-23	$\overline{IV0}$-$\overline{IV7}$	Interface Vector (Input/Output Bus — three-state, bidirectional, MicroController data bus; $\overline{IV0}$ corresponds to UD0.
24	V_{CC}	+5V power supply.

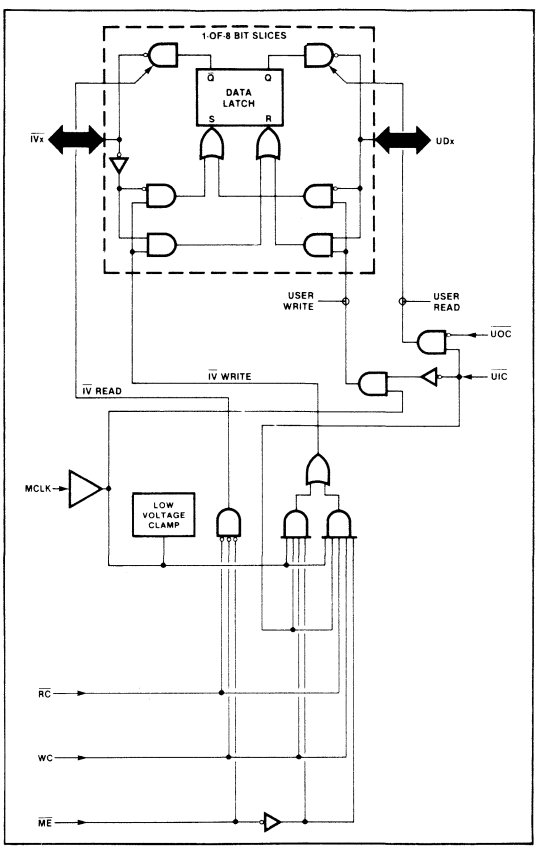


Figure 1. Logic Diagram for 8X371 I/O Port

FUNCTIONAL OPERATION

UD Bus Control

As shown in Table 1, the User Data (UD) bus interface is controlled by the $\overline{UIC}$ and $\overline{UOC}$ inputs. Data input to the UD bus is synchronous with MCLK, that is, with $\overline{UIC}$ low, information is written into the data latches only when MCLK is high. Output drivers on the UD bus are enabled when $\overline{UOC}$ is low and $\overline{UIC}$ is high.

Table 1. INPUT/OUTPUT CONTROL OF UD BUS

$\overline{UIC}$	$\overline{UOC}$	MCLK	FUNCTION OF UD BUS
H	L	X	Output data
L	X	H	Input data
L	X	L	Inactive
H	H	X	Inactive

X = don't care

$\overline{IV}$ Bus Control

Input/output control of the $\overline{IV}$ bus is shown in Table 2; this bus is controlled by $\overline{RC}$, WC, $\overline{ME}$, and MCLK. The $\overline{IV}$ bus is enabled for output (MicroController read operation) when $\overline{ME}$, $\overline{RC}$, and WC are all low. Data is written into the data latches from the $\overline{IV}$ bus when $\overline{ME}$ is low and both WC and MCLK are high. To avoid data-input conflicts, inputs from the $\overline{IV}$ bus are inhibited when $\overline{UIC}$ is low; under all other conditions, the $\overline{IV}$ and UD busses operate independently. The MicroController Left Bank ($\overline{LB}$) and Right Bank ($\overline{RB}$) outputs can control the $\overline{ME}$ inputs for two banks of I/O devices, thus acting as a ninth address bit. If more than one I/O Port (including the addressable parts—8X372, 8X376, 8X382, etc.) are to be connected to the same bank ($\overline{LB}$ or $\overline{RB}$) of the MicroController, selection of each 8X371 must be accomplished with external control logic to avoid bus conflicts.

Table 2. INPUT/OUTPUT CONTROL OF IV BUS

$\overline{ME}$	$\overline{RC}$	WC	MCLK	$\overline{UIC}$	FUNCTION OF IV BUS
L	L	L	X	X	Output Data
L	X	H	H	H	Input Data
L	H	L	X	X	Inactive
L	X	H	X	L	Inactive
L	X	H	L	H	Inactive
H	X	X	X	X	Inactive

Bus Logic Levels

Data written into the I/O port from either bus will appear inverted when read from the other bus. Data written into either bus will not be inverted when read from the same bus. (Note. *A logic "1" in MicroController software corresponds to a high level on the UD bus even though the $\overline{IV}$ bus is inverted.*) The 8X382 wakes up in the unselected state with all data bits latched at the "logic 1" level (UD bus outputs high if enabled).

DC ELECTRICAL CHARACTERISTICS

ABSOLUTE MAXIMUM RATINGS

PARAMETER		RATING	UNIT
V_{CC}	Power supply voltage	+7	Vdc
V_{IN}	Input voltage	+5.5	Vdc
T_{STG}	Storage temperature range	-65 to +150	°C

COMMERCIAL: $4.75V \le V_{CC} \le 5.25V$, $0°C \le T_A \le 70°C$
MILITARY: $4.5V \le V_{CC} \le 5.5V$, $-55°C \le T_C \le 125°C$

PARAMETER		TEST CONDITIONS	LIMITS (COMMERCIAL)			LIMITS (MILITARY)			UNIT
			Min	Typ	Max	Min	Typ	Max	
V_{CC}	Supply Voltage		4.75	5	5.25	4.5	5	5.5	V
V_{IH}	High Level Input Voltage		2.0			2.0			V
V_{IL}	Low Level Input Voltage				0.8			0.8	V
V_{CL}	Input Clamp Voltage	V_{CC} = Min; I_I = -10mA			-1.5			-1.5	V
I_{IH}	High Level Input Current[1]	V_{CC} = Max; V_{IH} = 2.7V		5	100		5	100	μA
I_{IL}	Low Level Input Current[1]	V_{CC} = Max; V_{IL} < 0.5V		-350	-550		-350	-550	μA
V_{OL}	Low Level Output Voltage $\overline{IV}$ Bus ($\overline{IV0}$-$\overline{IV7}$)	V_{CC} = Min; I_{OL} = 16mA			0.55			0.55	V
	User Bus (UD4-UD7)	V_{CC} = Min; I_{OL} = 24mA			0.55			0.55	V
V_{OH}	High Level Output Voltage	V_{CC} = Min; I_{OH} = -3.2mA	2.4			2.4			V
I_{OS}	Short Circuit Output Current[3] $\overline{IV}$ Bus ($\overline{IV0}$-$\overline{IV7}$)	V_{CC} = Max	-20			-20			mA
	UD Bus (UD4-UD7)	V_{CC} = Max	-10			-10			mA
I_{CC}	Supply Current	V_{CC} = Max; $\overline{ME}$ = $\overline{UOC}$ = V_{CC}		90	150		90	150	mA

Notes:
1. The input current includes the Three-state leakage current of the output driver on the data lines.
2. Only one output may be shorted at a time.

AC ELECTRICAL CHARACTERISTICS (Cont'd)

COMMERCIAL: $4.75 \le V_{CC} \le 5.25V$, $0°C \le T_A \le 70°C$
MILITARY: $4.5V \le V_{CC} \le 5.5V$, $-55°C \le T_C \le 125°C$

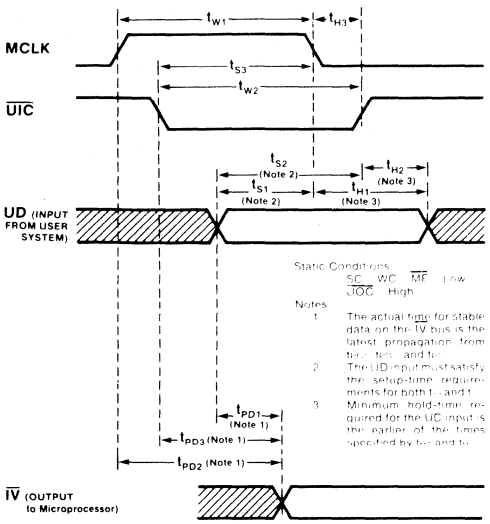
LOADING: See TEST LOADING CIRCUITS

PARAMETER		REFERENCES		TEST CONDITIONS	LIMITS (Commercial)			LIMITS (Military)			UNIT
		FROM	TO		Min	Typ	Max	Min	Typ	Max	
Pulse Widths:											
t_{W1}	Clock High	↑MCLK	↓MCLK		35			35			ns
t_{W2}	User Input Control	↓$\overline{UIC}$	↑$\overline{UIC}$	MCLK = High	35			35			ns
Propagation Delays:											
t_{PD1}	UD Propagation Delay	UD	$\overline{IV}$	MCLK = High $\overline{RC}$ = WC = $\overline{ME}$ = $\overline{UCI}$ = Low			30			30	ns
t_{PD2}	UD Clock Delay	↑MCLK	$\overline{IV}$	UD = Stable; $\overline{RC}$ = WC = $\overline{ME}$ = $\overline{UIC}$ = Low			50			50	ns
t_{PD3}	UD Input Delay	↓$\overline{UIC}$	$\overline{IV}$	UD = Stable; MCLK = High $\overline{RC}$ = WC = $\overline{ME}$ = Low			50			50	ns
t_{PD4}	$\overline{IV}$ Data Propagation Delay	$\overline{IV}$	UD	MCLK = WC = $\overline{UIC}$ = High; $\overline{ME}$ = $\overline{UOC}$ = $\overline{RC}$ = Low			45			45	ns
t_{PD5}	$\overline{IV}$ Data Clock Delay	↑MCLK	UD	WC = $\overline{UIC}$ = High; $\overline{IV}$ = Stable $\overline{ME}$ = $\overline{UOC}$ = $\overline{RC}$ = Low			55			55	ns
Output Enable Timing:											
t_{OE1}	UD Output Enable	↓UOC	UD	$\overline{UIC}$ = High			30			30	ns
t_{OE2}	UD Input Recovey	↑$\overline{UIC}$	UD	$\overline{UOC}$ = Low			30			30	ns
t_{OE3}	$\overline{IV}$ Data Master Enable	↓$\overline{ME}$	$\overline{IV}$	WC = $\overline{RC}$ = Low			22			25	ns
t_{OE4}	$\overline{IV}$ Data Read Enable	↓$\overline{RC}$	$\overline{IV}$	WC = $\overline{ME}$ = Low			25			25	ns
t_{OE5}	$\overline{IV}$ Data Write Recovery	↓WC	$\overline{IV}$	$\overline{RC}$ = $\overline{ME}$ = Low			25			25	ns
Output Disable Timing:											
t_{OD1}	UD Output Disable	↑$\overline{UOC}$	UD	$\overline{UIC}$ = High			25			25	ns
t_{OD2}	UD Input Override	↓$\overline{UIC}$	UD	$\overline{UOC}$ = Low			30			30	ns
t_{OD3}[1]	$\overline{IV}$ Data Master Disable	↑$\overline{ME}$	$\overline{IV}$	WC = $\overline{RC}$ = Low			20			20	ns
t_{OD4}[1]	$\overline{IV}$ Data Read Disable	↑$\overline{RC}$	$\overline{IV}$	WC = $\overline{ME}$ = Low			20			20	ns
t_{OD5}	$\overline{IV}$ Data Write Override	↑WC	$\overline{IV}$	$\overline{RC}$ = $\overline{ME}$ = Low			20			20	ns
Setup Time:											
t_{S1}	UD Clock Setup Time	UD	↓MCLK	$\overline{UIC}$ = Low	15			15			ns
t_{S2}	UD Setup Time	UD	↑UIC	MCLK = High	15			15			ns
t_{S3}	User Input Control Setup Time	↓$\overline{UIC}$	↓MCLK		25			25			ns
t_{S4}	$\overline{IV}$ Data Setup Time	$\overline{IV}$	↓MCLK	WC = $\overline{UIC}$ = High; $\overline{ME}$ = Low	35			35			ns
t_{S5}[2]	$\overline{IV}$ Master Enable Setup Time	↓$\overline{ME}$	↓MCLK	WC = $\overline{UIC}$ = High	30			30			ns
t_{S6}	$\overline{IV}$ Write Control Setup Time	↑WC	↓MCLK	$\overline{ME}$ = Low; $\overline{UIC}$ = High	30			30			ns

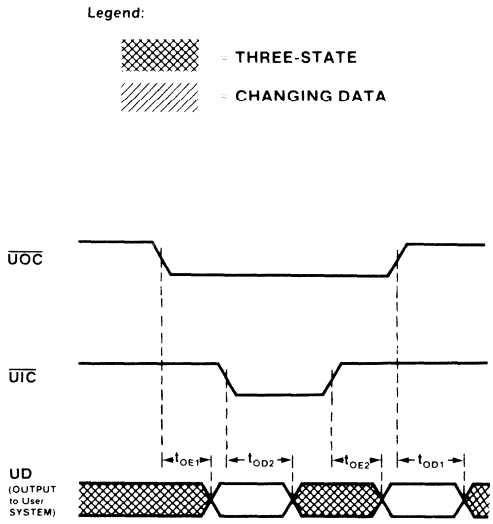
AC ELECTRICAL CHARACTERISTICS (Cont'd)

PARAMETER	REFERENCES		TEST CONDITIONS	LIMITS (Commercial)			LIMITS (Military)			UNIT
	FROM	TO		Min	Typ	Max	Min	Typ	Max	
Hold Times:										
t_{H1} UD Clock Hold Time	↓MCLK	UD	$\overline{UIC}$ = Low	15			15			ns
t_{H2} UD Control Hold Time	↑$\overline{UIC}$	UD	MCLK = High	15			15			ns
t_{H3} User Input Control Hold Time	↓MCLK	↑UIC		0			0			ns
t_{H4} $\overline{IV}$ Data Hold Time	↓MCLK	$\overline{IV}$	WC = $\overline{UIC}$ = High; $\overline{ME}$ = Low	5			5			ns
t_{H5}[2] $\overline{IV}$ Master Enable Hold Time	↓MCLK	↑$\overline{ME}$	WE = $\overline{UIC}$ = High	0			0			ns
t_{H6} $\overline{IV}$ Write Control Hold Time	↓MCLK	↓WC	$\overline{ME}$ = Low; = $\overline{UIC}$ = High	0			0			ns

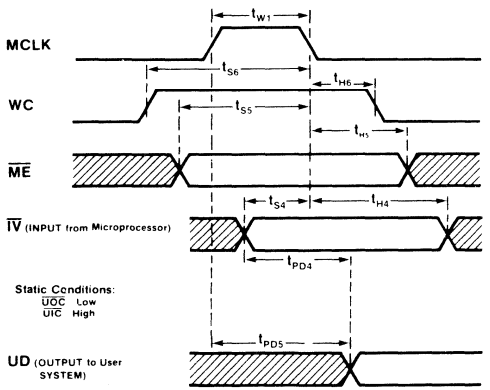
Notes:
1. These parameters are measured with a capacitive loading of 50 *pf* and represent the output driver turn-off time.
2. If $\overline{ME}$ is to be high (inactive), it must be setup before the rising edge and held after the falling edge of MCLK to avoid unintended writing into or selection of the I/O port.



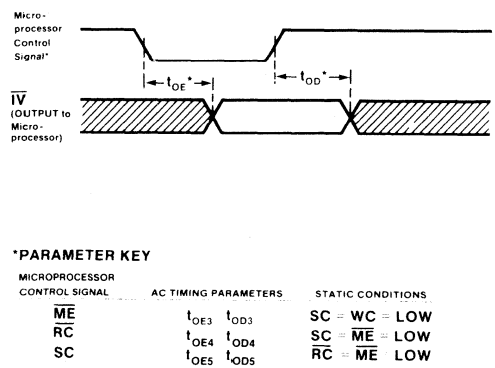
a. User Data Input Timing



b. User Data Output Timing



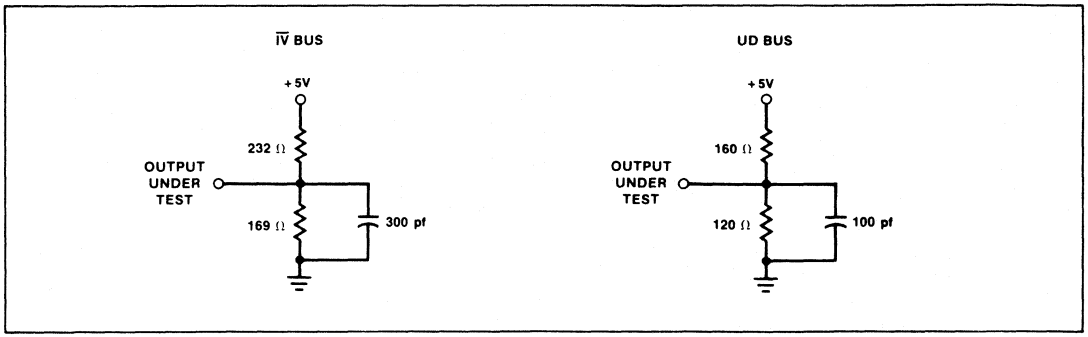
c. MicroController Write Cycle Timing



d. MicroController Select Cycle Timing

Figure 2. Timing Diagram

TEST LOADING CIRCUITS



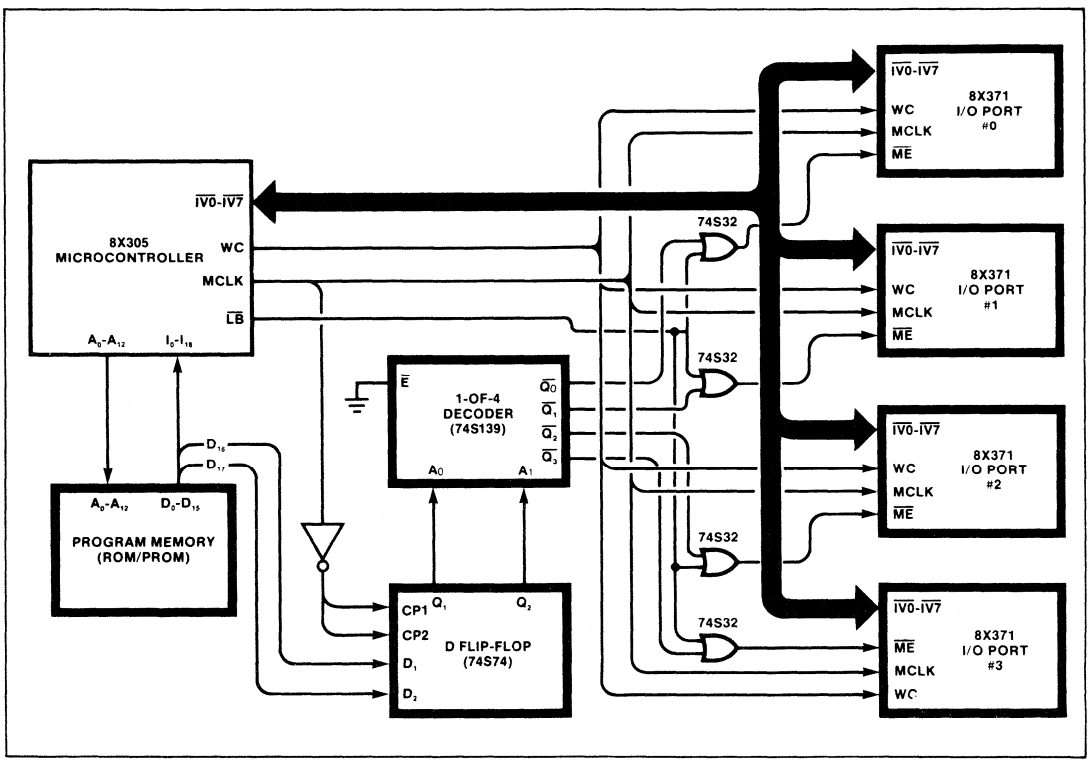
APPLICATIONS

In some applications, performance of a MicroController system can be enhanced by using the 8X371 I/O Port instead of an addressable 8X372 port. Using a technique referred to as Extended Microcode or Fast $\overline{IV}$ Select, the address select cycles which normally precede a read or write operation when using an 8X372 can be eliminated by use of the 8X371.

This technique is often used in bit slice microprocessor designs and involves widening the program memory beyond the normal 16-bit requirement of the MicroController. The extra bits are used as enable signals for the 8X371 ports. Thus, the 8X371 is enabled during the instruction cycle in which it is required for input/output operations. Since the software overhead of separate address select cycles is eliminated, the overall system performance is improved.

As shown in the accompanying diagram, the program memory is extended by two bit positions (D_{16} and D_{17}), permitting any one of four 8X371 ports to be enabled during those instructions that perform input/output operations. Because of timing considerations, latches must be used to hold the Extended Microcode through the end of the instruction cycle. A decoder is used to obtain four enable signals from the two extra bits. The decoder outputs are ORed with the $\overline{LB}$ output of the 8X305; thus, all four I/O ports are placed on the Left Bank of the $\overline{IV}$ bus.

I/O PORT SELECTION USING EXTENDED MICROCODE



ADDRESSABLE/BIDIRECTIONAL I/O PORTS

Originally published by Signetics January 1984

FEATURES

- **Two bidirectional 8-bit busses**
- **Independent bus operation (user-bus priority for data entry)**
- **User data input synchronous (8X372) or asynchronous (8X376) with respect to MCLK**
- **Programmed MicroController port address**
- **Three-state TTL outputs with high-drive capabilities**
- **Power-up to predetermined logic state**
- **Directly compatible with 8X305 or 8X300 MicroControllers**
- **Single +5V supply**
- **0.4 inch 24-pin DIP**

PRODUCT IDENTITY

8X372—Synchronous, three-state, bidirectional I/O port with programmed address.
8X376—Asynchronous, three-state, bidirectional I/O port with programmed address.

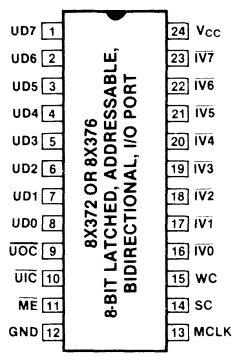
PRODUCT DESCRIPTION

Each of these I/O ports is an addressable device designed for use as a bidirectional interface element in systems that use TTL-compatible busses. Typically, these I/O ports are used with the 8X305 MicroController and its associated Interface Vector ($\overline{IV}$) bus; however, either port can also be used with the 8X300 MicroController or an equivalent microprocessor. The 8X372 and 8X376 are funtionally the same and pin-for-pin compatible with their respective counterparts, the 8T32/8X32 and 8T36/8X36; however, the new parts feature better performance, increased drive current, and improved programming procedures.

As shown in the logic diagram of Figure 1, each I/O port consists of eight identical data latches—bits 0 through 7. These latches are accessed through either of two 8-bit busses—one connecting to the MicroController ($\overline{IV}$ bus) and the other to the user system (UD bus). Separate controls are provided for each bus and both busses operate independently, except when both attempt to input data at the same time. In such situations, the user bus always has priority. The data latches are transparent, in that, while either bus is enabled for input, all transitions in input data are propagated to the other bus, if enabled for output.

Both the 8X372 and 8X376 are available with preprogrammed addresses (0_{10} through 255_{10}); either device can be field-programmed over the same address range. Input/output operations can begin once the I/O port is selected and appropriate control signals are generated. Port selection is implemented by putting the I/O port address (0_{10}-255_{10}) on the $\overline{IV}$ bus; once selected, the I/O port remains selected until a different "port address" is put on the bus. Thus, software overhead is minimized. Data is accessible on the UD bus at all times. A Master Enable ($\overline{ME}$) input, which is typically connected to the Left Bank ($\overline{LB}$) or Right Bank ($\overline{RB}$) output of the MicroController, provides the capability of organizing the $\overline{IV}$ bus into two separate and independent banks of I/O devices.

N, I PACKAGE



TOP VIEW

ORDER NUMBERS

N8X372N or N8X376N
N8X372I or N8X376I
S8X372I/883B or S8X372I/883C
S8X376I/883B or S8X376I/883C

PIN. NO.	IDENTIFIER	FUNCTION
1-8	UD7-UD0	Three-state bidirectional User Data (UD) bus. UD0 corresponds to $\overline{IV0}$
9	$\overline{UOC}$	User Output Control—active low input to enable data output to UD0-UD7
10	$\overline{UIC}$	User Input Control—active low input to enable data input from UD0-UD7
11	$\overline{ME}$	Master Enable—active low input to enable the $\overline{IV}$ bus for data input, data output, or $\overline{IV}$ address selection; deselection UD-bus operations are unaffected
12	GND	Ground
13	MCLK	Master Clock—active high input from MicroController used to strobe data into data latches from the IV bus and for the synchronous 8X372 from the UD bus. MCLK also synchronizes IV address selection
14	SC	Select Command—active high input from MicroController to enable $\overline{IV}$ address input from the $\overline{IV}$ bus for device selection
15	WC	Write Command—active high input from MicroController to enable the writing of data into the data latches from the $\overline{IV}$ bus, provided $\overline{UIC}$ is not low
16-23	$\overline{IV0}$-$\overline{IV7}$	Interface Vector Input Output Bus—three-state bidirectional MicroController data bus. $\overline{IV0}$ corresponds to UD0
24	V_{CC}	5V power supply

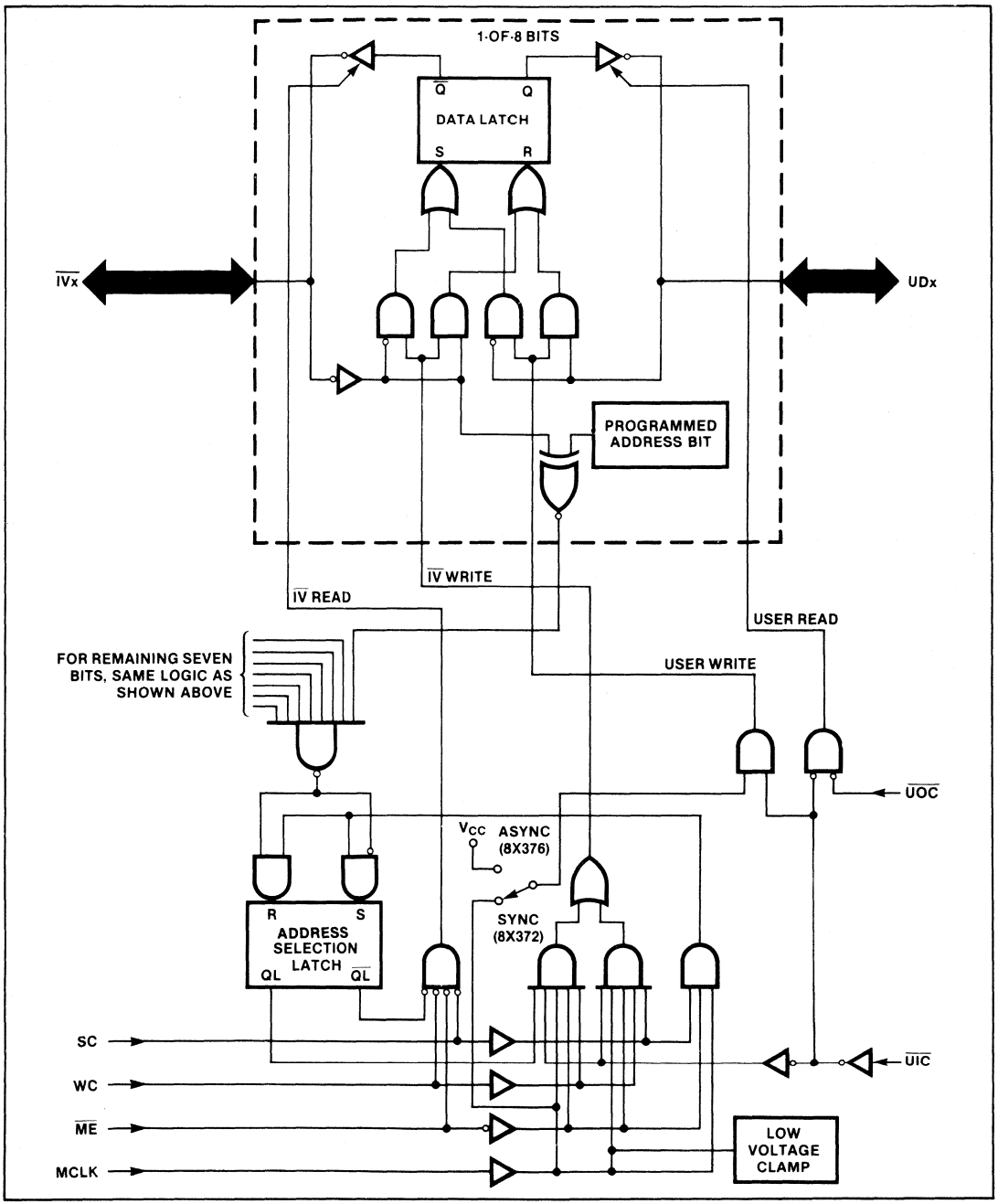


Figure 1. Logic Diagram for 8X372/8X376 I/O Ports

FUNCTIONAL OPERATION

UD Bus Control

As shown in Table 1, the User Data (UD) bus interface is controlled by the $\overline{UIC}$ and $\overline{UOC}$ inputs. For the 8X372, data input from the UD bus is written synchronously with MCLK, that is, with $\overline{UIC}$ low, information is written into the data latches only when MCLK is high. In the case of the 8X376, data input is asynchronous, in that, with $\overline{UIC}$ low, data is latched in without regard to the level of MCLK. Note. *To avoid the possibility of processor error when using the asynchronous 8X376, the $\overline{IV}$ bus should not be read during the time the data latches are changing due to user input.* Output drivers on the UD bus are enabled when UOC is low and $\overline{UIC}$ is high.

Table 1. INPUT/OUTPUT CONTROL OF UD BUS

$\overline{UIC}$	$\overline{UOC}$	MCLK	FUNCTION OF UD BUS	
			8X372	8X376
H	L	X	Output data	Output data
L	X	H	Input data	Input data
L	X	L	Inactive	Input data
H	H	X	Inactive	Inactive

X = don't care

$\overline{IV}$ Bus Control

Input/output control of the $\overline{IV}$ bus is shown in Table 2; this bus is controlled by SC, WC, $\overline{ME}$, MCLK and the current state of the internal address selection latch. AS shown in Table 2, $\overline{UIC}$ is required to indicate priority of the UD bus for data input operations. The selection latch in the I/O port stores the result of the most recent $\overline{IV}$ address selection. The latch is set when the internally preprogrammed address of the port matches the address on the $\overline{IV}$ bus during an address-selection operation (SC = MCLK = High/WC = Low). The latch is cleared when the two 8-bit address patterns are in disagreement. The $\overline{IV}$ bus can transfer data only when the selection latch is set. As shown in the APPLICATION DIAGRAM, the MicroController Left Bank ($\overline{LB}$) and Right Bank ($\overline{RB}$) outputs can control the $\overline{ME}$ inputs for two banks of I/O devices, thus, acting as a ninth address bit.

Table 2. INPUT/OUTPUT CONTROL OF IV BUS

$\overline{ME}$	SC	WC	MCLK	$\overline{UIC}$	SELECTION LATCH	FUNCTION OF $\overline{IV}$ BUS
L	L	L	X	X	Set	Output Data
L	L	H	H	H	Set	Input Data
L	H	L	H	X	X	Input Address*
L	H	H	H	H	X	Input data and address*
L	H	H	H	L	X	Input Address*
L	X	H	L	X	X	Inactive
L	H	X	L	X	X	Inactive
L	L	H	H	L	X	Inactive
L	L	X	X	X	Not Set	Inactive
H	X	X	X	X	X	Inactive

X = don't care

* Selection latch is updated

Data is written into the data latches of a selected device from the $\overline{IV}$ bus when WC, MCLK, and $\overline{UIC}$ are all high and $\overline{ME}$ is low. To prevent data-input conflicts, inputs from the $\overline{IV}$ bus are inhibited when $\overline{UIC}$ is low; under all other conditions, the $\overline{IV}$ and UD busses operate independently. Output drivers on the $\overline{IV}$ bus of a selected device are enabled when $\overline{ME}$, WC, and SC are all low and the address selection latch is set. With SC and WC both high, shaded entry of Table 2, the bit pattern present on $\overline{IV0}$-$\overline{IV7}$ is interpreted as both input data and $\overline{IV}$ address. Provided $\overline{UIC}$ is high, the data is latched into the data latches whether or not the I/O port has been previously selected. If the preprogrammed address of the I/O port matches the bit pattern on $\overline{IV0}$-$\overline{IV7}$ when SC and WC are both high, the selection latch is set; otherwise, it is reset. Note. *The MicroController never drives both SC and WC high at the same time.*

Bus Logic Levels

Data written into the I/O port from either bus will appear inverted when read from the other bus. Data written into either bus will not be inverted when read from the same bus. (Note. *A logic "1" in MicroController software corresponds to a high level on the UD bus even though the $\overline{IV}$ bus is inverted.* Both the 8X372 and 8X376 wakeup with the address selection latch in the unselected state and all data bits latched at the "logic 1" level (UD bus outputs high if enabled).

ADDRESS PROGRAMMING AND ADDRESS PROTECT

Programming Procedures

Both 8X372 and 8X376 can be programmed to respond to any address within a range of 0_{10} through 255_{10}. In an unprogrammed state, low level (≤0.8V) inputs on all $\overline{IV}$ bus lines (address 255_{10}) will select the device. To program a given address bit to match a high level (≥2.0V) input on the corresponding $\overline{IV}$ pin (a logical "0" to the MicroController), the counterpart UD-bus pin must be pulsed according to Table 3 and the following procedures:

Step 1: Set all control inputs to the inactive state—$\overline{UIC} = \overline{UOC} = \overline{ME} = V_{CC}$ and SC = WC = MCLK = GND; leave the UD and $\overline{IV}$ bus pins open.

Step 2: Increase V_{CC} to V_{CCP}.

Step 3: After V_{CC} has stabilized, apply a single programming pulse (Figure 2) to the user-bus bit that corresponds to the desired high-level $\overline{IV}$ address bit. The I/O port is programmed from the user bus (UD0-UD7) for addressing from the MicroController bus ($\overline{IV0}$-$\overline{IV7}$).

Step 4: Return V_{CC} to 0-volts. Note. *If the programming of all address bits is completed in less than 1-second, V_{CC} can remain at 9.0-volts for the required interval of time.*

Step 5: Step 1 through 3 are applicable to the programming of each address bit that requires a high-level $\overline{IV}$ match.

Table 3. PROGRAMMING SPECIFICATIONS

PARAMETERS	LIMITS			UNITS
	Min	Typ	Max	
V_{CCP} — Programming supply voltage:				
Address	8.75	9.0	9.25	V
Protect		0		V
Maximum time V_{CCP} >5.25V			1.0	Sec
Programming voltage:				
Address	8.75	9.0	9.25	V
Protect	8.75		9.25	V
Programming current:				
Address			5	mA
Protect			50	mA
t_r — Programming pulse rise time:				
Address	10		100	μS
Protect	10		100	μS
t_w — Programming pulse width	0.5		1.0	mS

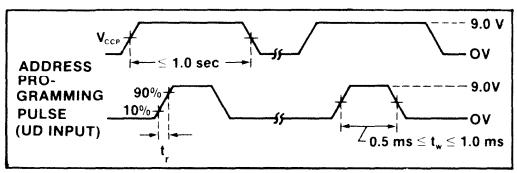


Figure 2. Address Programming Pulse

Step 6: To verify that the address is properly programmed, return V_{CC} to +5V, set $\overline{IV0}$-$\overline{IV7}$ to the desired (inverted) binary address pattern, set $\overline{ME}$ = WC = Low and SC = MCLK = High. If there are no programming errors, subsequent data written from $\overline{IV0}$-$\overline{IV7}$ (WC = High) will appear inverted on UD0-UD7.

Address Protect

After programming the I/O Port, steps should be taken to isolate the address circuits and make these circuits permanently immune to further change.

Step 1: Set V_{CC} and all control inputs to 0-volts (V_{CC} = $\overline{UIC}$ = $\overline{UOC}$ = $\overline{ME}$ = SC = WC = MCLK = 0V); $\overline{IV0}$-$\overline{IV7}$ = open circuit.

Step 2: Taking one pin at a time, apply a protect programming pulse (Figure 3) to each user-bus bit (UD0-UD7)—refer to Table 3 for min/max specifications pertaining to voltage and current.

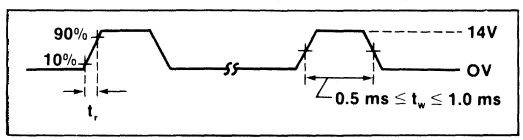


Figure 3. Protect Programming Pulse

Step 3: Verify that the address circuits for each bit is isolated by applying 9-volts, in turn, to each user-bus pin (UD0-UD7) and measuring less than 200 microamperes of input current. (Note. *Setup conditions are the same as those in Step 1.*)

DC ELECTRICAL CHARACTERISTICS

COMMERCIAL: 4.75V ≤ V_{CC} ≤ 5.25V, 0°C ≤ T_A ≤ 70°C
MILITARY: 4.5V ≤ V_{CC} ≤ 5.5V, -55°C ≤ T_C ≤ 125°C

ABSOLUTE MAXIMUM RATINGS

PARAMETER		RATING	UNIT
V_{CC}	Power supply voltage[3]	+7	Vdc
V_{IN}	Input voltage[3]	+5.5	Vdc
T_{STG}	Storage temperature range	-65 to +150	°C

PARAMETER		TEST CONDITIONS	LIMITS (COMMERCIAL)			LIMITS (MILITARY)			UNIT
			Min	Typ	Max	Min	Typ	Max	
V_{CC}	Supply Voltage		4.75	5	5.25	4.5	5	5.5	V
V_{IH}	High Level Input Voltage		2.0			2.0			V
V_{IL}	Low Level Input Voltage				0.8			0.8	V
V_{CL}	Input Clamp Voltage	V_{CC} = Min; I_I = -10mA			-1.5			-1.5	V
I_{IH}	High Level Input Current[1]	V_{CC} = Max; V_{IH} = 2.7V		5.0	100		5.0	100	μA
I_{IL}	Low Level Input Current[1]	V_{CC} = Max; V_{IL} = 0.5V		-350	-550		-350	-550	μA
V_{OL}	Low Level Output Voltage $\overline{IV}$ Bus ($\overline{IV0}$-$\overline{IV7}$)	V_{CC} = Min; I_{OL} = 16mA			0.55		0.55	V	
	User Bus (UD0-UD7)	V_{CC} = Min; I_{OL} = 24mA			0.55			0.55	V
V_{OH}	High Level Output Voltage	V_{CC} = Min; I_{OH} = -3.2mA	2.4			2.4			V
I_{OS}	Short Circuit Output Current[2] $\overline{IV}$ Bus ($\overline{IV0}$-$\overline{IV7}$)	V_{CC} = Max	-20			-20			mA
	UD Bus (UD0-UD7)	V_{CC} = Max	-10			-10			mA
I_{CC}	Supply Current	V_{CC} = Max; $\overline{ME}$ = $\overline{UOC}$ = V_{CC}		90	150		90	150	mA

NOTES:
1. The input current includes the Three-state leakage current of the output driver on the data lines.
2. Only one output may be shorted at a time.
3. These limits do not apply during address programming.

AC ELECTRICAL CHARACTERISTICS

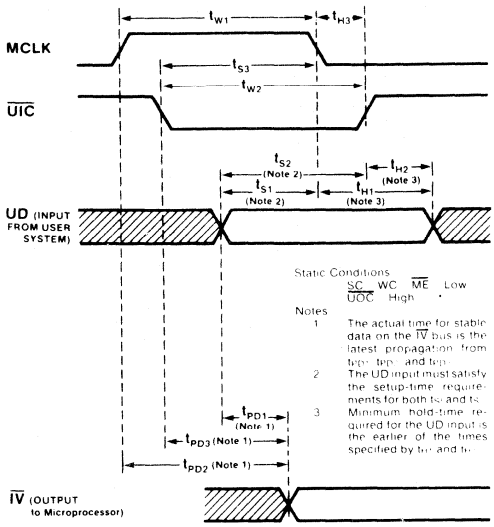
COMMERCIAL: $4.75V \leq V_{CC} \leq 5.25V$, $0°C \leq T_A \leq 70°C$
MILITARY: $4.5V \leq V_{CC} \leq 5.5V$, $-55°C = T_C \leq 125°C$
LOADING: See TEST LOADING CIRCUITS

PARAMETER		REFERENCES[1] FROM	REFERENCES[1] TO	TEST CONDITIONS	LIMITS (COMMERCIAL) Min	Typ	Max	LIMITS (MILITARY) Min	Typ	Max	UNIT
Pulse Widths:											
t_{W1}	Clock High	↑MCLK	↓MCLK		35			35			ns
t_{W2}	User Input Control	↓$\overline{UIC}$	↑$\overline{UIC}$	MCLK = High	35			35			ns
Propagation Delays:											
t_{PD1}	UD Propagation Delay	UD	$\overline{IV}$	MCLK = High SC = WC = $\overline{ME}$ = $\overline{UIC}$ = Low			30			30	ns
t_{PD2}	UD Clock Delay (8X732 only)	↑MCLK	$\overline{IV}$	UD = Stable; SC = WC = $\overline{ME}$ = $\overline{UIC}$ = Low			50			50	ns
t_{PD3}	UD Input Delay	↓$\overline{UIC}$	$\overline{IV}$	UD = Stable; MCLK = High; SC = WC = $\overline{ME}$ = Low			50			50	ns
t_{PD4}	$\overline{IV}$ Data Propagation Delay	$\overline{IV}$	UD	MCLK = WC = $\overline{UIC}$ = High; $\overline{ME}$ = $\overline{UOC}$ = SC = Low			45			45	ns
t_{PD5}	$\overline{IV}$ Data Clock Delay	↑MCLK	UD	WC = $\overline{UIC}$ = High, $\overline{IV}$ = Stable, $\overline{ME}$ = $\overline{UOC}$ = SC = Low			55			55	ns
Output Enable Timing:											
t_{OE1}	UD Output Enable	↓$\overline{UOC}$	UD	$\overline{UIC}$ = High			30			30	ns
t_{OE2}	UD Input Recovery	↑$\overline{UIC}$	UD	$\overline{UOC}$ = Low			30			30	ns
t_{OE3}	$\overline{IV}$ Data Master Enable	↓$\overline{ME}$	$\overline{IV}$	WC = SC = Low			22			25	ns
t_{OE5}	$\overline{IV}$ Data Write Recovery	↓WC	$\overline{IV}$	SC = $\overline{ME}$ = Low			25			25	ns
t_{OE6}	$\overline{IV}$ Data Select Recovery	↓SC	$\overline{IV}$	SC = $\overline{ME}$ = Low			25			25	ns
Output Disable Timing:											
t_{OD1}	UD Output Disable	↑$\overline{UOC}$	UD	$\overline{UIC}$ = High			25			25	ns
t_{OD2}	UD Input Override	↓$\overline{UIC}$	UD	$\overline{UOC}$ = Low			30			30	ns
t_{OD3}[2]	$\overline{IV}$ Data Master Disable	↑$\overline{ME}$	$\overline{IV}$	WC = SC = Low			20			20	ns
t_{OD4}[2]	$\overline{IV}$ Data Write Override	↑WC	$\overline{IV}$	SC = $\overline{ME}$ = Low			20			20	ns
t_{OD5}[2]	$\overline{IV}$ Data Select Override	↑SC	$\overline{IV}$	WC = $\overline{ME}$ = Low			20			20	ns
Setup Times:											
t_{S1}	UD Clock Setup Time (8X372 only)	UD	↓MCLK	$\overline{UIC}$ = Low	15			15			ns
t_{S2}	UD Control Setup Time	UD	↑$\overline{UIC}$	MCLK = High	15			15			ns
t_{S3}	User Input Control Setup Time (8X372 only)	↓$\overline{UIC}$	↓MCLK		25			25			ns
t_{S4}	$\overline{IV}$ Data Setup Time	$\overline{IV}$	↓MCLK	WC = High or SC = High; $\overline{ME}$ = Low; $\overline{UIC}$ = High	35			35			ns
t_{S5}[3]	$\overline{IV}$ Master Enable Setup Time	↓ME	↓MCLK	WC = High or SC = High; $\overline{UIC}$ = High	30			30			ns
t_{S6}	$\overline{IV}$ Write Control Setup Time	↑WC	↓MCLK	SC = $\overline{ME}$ = Low; $\overline{UIC}$ = High	30			30			ns

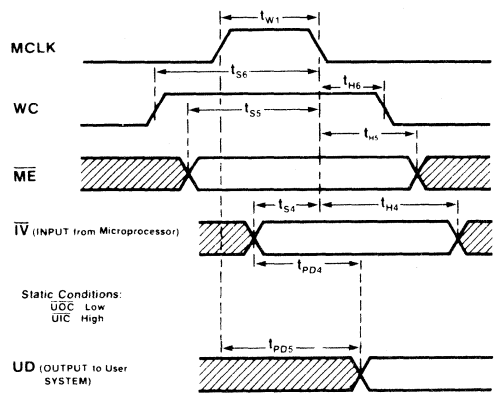
AC ELECTRICAL CHARACTERISTICS (Cont'd)

PARAMETER	REFERENCES FROM	REFERENCES TO	TEST CONDITIONS	LIMITS (COMMERCIAL) Min	Typ	Max	LIMITS (MILITARY) Min	Typ	Max	UNIT
t_{S7} $\overline{IV}$ Select Control Setup Time	↑SC	↓MCLK	WC = $\overline{ME}$ = Low	30			30			ns
Hold Times:										
t_{H1} UD Clock Hold Time (8X372 only)	↓MCLK	UD	$\overline{UIC}$ = Low	15			15			ns
t_{H2} UD Control Hold Time	↑$\overline{UIC}$	UD	MCLK = High	15			15			ns
t_{H3} User Input Control Hold Time (8X372 only)	↓MCLK	↑$\overline{UIC}$		0			0			ns
t_{H4} $\overline{IV}$ Data Hold Time	↓MCLK	$\overline{IV}$	WC = High or SC = High; $\overline{ME}$ = Low, $\overline{UIC}$ = High	5			5			ns
t_{H5}[3] $\overline{IV}$ Master Enable Hold Time	↓MCLK	↑$\overline{ME}$	WC = High or SC = High; $\overline{UIC}$ = High	0			0			ns
t_{H6} $\overline{IV}$ Write Control Hold Time	↓MCLK	↓WC	SC = $\overline{ME}$ = Low; $\overline{UIC}$ = High	0			0			ns
t_{H7} $\overline{IV}$ Select Control Hold Time	↓MCLK	↓SC	WC = $\overline{ME}$ = Low	0			0			ns

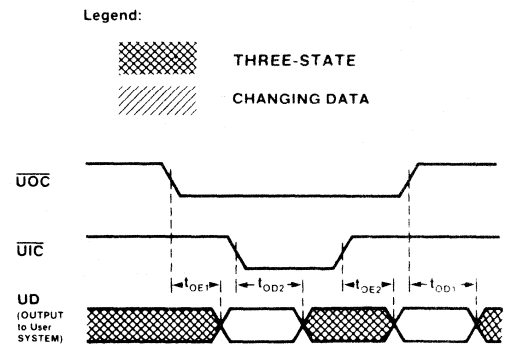
Notes:
1. All measurements to the $\overline{IV}$ bus assumes the address selection latch is set.
2. These parameters are measured with a capacitive loading of 50pf and represent the output driver turn-off time.
3. If $\overline{ME}$ is to be high (inactive), it must be setup before the rising edge and held after the falling edge of MCLK to avoid unintended writing into or selection of the I/O port.



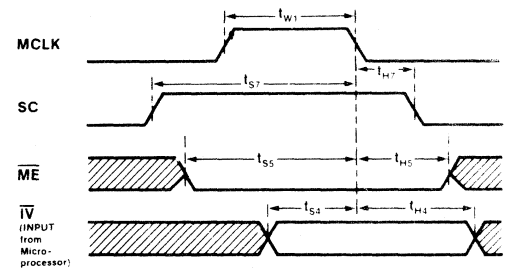
a. User Data Input Timing



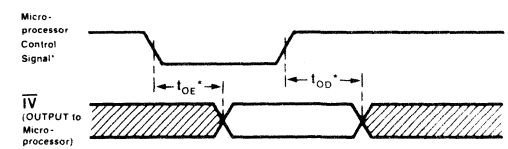
c. MicroController Write Cycle Timing



b. User Data Output Enable



d. MicroController Seclect Cycle Timing



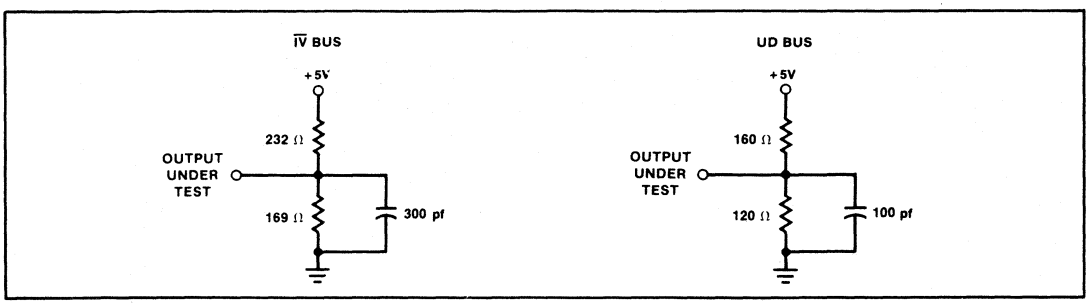
*PARAMETER KEY

MICROPROCESSOR CONTROL SIGNAL	AC TIMING PARAMETERS	STATIC CONDITIONS
ME	t_{OE3} t_{OD3}	SC WC LOW
WC	t_{OE5} t_{OD5}	SC ME LOW
SC	t_{OE6} t_{OD6}	WC ME LOW

e. MicroController Output Enable Timing

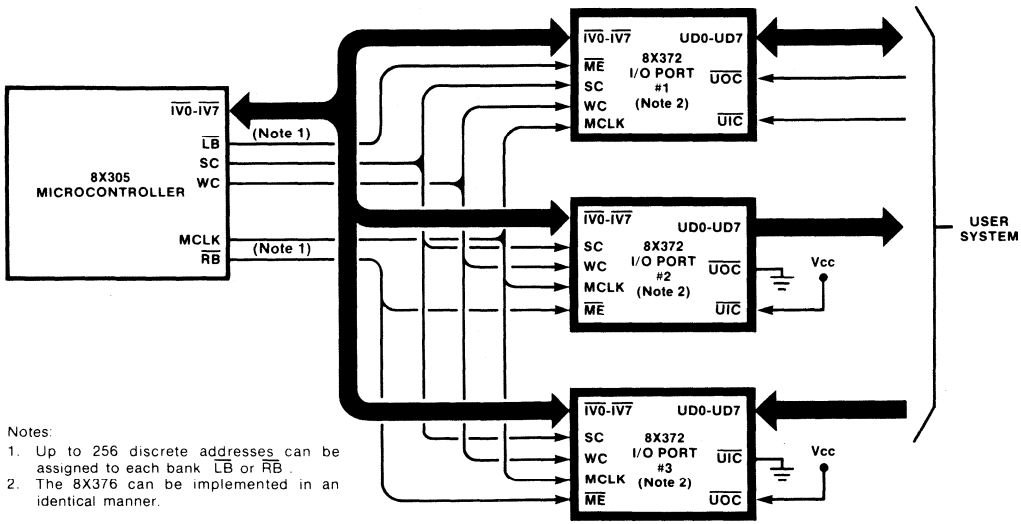
Figure 2. Timing Diagram

TEST LOADING CIRCUITS



APPLICATIONS

One way of using I/O Ports in a microprocessor-based system is shown in the following application diagram; there are many other ways of implementing I/O functions with these parts, both singly and in combination. By proper control of the $\overline{UIC}$ and $\overline{UOC}$ lines, the user can implement bidirectional data transfers, exercise system control, and/or read system status. In the concept shown here, I/O Port #1 is setup for bidirectional data transfers and I/O Ports #2 and #3, respectively, serve as dedicated output and input devices.



Notes:
1. Up to 256 discrete addresses can be assigned to each bank $\overline{LB}$ or $\overline{RB}$.
2. The 8X376 can be implemented in an identical manner.

ADDRESSABLE/BIDIRECTIONAL I/O PORT WITH PARITY

Originally published by Signetics January 1984

FEATURES

- **Two bidirectional 8-bit busses**
- **Independent bus operation (user bus priority for data entry)**
- **Parity generate/check logic with:**
  - **Odd/Even parity select**
  - **Strobed error flag output**
- **Synchronous data input**
- **Programmable MicroController port address**
- **Three-state TTL outputs (for all except parity error flag)**
- **High drive capabilities**
- **Power-up to predetermined state**
- **Directly compatible with 8X305 MicroController**
- **Single +5V supply**
- **0.6 inch, 28-pin DIP**

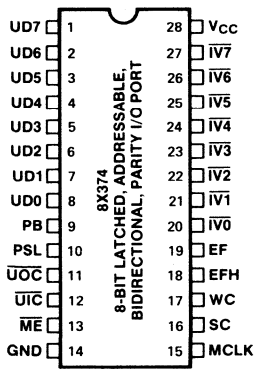
PRODUCT DESCRIPTION

The Signetics 8X374 is an addressable 8-bit I/O Port that features on-chip parity generate/check logic. The 8X374 port is designed for applications that require an 8-bit bidirectional interface element with parity-generate and parity-check capabilities. Typically, the 8X374 is used with the 8X305 MicroController and its associated Interface Vector (IV) bus.

As shown in the logic diagram of Figure 1, the 8X374 consists of eight identical latches, bits 0 through 7. These latches are accessed through either of two 8-bit busses, one connecting to the MicroController (IV bus) and the other to the user system (UD bus). Separate controls are provided for each bus and both busses operate independently, except when both attempt to input data at the same time. In such situations, the user bus always has priority. The data latches are transparent, in that, while either bus is enabled for input, all transitions in input data are propagated to the other bus, if enabled for output. The data latch in Figure 1 is common to both busses, that is, data traveling from the IV bus to the UD bus, or vice-versa, is latched and applied to the parity generate/check logic. The parity-bit latch is interfaced to the UD bus and latches the parity bit. The user can implement the parity features of the chip by simply selecting odd or even parity via the Parity SeLect (PSL) input pin. When data is output to the UD bus, a parity bit is generated and appended to each byte of data; for incoming data, parity is checked and the result is transmitted to an error-flag latch. The status of the latch (0 = no parity error/1 = parity error) is reflected by the Error Flag (EF) output pin. Operation of the error-flag latch is controlled by the Error Flag Hold (EFH) signal. With EFH low, the operation is transparent; when high the contents of the latch are frozen to avoid false errors while data latches are changing.

8X374 PACKAGE AND PIN DESIGNATIONS

N,F PACKAGE



TOP VIEW

ORDER NUMBERS

N8X374N, N8X374F
S8X374F/883B, S8X374F/883C

Pin No.	Identifier	Function
1-8	UD7-UD0	Three-state bidirectional User Data (UD) bus; UD0 corresponds to $\overline{IV0}$.
9	PB	User port Parity Bit I/O pin.
10	PSL	Parity SeLect input control; even parity = 1 and odd parity = 0.
11	$\overline{UOC}$	User Output Control — active low input to enable data output from UD0-UD7.
12	$\overline{UIC}$	User Input Control — active low input to enable data input from UD0-UD7.
13	$\overline{ME}$	Master Enable — active low input to enable the IV bus for data input, data output, or IV address selection/deselection; UD-bus operations are unaffected.
14	GND	Ground
15	MCLK	Master Clock — active high input from MicroController used to strobe data into the data latches; MCLK also synchronizes IV address selection.
16	SC	Select Command — active high input from MicroController to enable IV address input from the IV bus for device selection.
17	WC	Write Command — active high input from MicroController to enable the writing of data into the data latches from the IV bus, provided $\overline{UIC}$ is not low.
18	EFH	Error Flag Hold signal to control error-flag latch. When low, latch operation is transparent; when high, contents of latch are frozen.
19	EF	Error Flag output; no parity error = 0, parity error = 1.
20-27	$\overline{IV0}$-$\overline{IV7}$	Interface Vector (Input/Output Bus), three-state, bidirectional, MicroController data bus; $\overline{IV0}$ corresponds to UD0.
28	V_{CC}	+5V power supply.

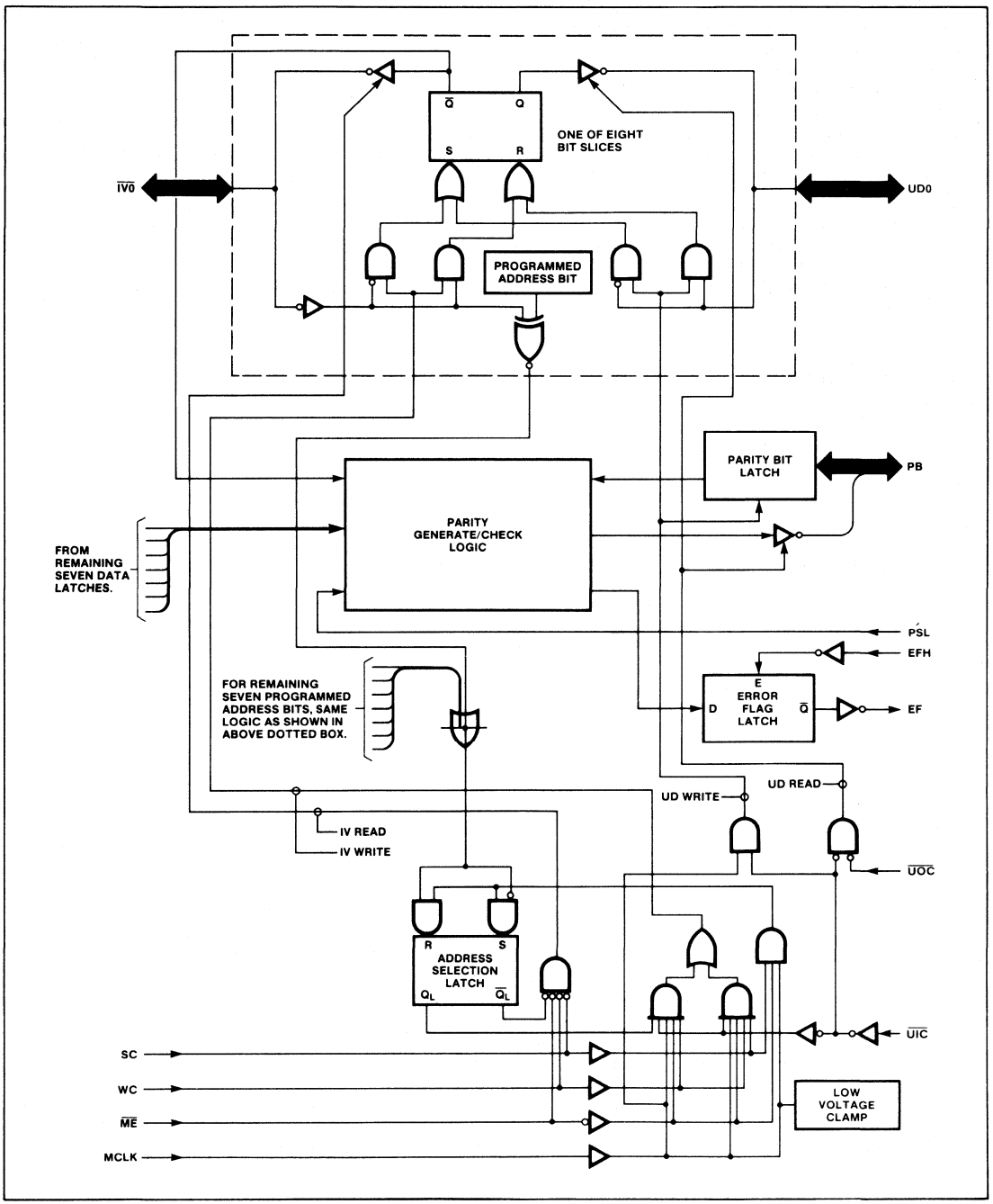


Figure 1. Logic Diagram for 8X374 I/O Port

The 8X374 is available with either preprogrammed addresses (0_{10} to 255_{10}) or unprogrammed; the device can be field-programmed over the same address range as the preprogrammed port. Input/Output operations to the Micro-Controller bus can begin once the 8X374 enabling address has been selected and appropriate control signals from the IV bus are generated. Port selection is implemented by putting the 8X374 address (0_{10} to 255_{10}) on the IV bus. Once selected, the I/O port remains selected until a different port address is put on the bus.

With appropriate control inputs, data is accessible on the UD bus at all times. A Master Enable (ME) input, which is typically connected to the Left Bank (LB) or Right Bank (RB) output of the MicroController, provides the capability of organizing the IV bus into two separate and independent banks of I/O devices.

FUNCTIONAL OPERATION

UD Bus Control

As shown in Table 1, the User Data (UD) bus and parity-bit interface are controlled by the UIC and UOC inputs. Data from the UD bus is written synchronously with MCLK, that is with UIC low, information is written into the data latches only when MCLK is high. Output drivers on the UD bus are enabled when UOC is low and UIC is high.

Table 1. INPUT/OUTPUT CONTROL OF UD BUS

			Function of UD Bus	
UIC	UOC	MCLK	8-Bit Data Bus	Parity Bit
H	L	X	Output data	Output parity
L	X	H	Input data	Input parity
L	X	L	Inactive	Inactive
H	H	X	Inactive	Inactive

X = Don't Care.

IV Bus Control

Input/Output control of the IV bus is shown in Table 2; this bus is controlled by SC, WC, ME, MCLK and the current state of the internal address selection latch. As shown in Table 2, UIC is required to indicate priority of the UD bus for data input operations. The selection latch in the I/O port stores the result of the most recent IV address selection. The latch is set when the internally preprogrammed address of the port matches the address on the IV bus during an address-selection operation (SC = MCLK = High; ME = WC = Low). The latch is cleared when the two 8-bit address patterns are in disagreement. The IV bus can transfer data only when the selection latch is set. As shown in the APPLICATION DIAGRAM, the 8X305 Left Bank (LB) and Right Bank (RB) outputs can control the ME inputs for two banks of I/O devices, thus, acting as a ninth address bit.

Table 2. INPUT/OUTPUT CONTROL OF IV BUS

ME	SC	WC	MCLK	UIC	Selection Latch	Function of IV Bus
L	L	L	X	X	Set	Output Data
L	L	H	H	H	Set	Input Data
L	H	L	H	X	X	Input Address*
L	X	H	L	X	X	Inactive
L	H	X	L	X	X	Inactive
L	L	H	H	L	X	Inactive
L	L	X	X	X	Not Set	Inactive
H	X	X	X	X	X	Inactive

X = Don't Care. *Selection latch is updated.

Data is written into the data latches of a selected device from the IV bus when WC, MCLK and UIC are all high and ME is low. To prevent data-input conflicts, inputs from the IV bus are inhibited when UIC is low; under all other conditions, the IV and UD busses operate independently. Output drivers on the IV bus of a selected device are enabled when ME, WC, and SC are all low and the address selection latch is set.

Parity Generate/Check Logic

The Parity Bit (PB) pin provides both parity-generate and parity-check capabilities according to user data bus controls. With UIC low (active), a parity check is performed on the input data stream; with UOC low (active) and UIC high, the 8X374 generates the parity-bit for the output data stream. The user can select odd or even parity via the Parity SeLect (PSL) input control, 1 = even parity and 0 = odd parity. As data and parity are input to the data latches and the parity-bit latch from the UD bus and PB line (Figure 1), parity errors (if any) are continuously detected by the parity-check logic. Parity error status enters the error flag latch (if enabled) and appears at the EF output pin. The error latch can be strobed by the Error Flag Hold (EFH) control to latch in valid error status; otherwise, the error flag is transparent to the user. (Note: If the sytem uses less than eight data bits, keeping zeros in unused data latches preserves proper parity operation.)

Bus Logic Levels

Data written into the I/O port from either bus will appear inverted when read from the other bus. Data written into either bus will not be inverted when read from the same bus. (Note: A logic "1" in MicroController software corresponds to a high level on the UD bus even though the IV bus

is inverted.) The 8X374 wakes up with the address selection latch in the unselected state, all data bits latched at the "logic 1" level (UD bus outputs high if enabled), and the EF output high.

ADDRESS PROGRAMMING AND ADDRESS PROTECT

Programming Procedures

The 8X374 can be programmed to respond to any address within a range of 0_{10} through 255_{10}. In an unprogrammed state, low level (≤ 0.8 V) inputs on all IV bus lines (address 255_{10}) will select the device. To program a given address bit to match a high level (≥ 2.0 V) input on the corresponding IV pin (a logical "0" to the MicroController), the counterpart UD-bus pin must be pulsed according to Table 3 and the following procedures:

Step 1: Set all control inputs to the inactive state, UIC = UOC = ME = V_{CC} and SC = WC = MCLK = 0 V; leave the UD and IV bus pins open.

Table 3. PROGRAMMING SPECIFICATIONS

Parameters	Limits Min.	Typ.	Max.	Units
V_{CCP} — Programming supply voltage:				
Address	8.75	9.0	9.25	V
Protect		0		V
Maximum Time V_{CC} > 5.25 V			1.0	sec
Programming voltage:				
Address	8.75	9.0	9.25	V
Protect	8.75		9.25	V
Programming current:				
Address			5	mA
Protect			50	mA
t_r — Programming pulse rise time:				
Address	10		100	μs
Protect	10		100	μs
t_W — Programming pulse width	0.5		1.0	ms

Step 2: Increase V_{CC} to V_{CCP}.

Step 3: After V_{CC} has stabilized, apply a single programming pulse (Figure 2) to the user-bus bit that corresponds to the desired high-level IV address bit. The I/O port is programmed from the user bus (UD0-UD7) for addressing from the MicroController bus (IV0-IV7).

Step 4: Return V_{CC} to 0 volts. (Note: If the programming of all address bits is completed in less than one second, V_{CC} can remain at V_{CCP} for the required interval of time.)

Step 5: Step 1 through Step 3 are applicable to the programming of each address bit that requires a high-level IV match.

Step 6: To verify that the address is properly programmed, return V_{CC} to +5 V and set IV0-IV7 to the desired address pattern (inverted). Set ME = WC = Low and SC = MCLK = High to select the programmed I/O port. With ME = SC = Low and WC = MCLK = High, write an 8-bit pattern to the port. If there are no programming errors, the transmitted data pattern will appear inverted at UD0-UD7 of selected port.

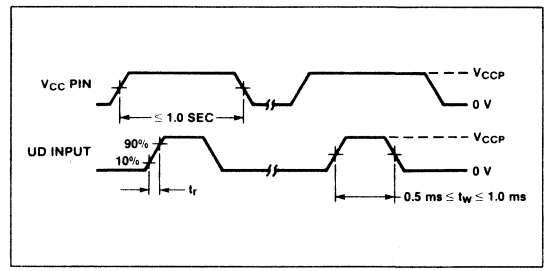


Figure 2. Address Programming Pulse

ADDRESS PROTECT

After programming the I/O Port, optional steps can be taken to isolate the fuse circuits and to make these circuits permanently immune to further change.

Step 1: Set V_{CC} and all control inputs to 0 volts; V_{CC} = UIC = UOC = ME = SC = WC = MCLK = 0V, IV0-IV7 = open circuit.

Step 2: Taking one pin at a time, apply a protect programming pulse (Figure 3) to each user-bus bit (UD0-UD7). Refer to Table 3 for min/max specifications pertaining to voltage and current.

Step 3: Verify that the address circuits for each bit are isolated by applying V_{CCP}, in turn, to each user-bus pin (UD0-UD7) and measuring less than 200 microamperes of input current. (Note: Setup conditions are the same as those in Step 1.)

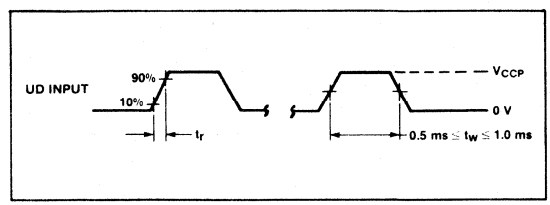


Figure 3. Protect Programming Pulse

ABSOLUTE MAXIMUM RATINGS

Parameter		Rating	Unit
V_{CC}	Power supply voltage[3]	+7	V DC
V_{IN}	Input voltage[3]	+5.5	V DC
T_{STG}	Storage temperature range	-65 to +150	°C

DC ELECTRICAL CHARACTERISTICS

COMMERCIAL: V_{CC} = 5 V (±5%); $T_A \geq 0°C$
$T_A \leq 70°C$
MILITARY: V_{CC} = 5 V (±10%); $T_A \geq -55°C$
$T_C \leq 125°C$

Parameter		Test Conditions	Limits (Commercial)			Limits (Military)			Unit
			Min	Typ	Max	Min	Typ	Max	
V_{CC}	Supply Voltage		4.75	5	5.25	4.5	5	5.5	V
V_{IH}	High Level Input Voltage		2.0			2.0			V
V_{IL}	Low Level Input Voltage				0.8			0.8	V
V_{CL}	Input Clamp Voltage	V_{CC} = Min; I_I = -10 mA			-1.5			-1.5	V
I_{IH}	High Level Input Current[1]	V_{CC} = Max; V_{IH} = 2.7 V		5.0	100		5.0	250	µA
I_{IL}	Low Level Input Current[1]	V_{CC} = Max; V_{IL} = 0.5 V		-350	-550		-350	-550	µA
V_{OL}	Low Level Output Voltage $\overline{IV}$ Bus ($\overline{IV0}$-$\overline{IV7}$)	V_{CC} = Min; I_{OL} = 16 mA			0.55			0.55	V
	User Bus (UD0-UD7) and PB	V_{CC} = Min; I_{OL} = 24 mA			0.55			0.55	V
	EF	V_{CC} = Min; I_{OL} = 8 mA			0.55			0.55	V
V_{OH}	High Level Output Voltage EF	V_{CC} = Min; I_{OH} = -1 mA	2.4			2.4			V
	Others	V_{CC} = Min; I_{OH} = -3.2 mA	2.4			2.4			V
I_{OS}	Short Circuit Output Current[2] $\overline{IV}$ Bus ($\overline{IV0}$-$\overline{IV7}$)	V_{CC} = Max	-20			-20			mA
	UD Bus (UD0-UD7)	V_{CC} = Max	-10			-10			mA
I_{CC}	Supply Current	V_{CC} = Max; $\overline{ME}$ = $\overline{UOC}$ = V_{CC}		90	150		90	160	mA

Notes:
1. The input current includes the high-Z leakage current of the output drivers ($\overline{IV0}$-$\overline{IV7}$, UD0-UD7) on the data lines.
2. Only one output may be shorted at a time for testing purposes.
3. These limits do not apply during address programming.

AC ELECTRICAL CHARACTERISTICS

COMMERCIAL: V_{CC} = 5 V (±5%); $T_A \geq 0°C$, $T_A \leq 70°C$
MILITARY: V_{CC} = 5 V (±10%); $T_A \geq -55°C$, $T_C \leq 125°C$
LOADING: See TEST LOADING CIRCUITS

Parameter		References From	References To	Test Conditions[1]	Limits (Commercial) Min	Typ	Max	Limits (Military) Min	Typ	Max	Unit
Pulse Widths:											
t_{W1}	Clock High	↑MCLK	↓MCLK		35			35			ns
t_{W2}	User Input Control	↑$\overline{UIC}$	↑$\overline{UIC}$	MCLK = High	35			35			ns
Propagation Delays:											
t_{PD1}	UD Propagation Delay	UD	$\overline{IV}$	MCLK = High; SC = WC = $\overline{ME}$ = $\overline{UIC}$ = Low			40			40	ns
t_{PD2}	UD Clock Delay	↑MCLK	$\overline{IV}$	UD = Stable; SC = WC = $\overline{ME}$ = $\overline{UIC}$ = Low			50			50	ns
t_{PD3}	UD Input Delay	↑$\overline{UIC}$	$\overline{IV}$	UD = Stable; MCLK = High; SC = WC = $\overline{ME}$ = Low			50			50	ns
t_{PD4}	$\overline{IV}$ Data Propagation Delay	$\overline{IV}$	UD	MCLK = WC = $\overline{UIC}$ = High; $\overline{ME}$ = $\overline{UOC}$ = SC = Low			45			45	ns
t_{PD5}	$\overline{IV}$ Data Clock Delay	↑MCLK	UD	WC = $\overline{UIC}$ = High; $\overline{IV}$ = Stable, $\overline{ME}$ = $\overline{UOC}$ = SC = Low			55			55	ns
t_{PD6}	Error Flag Propagation Delay	UD, PB	EF	MCLK = High; $\overline{UIC}$ = EFH = Low			55			55	ns
t_{PD7}	Parity Generate Propagation Delay	$\overline{IV}$	PB	MCLK = WC = $\overline{UIC}$ = High; $\overline{UOC}$ = $\overline{ME}$ = Low			55			55	ns
t_{PD8}	Error Flag Strobe Delay[3]	↑EFH	EF				20			20	ns
Output Enable Timing:											
t_{OE1}	UD Output Enable	↑$\overline{UOC}$	UD, PB	$\overline{UIC}$ = High			30			30	ns
t_{OE2}	UD Input Recovery	↑$\overline{UIC}$	UD, PB	$\overline{UOC}$ = Low			30			30	ns
t_{OE3}	$\overline{IV}$ Data Master Enable	↑$\overline{ME}$	$\overline{IV}$	WC = SC = Low			22			25	ns
t_{OE4}	$\overline{IV}$ Data Write Recovery	↑WC	$\overline{IV}$	SC = $\overline{ME}$ = Low			25			25	ns
t_{OE5}	$\overline{IV}$ Data Select Recovery	↑SC	$\overline{IV}$	SC = $\overline{ME}$ = Low			25			25	ns

AC ELECTRICAL CHARACTERISTICS (Continued)

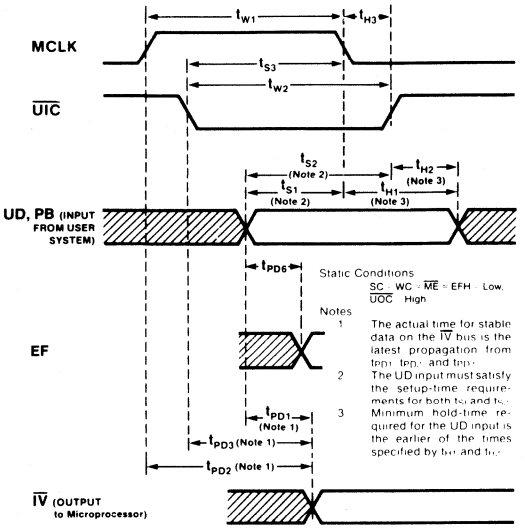
Parameter	References		Test Conditions[1]	Limits (Commercial)			Limits (Military)			Unit
	From	To		Min	Typ	Max	Min	Typ	Max	
Output Disable Timing:										
t_{OD1} UD Output Disable	↓$\overline{UOC}$	UD, PB	$\overline{UIC}$ = High			25			25	ns
t_{OD2} UD Input Override	↓$\overline{UIC}$	UD, PB	$\overline{UOC}$ = Low			30			30	ns
t_{OD3} $\overline{IV}$ Data Master Disable	↓$\overline{ME}$	$\overline{IV}$	WC = SC = Low			20			20	ns
t_{OD4} $\overline{IV}$ Data Write Override	↓WC	$\overline{IV}$	SC = $\overline{ME}$ = Low			20			20	ns
t_{OD5} $\overline{IV}$ Data Select Override	↓SC	$\overline{IV}$	WC = $\overline{ME}$ = Low			20			20	ns
Setup Times:										
t_{S1} UD Clock Setup Time	UD, PB	↑MCLK	$\overline{UIC}$ = Low	15			15			ns
t_{S2} UD Control Setup Time	UD, PB	↓$\overline{UIC}$	MCLK = High	15			15			ns
t_{S3} User Input Control Setup Time	↓$\overline{UIC}$	↑MCLK		25			25			ns
t_{S4} $\overline{IV}$ Data Setup Time	$\overline{IV}$	↑MCLK	WC = High or SC = High; $\overline{ME}$ = Low; $\overline{UIC}$ = High	35			35			ns
t_{S5}[2] $\overline{IV}$ Master Enable Setup Time	↓$\overline{ME}$	↑MCLK	WC = High or SC = High; $\overline{UIC}$ = High	30			30			ns
t_{S6} $\overline{IV}$ Write Control Setup Time	↓WC	↑MCLK	SC = $\overline{ME}$ = Low; $\overline{UIC}$ = High	30			30			ns
t_{S7} $\overline{IV}$ Select Control Setup Time	↓SC	↑MCLK	WC = $\overline{ME}$ = Low	30			30			ns
Hold Times:										
t_{H1} UD Clock Hold Time	↑MCLK	UD, PB	$\overline{UIC}$ = Low	15			15			ns
t_{H2} UD Control Hold Time	↓$\overline{UIC}$	UD, PB	MCLK = High	15			15			ns
t_{H3} User Input Control Hold Time	↑MCLK	↓$\overline{UIC}$		0			0			ns
t_{H4} $\overline{IV}$ Data Hold Time	↑MCLK	$\overline{IV}$	WC = High or SC = High; $\overline{ME}$ = Low, $\overline{UIC}$ = High	5			5			ns
t_{H5}[2] $\overline{IV}$ Master Enable Hold Time	↑MCLK	↑$\overline{ME}$	WC = High or SC = High; $\overline{UIC}$ = High	0			0			ns
t_{H6} $\overline{IV}$ Write Control Hold Time	↑MCLK	↑WC	SC = $\overline{ME}$ = Low; $\overline{UIC}$ = High	0			0			ns
t_{H7} $\overline{IV}$ Select Control Hold Time	↑MCLK	↑SC	WC = $\overline{ME}$ = Low	0			0			ns

Notes:

1. All measurements to the $\overline{IV}$ bus assumes the address selection latch is set.
2. If $\overline{ME}$ is to be high (inactive), it must be setup before the rising edge and held after the falling edge of MCLK to avoid unintended writing into or selection of the I/O port.
3. Parameters are measured by holding $\overline{UIC}$ = High and MCLK = Low and changing the state of the PSL input before each EFH pulse.

TIMING DIAGRAMS

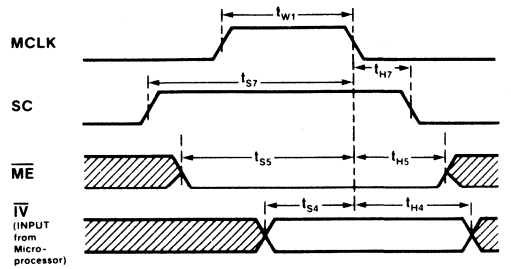
USER DATA INPUT TIMING



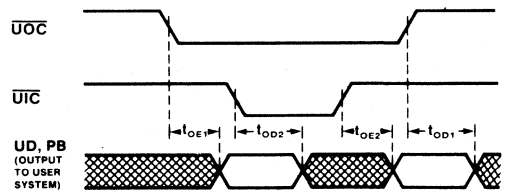
Legend:

- HIGH-Z STATE
- CHANGING DATA

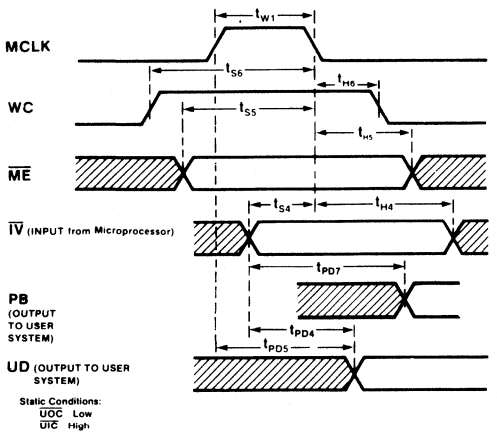
MICROCONTROLLER SELECT CYCLE TIMING



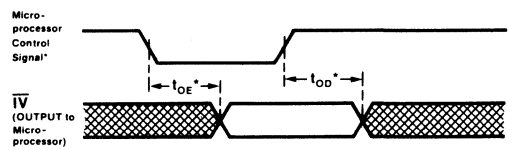
USER DATA OUTPUT ENABLE TIMING



MICROCONTROLLER WRITE CYCLE TIMING



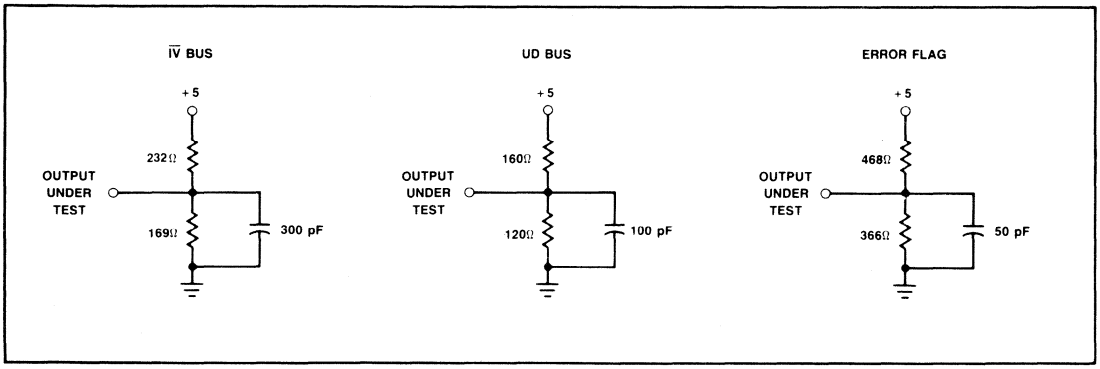
MICROCONTROLLER OUTPUT ENABLE TIMING



***PARAMETER KEY**

MICROPROCESSOR CONTROL SIGNAL	AC TIMING PARAMETERS	STATIC CONDITIONS
ME	t_{OE3} t_{OD3}	SC = WC = LOW
WC	t_{OE5} t_{OD5}	SC = ME = LOW
SC	t_{OE6} t_{OD6}	WC = ME = LOW

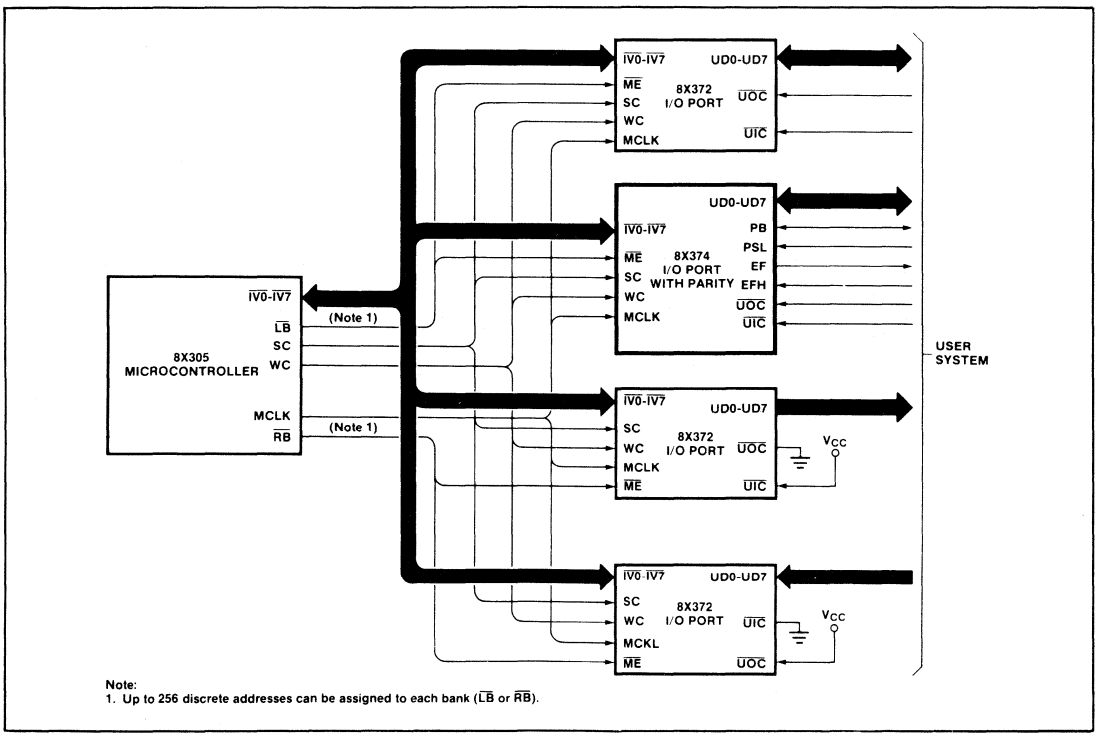
TEST LOADING CIRCUITS



APPLICATIONS

As shown in the following diagram, the 8X374 can be used with other I/O ports to provide a complete range of input/output functions. By proper control of the UIC and UOC lines, the user can perform bidirectional data transfers, exercise system control, read system status and, by using the 8X374, implement a bidirectional parity-controlled data stream. To use the parity capabilities, the user need only select even or odd parity (PSL = 1 or 0) and connect the PB pin to the system parity bit. The EFH and EF pins can be wired according to system requirements.

APPLICATIONS DIAGRAM



Note:
1. Up to 256 discrete addresses can be assigned to each bank ($\overline{LB}$ or $\overline{RB}$).

4-INPUT/4-OUTPUT ADDRESSABLE I/O PORT

Originally published by Signetics January 1984

FEATURES

- **Bidirectional 8-bit MicroController ($\overline{IV}$) bus**
- **User bus—four input bits and four output bits**
- **Independent bus operation**
- **Synchronous user data input**
- **Programmed MicroController port address**
- **Three-state TTL outputs with high-drive capabilities**
- **Power-up to predetermined logic state**
- **Directly compatible with 8X305 or 8X300 MicroControllers**
- **Single +5V supply**
- **0.4 inch 24-pin DIP**

PRODUCT DESCRIPTION

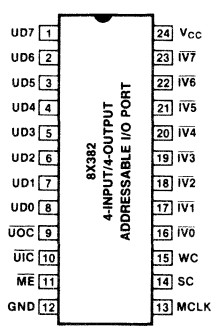
The 8X382 I/O Port is an addressable, three-state device designed for use as an interface element in systems that use TTL-compatible busses. Typically, the 8X382 is used with the 8X305 MicroController and its associated Interface Vector ($\overline{IV}$) bus; however, it can also be used with the 8X300 MicroController or an equivalent microprocessor. The 8X382 is functionally the same and pin-for-pin compatible with the older 8X42; however, the new port features better performance, increased drive current, and improved programming procedures.

As shown in the logic diagram of Figure 1, the I/O port consists of eight data latches—bits 0 through 7. These latches are accessed through either of two busses—an 8-bit bidirectional $\overline{IV}$ bus connected to the MicroController and a User Data (UD) bus consisting of four dedicated inputs (bits UD0 through UD3) and four dedicated outputs (bits UD4 through UD7). All eight bits may be read from or four data bits ($\overline{IV4}$-$\overline{IV7}$) can be written into via the $\overline{IV}$ bus; eight bits of I/O address can be written from the $\overline{IV}$ bus. Separate controls are provided for each bus and both busses operate independently. The I/O data latches are transparent, in that, when either bus is enabled for input, all transitions in input data are propagated to the other bus, if that bus in enabled for output.

The 8X382 is available with preprogrammed addresses (0_{10} through 255_{10}); it can also be field-programmed over the same address range. Input/output operations can begin once the I/O port is selected and appropriate control signals are generated. Port selection is implemented by putting the I/O port address (0_{10}-255_{10}) on the $\overline{IV}$ bus; once selected, the I/O port remains selected until a different "port address" is put on the bus. Thus, software overhead is minimized. Data is accessible on the UD bus at all times. A Master Enable ($\overline{ME}$) input, which is typically connected to the Left Bank ($\overline{LB}$) or Right Bank ($\overline{RB}$) output of the MicroController, provides the capability of organizing the $\overline{IV}$ bus into two separate and independent banks of I/O devices.

8X382 PACKAGE and PIN DESIGNATIONS

N, I PACKAGE



TOP VIEW

ORDER NUMBERS

N8X382N, N8X382I
S8X382I/883B, 8X382I/883C

PIN NO.	IDENTIFIER	FUNCTION
1-4	UD7-UD4	Three-state, dedicated output lines for user data; UD7 corresponds to $\overline{IV7}$.
5-8	UD3-UD0	Dedicated input lines for user data; UD0 corresponds to $\overline{IV0}$.
9	$\overline{UOC}$	User Output Control—active low input to enable data output to UD4-UD7.
10	$\overline{UIC}$	User Input Control—active low input to enable data input to UD0-UD3.
11	$\overline{ME}$	Master Enable—active low input to enable the $\overline{IV}$ bus for data input, data output, or $\overline{IV}$ address selection/deselection; UD-bus operations are unaffected.
12	GND	Ground.
13	MCLK	Master Clock—active high input from MicroController used to strobe data into data latches from the $\overline{IV}$ bus and bits UD0-UD3 of the UD bus; MCLK also synchronizes $\overline{IV}$ address selection.
14	SC	Select Command—active high input from MicroController to enable $\overline{IV}$ address input from the $\overline{IV}$ bus for device selection.
15	WC	Write Command—active high from MicroController to enable the writing of data into the data latches from the $\overline{IV}$ bus, provided $\overline{UIC}$ is not low.
16-23	$\overline{IV0}$-$\overline{IV7}$	Interface Vector Input/Output Bus —three-state, bidirectional, data bus; $\overline{IV0}$ corresponds to UD0.
24	V_{cc}	Supply Voltage.

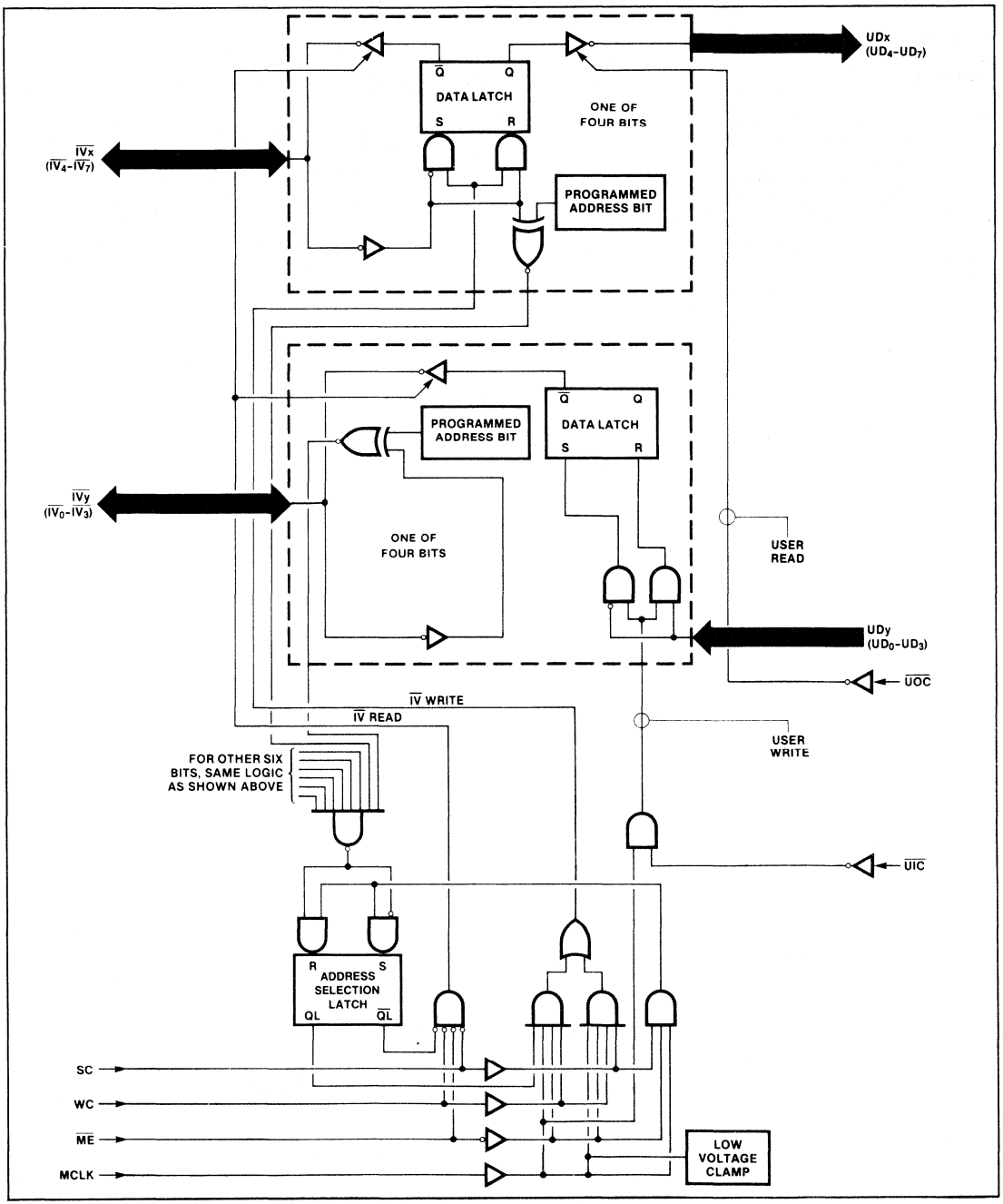


Figure 1. Logic Diagram for 8X382 I/O Port

FUNCTIONAL OPERATION

UD Bus Control

As shown in Table 1, the User Data-bus interface is controlled by the $\overline{UIC}$ and $\overline{UOC}$ inputs. Data input to UD0-UD3 is synchronous with MCLK, that is, with $\overline{UIC}$ low, information is written into the data latches only when MCLK is high. The output drivers of UD4-UD7 bus are enabled when $\overline{UOC}$ is low

Table 1. INPUT/OUTPUT CONTROL OF UD BUS

$\overline{UIC}$	$\overline{UOC}$	MCLK	FUNCTION OF UD BUS	
			UD0-UD3	UD4-UD7
H	L	X	Inactive	Output Data
L	X	H	Input Data	Inactive
L	X	L	Inactive	Inactive
H	H	X	Inactive	Inactive

X = don't care

$\overline{IV}$ Bus Control

Input/output control of the $\overline{IV}$ bus is shown in Table 2; this bus is controlled by SC, WC, $\overline{ME}$, MCLK and the current state of an internal address selection latch. The address selection latch in the I/O port stores the result of the most recent $\overline{IV}$ address selection. The latch is set when the internally preprogrammed address of the port matches the address on the $\overline{IV}$ bus during an address-selection operation (SC=MCLK=High/WC=Low). The latch is cleared when the two 8-bit address patterns are in disagreement. The $\overline{IV}$ bus can transfer data only when the selection latch is set. The MicroController Left Bank ($\overline{LB}$) and Right Bank ($\overline{RB}$) outputs can control the $\overline{ME}$ inputs for two banks of I/O devices, thus, acting as ninth address bit.

Data is written into the data latches of a selected device from the $\overline{IV}$ bus when WC = MCLK = High and $\overline{ME}$ = Low. Output drivers on the $\overline{IV}$ bus of the device with the address latch set are enabled with $\overline{ME}$, WC, and SC low. With SC and WC both high (shaded entry of Table 2), the bit pattern present on $\overline{IV0}$-$\overline{IV7}$ is interpreted as both input data ($\overline{IV4}$-$\overline{IV7}$ only) and $\overline{IV}$ address. The data in $\overline{IV4}$-$\overline{IV7}$ is latched in whether or not the I/O port has been previously selected. If the preprogrammed address of the I/O port matches the bit pattern on $\overline{IV0}$-$\overline{IV7}$ when SC and WC are both high, the selection latch is set; otherwise, it is reset. (Note. *The MicroController never drives both SC and WC high at the same time.*)

Table 2. INPUT/OUTPUT CONTROL OF IV BUS

$\overline{ME}$	SC	WC	MCLK	SEL. LATCH	FUNCTION OF $\overline{IV}$ BUS
L	L	L	X	Set	Output Data
L	L	H	H	Set	Input Data ($\overline{IV4}$-$\overline{IV7}$ only)
L	H	L	H	X	Input Address*
L	H	H	H	X	Input Data ($\overline{IV4}$-$\overline{IV7}$ only) and address*
L	X	H	L	X	Inactive
L	H	X	L	X	Inactive
L	L	X	X	Not set	Inactive
H	X	X	X	X	Inactive

X don't care * Selection latch is updated

Bus Logic Levels

Data written into the I/O port from either bus will appear inverted when read from the other bus. Data written into either bus will not be inverted when read from the same bus. (Note. *A logic "1" in MicroController software corresponds to a high level on the UD bus even though the IV bus is inverted).* The 8X382 wakes up in the unselected state with all data bits latched at the "logic 1" level (UD bus outputs high if enabled).

ADDRESS PROGRAMMING AND ADDRESS PROTECT

Programming Procedures

The 8X382 can be programmed to respond to any address within a range of 0_{10} through 255_{10}. In an unprogrammed state, low level (≤0.8V) inputs on all $\overline{IV}$ bus lines (address 255_{10}) will select the device. To program a given address bit to match a high level (≥2.0V) input on the corresponding $\overline{IV}$ pin (a logical "0" to the MicroController), the counterpart UD-bus pin must be pulsed according to Table 3 and the following procedures:

Step 1: Set all control inputs to the inactive state—$\overline{UIC}$ = $\overline{UOC}$ = $\overline{ME}$ = V_{CC} and SC = WC = MCLK = GND; leave the UD and $\overline{IV}$ bus pins open.

Step 2: Increase V_{CC} to V_{CCP}.

Step 3: After V_{CC} has stabilized, apply a single programming pulse (Figure 2) to the user-bus bit that corresponds to the desired high-level $\overline{IV}$ address bit. The I/O port is programmed from the user bus (UD0-UD7) for addressing from the MicroController bus ($\overline{IV0}$-$\overline{IV7}$).

Table 3. PROGRAMMING SPECIFICATIONS

PARAMETERS	LIMITS			UNITS
	Min	Typ	Max	
V_{CCP} — Programming supply voltage:				
Address	8.75	9.0	9.25	V
Protect		0		V
Maximum time V_{CCP} >5.25V			1.0	Sec
Programming voltage:				
Address	8.75	9.0	9.25	V
Protect	8.75		9.25	V
Programming current:				
Address			5	mA
Protect			50	mA
t_r — Programming pulse rise time:				
Address	10		100	μS
Protect	10		100	μS
t_w — Programming pulse width	0.5		1.0	mS

Step 4: Return V_{CC} to 0-volts. (Note. *If the programming of all address bits is completed in less than 1-second, V_{CC} can remain at 9.0-volts for the required interval of time.*)

Step 5: Steps 1 through 3 are applicable to the programming of each address bit that requires a high-level $\overline{IV}$ match.

Step 6: To verify that the address is properly programmed, return V_{CC} to +5V, set $\overline{IV0}$-$\overline{IV7}$ to the desired (inverted) binary address pattern, set $\overline{ME}$ = WC = Low and SC = MCLK = High. If there are no programming errors, subsequent data written from $\overline{IV4}$-$\overline{IV7}$ (WC = High) will appear inverted on UD4-UD7.

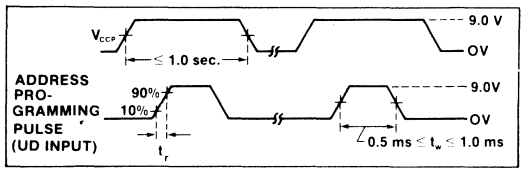


Figure 2. Address Programming Pulse

Address Protect

After programming the I/O Port, steps should be taken to isolate the address circuits and make these circuits permanently immune to further change.

Step 1: Set V_{CC} and all control inputs to 0-volts (V_{CC} = $\overline{UIC}$ = $\overline{UOC}$ = $\overline{ME}$ = SC = WC = MCLK = GND = 0.0V); $\overline{IV0}$-$\overline{IV7}$ = open circuit.

Step 2: Taking one pin at a time, apply a protect programming pulse (Figure 3) to each user-bus bit (UD0-UD7)—refer to Table 3 for min/max specifications pertaining to voltage and current.

Step 3: Verify that the address circuits for each bit is isolated by applying 9-volts, in turn, to each user-bus pin (UD0-UD7) and measuring less than 200 microamperes of input current. (Note. *Setup conditions are the same as those in Step 1.*)

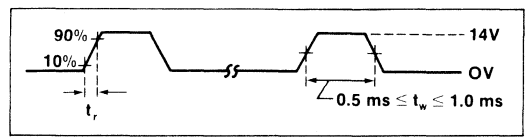


Figure 3. Protect Programming Pulse

DC ELECTRICAL CHARACTERISTICS

COMMERCIAL: 4.75V ≤ V_{CC} ≤ 5.25V, 0°C ≤ T_A ≤ 70°C
MILITARY: 4.5V ≤ V_{CC} ≤ 5.5V, -55°C ≤ T_C ≤ 125°C

ABSOLUTE MAXIMUM RATINGS

PARAMETER		RATING	UNIT
V_{CC}	Power supply voltage[3]	+7	Vdc
V_{IN}	Input voltage[3]	+5.5	Vdc
T_{STG}	Storage temperature range	-65 to +150	°C

PARAMETER		TEST CONDITIONS	LIMITS (COMMERCIAL)			LIMITS (MILITARY)			UNIT
			Min	Typ	Max	Min	Typ	Max	
V_{CC}	Supply Voltage		4.75	5	5.25	4.5	5	5.5	V
V_{IH}	High Level Input Voltage		2.0			2.0			V
V_{IL}	Low Level Input Voltage				0.8			0.8	V
V_{CL}	Input Clamp Voltage	V_{CC} = Min; I_I = -10mA			-1.5			-1.5	V
I_{IH}	High Level Input Current[1]	V_{CC} = Max; V_{IH} = 2.7V		5.0	100		5.0	100	μA
I_{IL}	Low Level Input Current[1]	V_{CC} = Max; V_{IL} = 0.5V		-350	-550		-350	-550	μA
I_{OZH}	High-Z State Output Current—High Level[4]	V_{CC} = Max; V_{OH} = 2.5V			100			100	μA
I_{OZL}	High-Z State Output Current—Low Level[4]	V_{CC} = Max; V_{OL} = 0.5V			-100			-100	μA
V_{OL}	Low Level Output Voltage $\overline{IV}$ Bus ($\overline{IV0}$-$\overline{IV7}$)	V_{CC} = Min; I_{OL} = 16mA			0.55			0.55	V
	User Bus (UD4-UD7)	V_{CC} = Min; I_{OL} = 24mA			0.55			0.55	V
V_{OH}	High Level Output Voltage	V_{CC} = Min; I_{OH} = -3.2mA	2.4			2.4			V
I_{OS}	Short Circuit Output Current[2] $\overline{IV}$ Bus ($\overline{IV0}$-$\overline{IV7}$)	V_{CC} = Max	-20			-20			mA
	UD Bus (UD4-UD7)	V_{CC} = Max	-10			-10			mA
I_{CC}	Supply Current	V_{CC} = Max; $\overline{ME}$ = $\overline{UOC}$ = V_{CC}		90	150		90	150	mA

Notes:
1. The input current includes the Three-state leakage current of the output driver on the data lines.
2. Only one output may be shorted at a time.
3. These limits do not apply during address programming.
4. Applies only to pins UD4-UD7.

AC ELECTRICAL CHARACTERISTICS

COMMERCIAL: $4.75V \leq V_{CC} \leq 5.25V$, $0°C \leq T_A \leq 70°C$

MILITARY: $4.5V \leq V_{CC} \leq 5.5V$, $-55°C \leq T_C \leq 125°C$

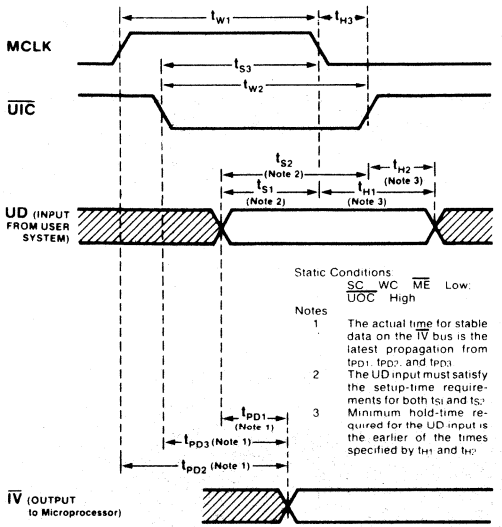
LOADING: See TEST LOADING CIRCUITS

PARAMETER	REFERENCES[1] FROM	REFERENCES[1] TO	TEST CONDITIONS	LIMITS (COMMERCIAL) Min	Typ	Max	LIMITS (MILITARY) Min	Typ	Max	UNIT
Pulse Widths:										
t_{W1} Clock High	↑MCLK	↓MCLK		35			35			ns
t_{W2} User Input Control	↓$\overline{UIC}$	↑$\overline{UIC}$	MCLK = High	35			35			ns
Propagation Delays:										
t_{PD1} UD Propagation Delay	UD_{0-3}	$\overline{IV}_{0-3}$	MCLK = High SC = WC = $\overline{ME}$ = $\overline{UIC}$ = Low			30			30	ns
t_{PD2} UD Clock Delay	↑MCLK	$\overline{IV}_{0-3}$	UD_{0-3} = Stable; MCLK = High; SC = WC = $\overline{ME}$ = $\overline{UIC}$ = Low			50			50	ns
t_{PD3} UD Input Delay	↓$\overline{UIC}$	$\overline{IV}_{0-3}$	UD_{0-3} = Stable; MCLK = High; SC = WC = $\overline{ME}$ = Low			50			50	ns
t_{PD4} $\overline{IV}$ Data Propagation Delay	$\overline{IV}_{4-7}$	UD_{4-7}	MCLK = WC = High; $\overline{ME}$ = $\overline{UOC}$ = SC = Low			45			45	ns
t_{PD5} $\overline{IV}$ Data Clock Delay	↑MCLK	UD_{4-7}	$\overline{IV}_{4-7}$ = Stable; WC = High; $\overline{ME}$ = $\overline{UOC}$ = SC = Low			55			55	ns
Output Enable Timing:										
t_{OE1} UD Output Enable	↓$\overline{UOC}$	UD_{4-7}				30			30	ns
t_{OE3} $\overline{IV}$ Data Master Enable	↓$\overline{ME}$	$\overline{IV}$	WC = SC = Low			22			25	ns
t_{OE5} $\overline{IV}$ Data Write Recovery	↓WC	$\overline{IV}$	SC = $\overline{ME}$ = Low			25			25	ns
t_{OE6} $\overline{IV}$ Data Select Recovery	↓SC	$\overline{IV}$	WC = $\overline{ME}$ = Low			25			25	ns
Output Disable Timing:										
t_{OD1} UD Output Disable	↑$\overline{UOC}$	UD_{4-7}				25			25	ns
t_{OD3}[2] $\overline{IV}$ Data Master Disable	↑$\overline{ME}$	$\overline{IV}$	WC = SC = Low			25			25	ns
t_{OD5}[2] $\overline{IV}$ Data Write Override	↑WC	$\overline{IV}$	SC = $\overline{ME}$ = Low			20			20	ns
t_{OD6}[2] $\overline{IV}$ Data Select Override	↑SC	$\overline{IV}$	WC = $\overline{ME}$ = Low			20			20	ns
Setup Times:										
t_{S1} UD Clock Setup Time	UD_{0-3}	↓MCLK	$\overline{UIC}$ = Low	15			15			ns
t_{S2} UD Control Setup Time	UD_{0-3}	↑$\overline{UIC}$	MCLK = High	15			15			ns
t_{S3} User Input Control Setup Time	↓$\overline{UIC}$	↓MCLK		25			25			ns
t_{S4} $\overline{IV}$ Data Setup Time	$\overline{IV}$	↓MCLK	WC = High or SC = High; $\overline{ME}$ = Low;	35			35			ns
t_{S5}[3] $\overline{IV}$ Master Enable Setup Time	↓$\overline{ME}$	↓MCLK	WC = High or SC = High	30			30			ns
t_{S6} $\overline{IV}$ Write Control Setup Time	↑WC	↓MCLK	SC = $\overline{ME}$ = Low;	30			30			ns
t_{S7} $\overline{IV}$ Select Control Setup Time	↑SC	↓MCLK	WC = $\overline{ME}$ = Low	30			30			ns

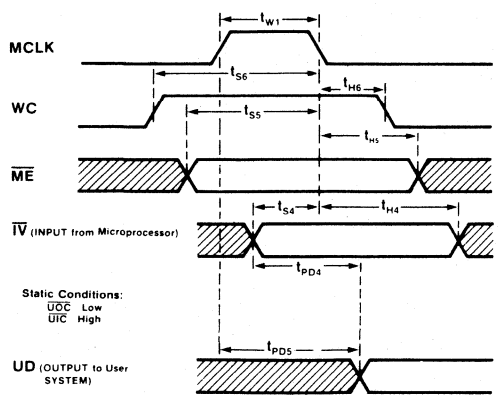
AC ELECTRICAL CHARACTERISTICS (Cont'd)

PARAMETER	REFERENCES		TEST CONDITIONS	LIMITS (COMMERCIAL)			LIMITS (MILITARY)			UNIT
	FROM	TO		Min	Typ	Max	Min	Typ	Max	
Hold Times:										
t_{H1} UD Clock Hold Time	↓MCLK	UD	$\overline{UIC}$ = Low	15			15			ns
t_{H2} UD Control Hold Time	↑$\overline{UIC}$	UD	MCLK = High	15			15			ns
t_{H3} User Input Control Hold Time	↓MCLK	↑$\overline{UIC}$		0			0			ns
t_{H4} Data Hold Time	↓MCLK	$\overline{IV}$	$\overline{WC}$ = High or SC = High; $\overline{ME}$ = Low	5			5			ns
t_{H5}[3] Master Enable Hold Time	↓MCLK	↑$\overline{ME}$	WC = High or SC = High;	0			0			ns
t_{H6} $\overline{IV}$ Write Control Hold Time	↑MCLK	↓WC	SC = $\overline{ME}$ = Low	0			0			ns
t_{H7} $\overline{IV}$ Select Control Hold Time	↑MCLK	↓SC	WC = $\overline{ME}$ = Low	0			0			ns

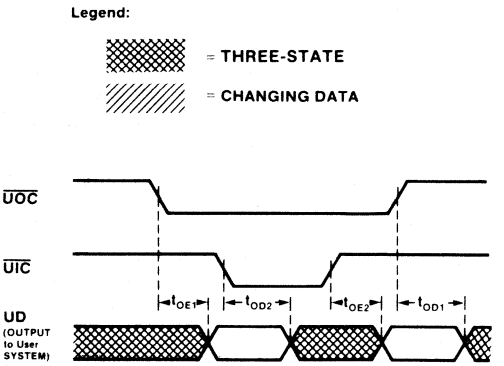
Notes:
1. All measurements to the $\overline{IV}$ bus assumes the address selection latch is set.
2. These parameters are measured with a capacitive loading of 50 *pf* and represent the output driver turn-off time.
3. If $\overline{ME}$ is to be high (inactive), it must be setup before the rising edge and held after the falling edge of MCLK to avoid unintended writing into or selection of the I/O port.



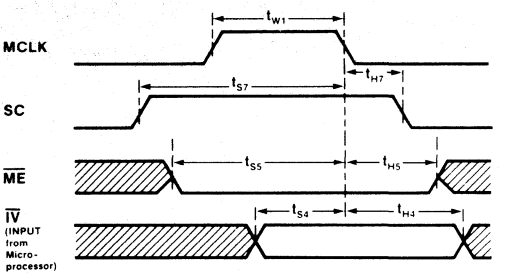
a. User Data Input Timing



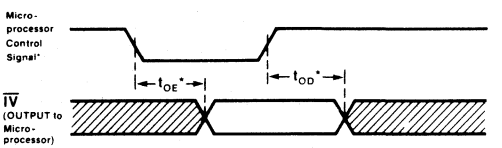
c. MicroController Write Cycle Timing



b. User Data Output Timing



d. MicroController Select Cycle Timing



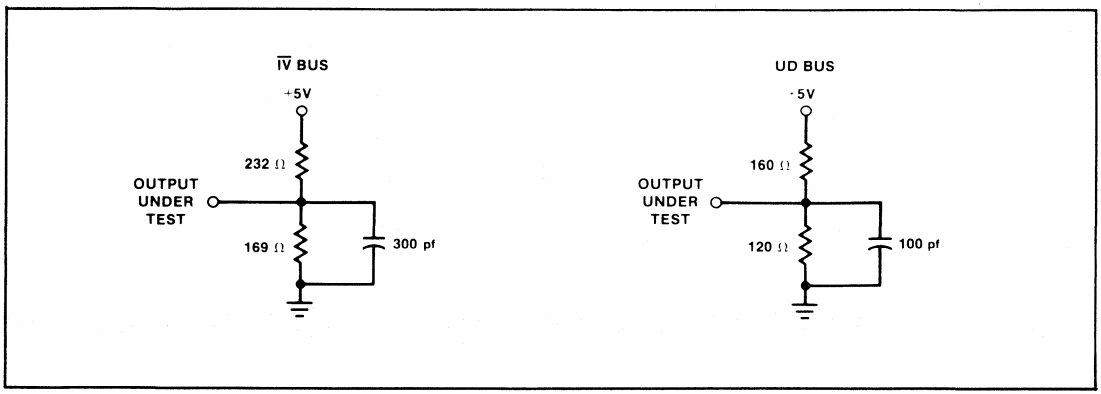
*PARAMETER KEY

MICROPROCESSOR CONTROL SIGNAL	AC TIMING PARAMETERS	STATIC CONDITIONS
ME	t_{OE3} t_{OD3}	SC = WC = LOW
WC	t_{OE5} t_{OD5}	SC = ME = LOW
SC	t_{OE6} t_{OD6}	WC = ME LOW

e. MicroController Output Enable Timing

Figure 4. Timing Diagrams

TEST LOADING CIRCUITS



APPLICATIONS

When compared to other MicroController ports in the 8X370 series, the 8X382 has some unique features that provide real design advantages in certain applications. Connection of the I/O port to the MicroController is simple and straightforward in that like pin names are tied together. The system designer must also decide on which bank of the MicroController to place the 8X382 and then connect the $\overline{ME}$ pin of the port to either the $\overline{LB}$ (Left Bank) or $\overline{RB}$ (Right Bank) of the MicroController.

The 8X382 is unique because it can be used for both dedicated input and output operations. In the system shown below, the user interface requires nine (9) dedicated inputs and eleven (11) dedicated outputs. Observe that by using an 8X382, the problem is solved by three devices, whereas, four 8X372 ports are required for the same solution.

Another important use of the 8X382 is in implementing a handshake interface. Since both input and output bits reside in the same port, I/O operations can be performed without port re-addressing. Users may also find the 8X382 an advantage in the layout of Printed Circuit boards, since random control/status signals can be grouped within the same device position.

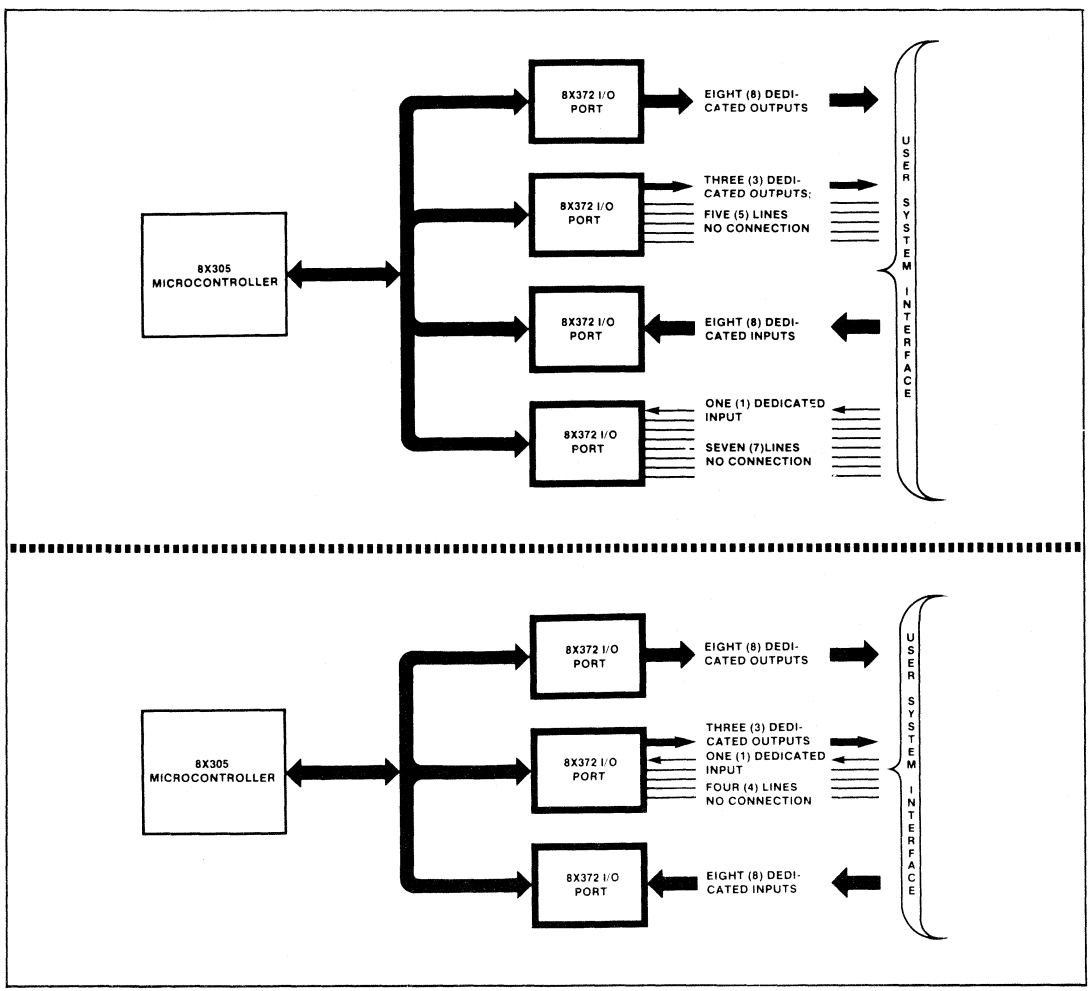


Figure 5. Logic Diagram for 8X382 I/O Port

Signetics

8X401
Microcontroller

Product Specification

Microprocessor Products

DESCRIPTION

The Signetics 8X401 Microcontroller is a very high-speed bipolar microprocessor implemented with internal ECL technology. The 8X401 Microcontroller combines speed, flexibility, and a bit-oriented instruction set to accommodate many sophisticated applications. It excels in systems that require high-speed bit or byte manipulations, such as high-speed controllers and data communications.

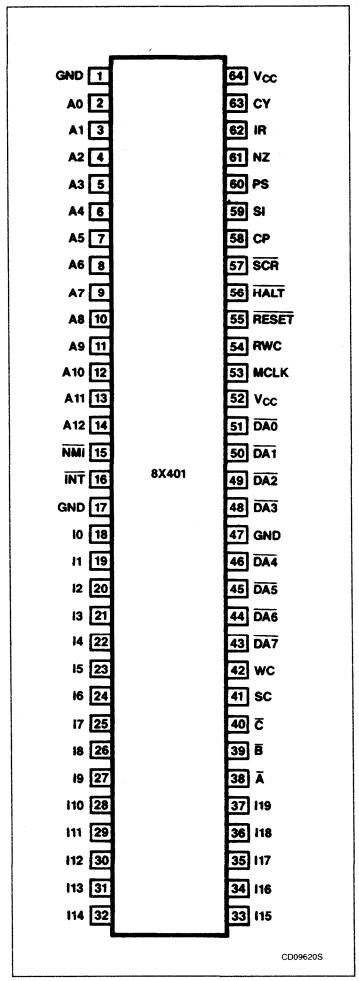
The 8X401 can fetch, execute, and generate the next instruction address for a 20-bit instruction in a minimum of 150ns. Within one instruction cycle, the 8X401 can be programmed to input, right-rotate and mask single or multiple bit subfields, perform an ALU operation, left-rotate, merge the subfield into the destination, and output.

To interface with program memory, the 8X401 uses a 13-bit address bus and a 20-bit instruction bus. An 8-bit bidirectional data/address bus, and an I/O control and timing bus is used to access external peripheral devices.

FEATURES

- **Fetches and executes all instructions in a minimum of 150ns**
- **Bit manipulation-oriented instruction set**
- **Separate buses for instruction, instruction address and I/O**
- **Sixteen 8-bit registers**
- **On-chip interrupt control**
- **TTL compatible I/O**
- **Single +5V supply**
- **0.9-inch 64-pin DIP, 68-pin plastic leaded chip carrier**
- **Two user-definable status flags**
- **Single TTL clock input**
- **Three independent I/O banks**
- **On-board control sequencer**
- **On-chip subroutine capabilities**
- **Fixed instruction set — 32 instructions**
- **Complete development support**

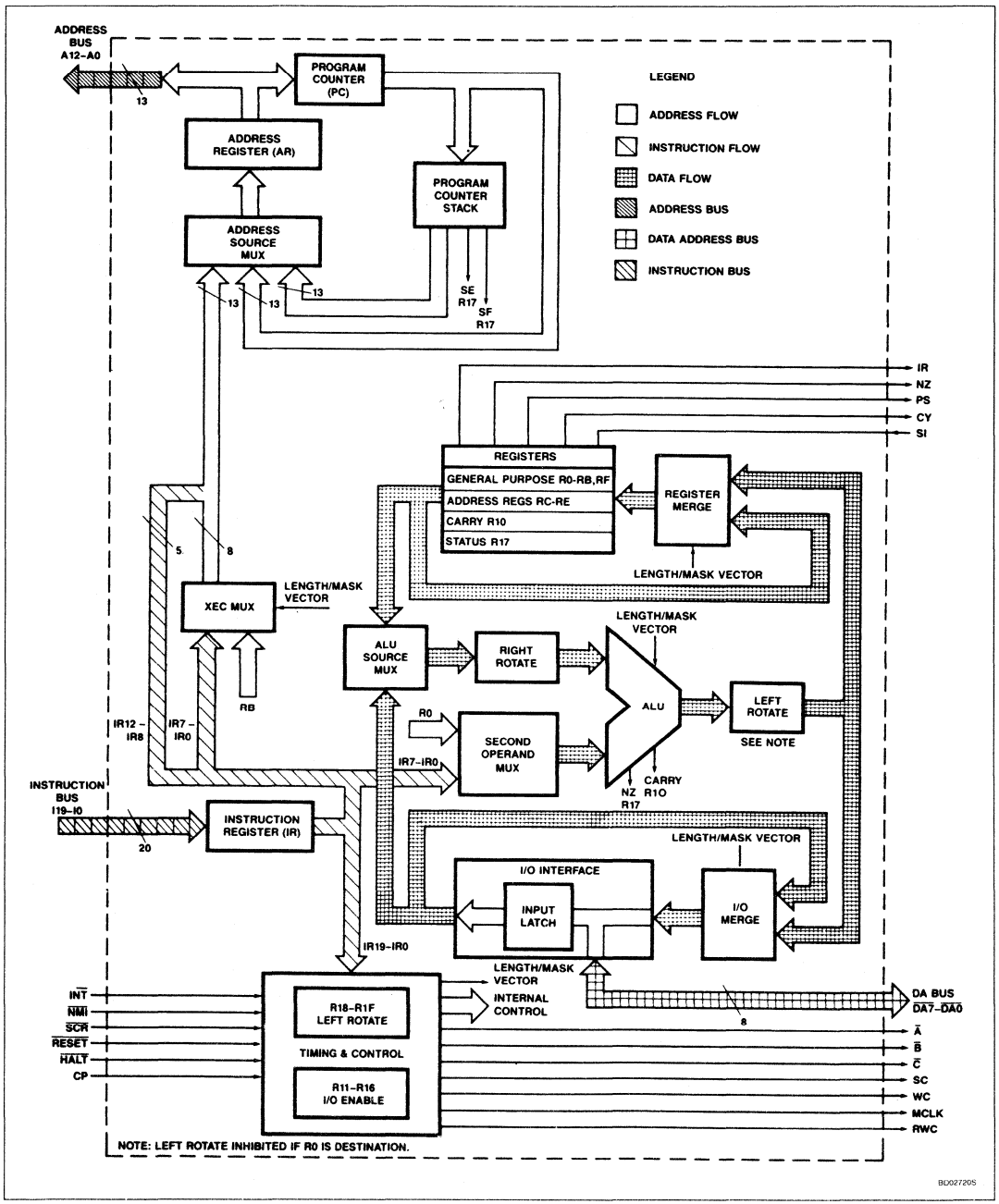
PIN CONFIGURATION



ORDERING INFORMATION

DESCRIPTION	ORDER CODE
64-pin CERAMIC DIP 900 Mil Wide	N8X401I

BLOCK DIAGRAM OF THE 8X401



PIN DESCRIPTION

PIN NO.	IDENTIFIER	FUNCTION
1, 17, 47	GND	Ground.
2 – 14	A12 – A0	**Program Address Lines:** These active-high outputs permit direct addressing of up to 8192 locations of program storage; A0 is LSB.
15	$\overline{NMI}$	**Non-Maskable Interrupt:** The falling edge of this active-low input pin generates a non-maskable interrupt.
16	$\overline{INT}$	**Interrupt:** This active-low input pin is tested during the fourth quarter of each instruction cycle. If an interrupt is indicated and if interrupts are enabled, the address of the next instruction that was to be executed is stored onto the program counter stack before the interrupt is serviced.
18 – 37	I19 – I0	**Instruction Lines:** These active-high input lines receive 20-bit instructions from program storage; I0 is LSB.
38	$\overline{A}$	**Bank A:** When low, devices connected to bank A are accessed. (Note: Typically, the A signal is tied to the $\overline{ME}$ input pin of I/O peripherals.)
39	$\overline{B}$	**Bank B:** When low, devices connected to bank B are accessed. (Note: Typically, the B signal is tied to the $\overline{ME}$ input pin of I/O peripherals.)
40	$\overline{C}$	**Bank C:** When low, devices connected to bank C are accessed. (Note: Typically, the C signal is tied to the $\overline{ME}$ input pin of I/O peripherals.)
41	SC	**Select Control:** When high, an address is being output on pins $\overline{DA7}$ through $\overline{DA0}$.
42	WC	**Write Control:** When high, data is being output on pins $\overline{DA7}$ through $\overline{DA0}$.
43 – 46 48 – 51	$\overline{DA7}$ – $\overline{DA0}$	**Data Address Bus:** These active-low, bidirectional, three-state lines are used for I/O; $\overline{DA0}$ is LSB.
52, 64	V_{CC}	+5V power supply.
53	MCLK	**Master Clock:** This active-high output signal is used to strobe data into data peripherals for clocking I/O devices and/or synchronization of external logic. MCLK is active-high in the fourth quarter cycle.
54	RWC	**Read/Write Clock:** This active-high output signal is used for synchronization of external logic and is active-high during the third and fourth quarter cycles.
55	$\overline{RESET}$	**Reset:** The $\overline{RESET}$ input pin is used to initialize the 8X401.
56	$\overline{HALT}$	**Halt:** The $\overline{HALT}$ input is sampled during the first quarter cycle of each instruction cycle. When the $\overline{HALT}$ input is low, the instruction cycle is not executed.
57	$\overline{SCR}$	**Slow Clock Request:** This active-low control input is sampled during the first quarter cycle of each instruction. When $\overline{SCR}$ is asserted, it will cause the current instruction to be executed at half of the normal clock rate. This control input is necessary to accommodate I/O devices that cannot operate at the 8X401's full speed, without having to continuously run the 8X401 at half speed.
58	CP	**Clock Pulse:** Each 8X401 quarter cycle will correspond to one full cycle of the clock pulse.
59	SI	**Status Input:** The value of the SI pin during the fourth quarter cycle is transferred to SI bit in the status register.
60	PS	**Programmable Status:** The programmable status pin is controlled entirely by the user program.
61	NZ	**Non-Zero:** The NZ bit of the status register is reflected on this pin.
62	IR	**Interrupt Receivable:** The IR pin indicates whether an interrupt applied at any point in time will be serviced. Interrupts are receivable when the interrupt mask (status register, bit 0) is clear and the stack is not full (IM = 0 and SF = 0).
63	CY	**Carry:** Carry bit from R10 is output on this pin.

SMALL SYSTEM CONFIGURATION

The system hookup shown on the next page, although of the simplest form, provides a fundamental example of the 8X401 Microcontroller and compatible peripheral relationships. As shown, the 8X401 can directly address up to 8K locations of program storage.

Each of the three bank pins ($\overline{A}$, $\overline{B}$, or $\overline{C}$) are capable of uniquely addressing 256 input/output locations via the Data Address bus ($\overline{DA7}$ – $\overline{DA0}$).

The addressable locations for each bank can be used in a variety of ways. The hookup shown below is just one method of implementation.

When a particular bank signal is asserted, that bank is enabled and any one of 256 locations on that bank can be accessed for input/output operations.

PROGRAM STORAGE INTERFACE

As shown in the 8X401 small system hookup, program memory is connected to output address lines A12 through A0 (A0 = LSB) and input instruction lines I19 through I0 (I0 = LSB). An address output on A12 – A0 identifies one 20-bit instruction word in program memory. The program memory outputs an instruction word on I19 – I0 which defines the microcontroller operation which is to follow. One instruction word equals one completed operation. Any TTL-compatible memory can be used for program storage, provided the worst-case access time is compatible with the instruction cycle time used for the application. See timing section for appropriate calculations.

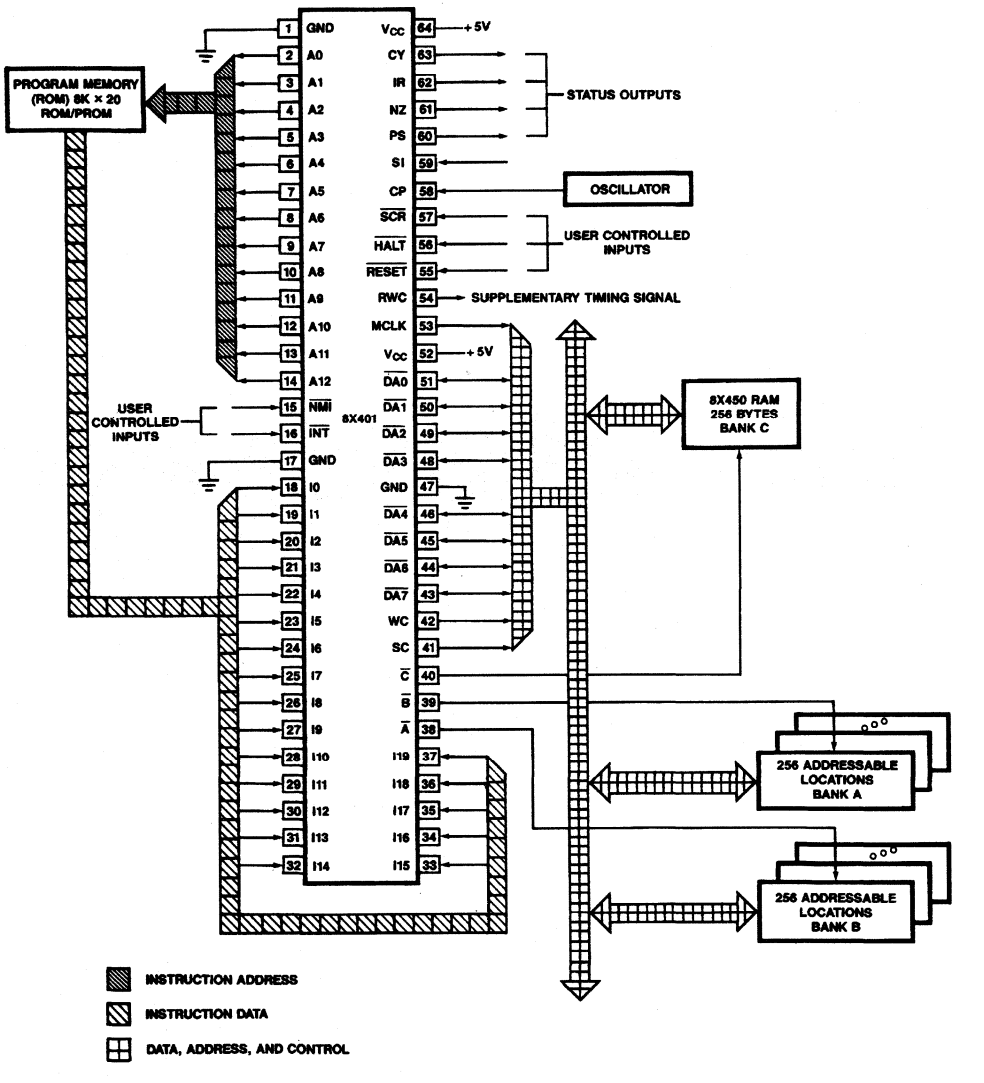
I/O INTERFACE AND CONTROL

The Data Address (DA) bus is an 8-bit bidirectional I/O bus which provides a communication link between the 8X401 and the three banks of the I/O devices. The $\overline{A}$ (A bank), $\overline{B}$ (B bank), and $\overline{C}$ (C bank) control signals identify which bank is enabled. When all three banks go high (inactive), neither bank is enabled and the DA bus is inactive (three-state). A functional analysis of the three bank signals is shown below:

A	B	C	FUNCTION
Low	Low	Low	This state is not generated by the 8X401.
Low	High	High	Enable A bank devices.
High	Low	High	Enable B bank devices.
High	High	Low	Enable C bank devices.
High	High	High	Disable all devices; DA bus is three-state

Both data and I/O address information are multiplexed on the DA bus. The SC (Select Command) and WC (Write Command) signals distinguish between data and I/O address information as shown in the table below. Although the table shows bank A only, the same conditions apply to banks B and C.

BANK A	SC	WC	FUNCTION
High	Low	Low	DA bus is three-state and not looking for input data.
Low	Low	Low	The DA bus is reading input data.
Low	Low	High	Data is being output.
Low	High	Low	Address is being output.
X	High	High	This condition is never generated.



8X401 Small System Hookup

DATA PROCESSING

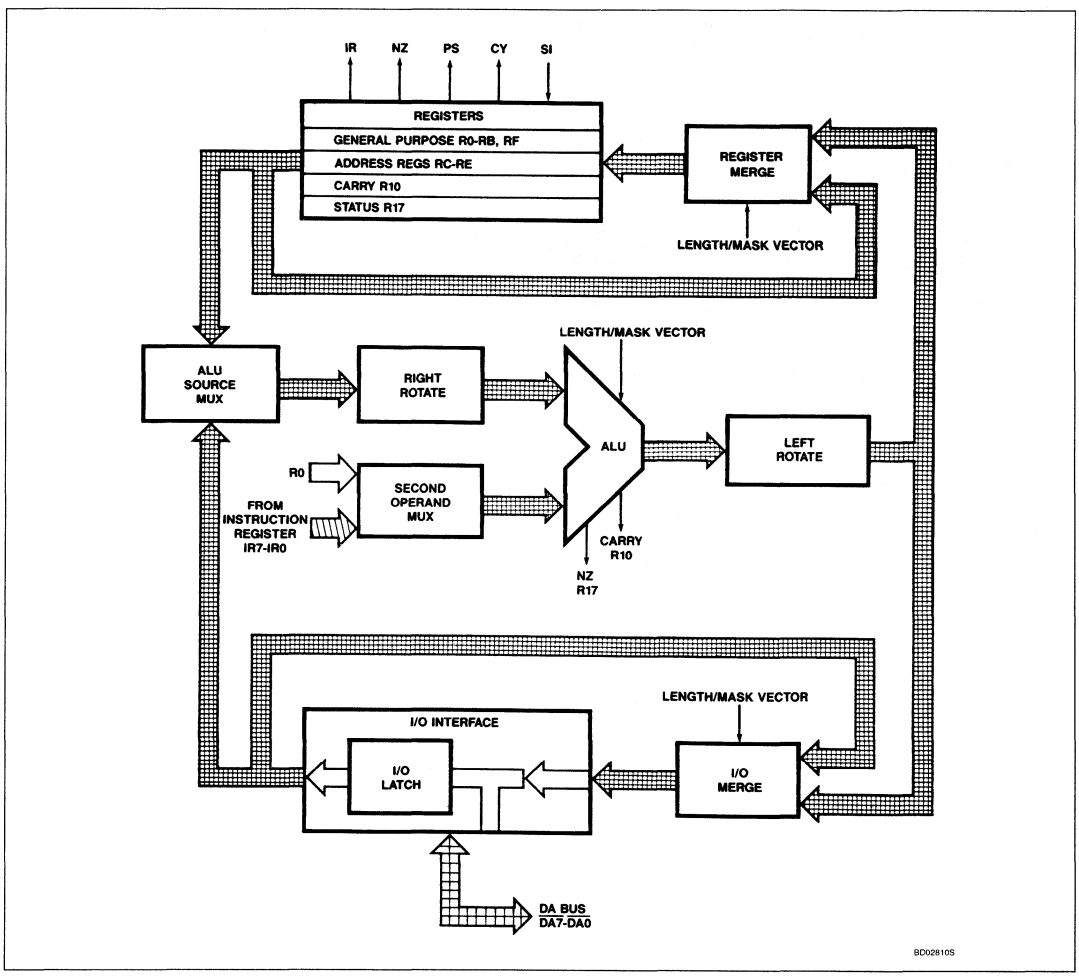
The data processing section of the 8X401 consists of a number of logical subsections. In order of processing, the data sees the right rotator, the ALU, the left rotator, and the merge circuits. Data sources and destinations can be various on-chip registers, the bidirectional DA bus, or immediate subfields. The data processing paths are shown below.

DATA REGISTERS

General-Purpose Storage

There are 13 source/destination general-purpose registers available on the 8X401. Three of these registers, specifically Registers 0, B, and F, have other special functions in addition to being general purpose. A summary of the registers is listed below:

- **R0 (Auxiliary Register)** — Register 0 is also used as the implied second operand for two operand instructions (ADD, ADD with CARRY, XOR, AND). The primary operand is specified in the source field of the instruction word and the AUX register is the implied second operand. Prior to performing arithmetic or logical operations (other than the IMMEDIATE operation), it is assumed that R0 contains the appropriate data. In order to reduce the possibility of erroneous results and to minimize the number of instructions required to transfer a right-justified second operand into the AUX, the left-rotate and merge functions are inhibited when specifying the AUX as a destination address. This allows subfields from any internal register or I/O bank to be transferred to the AUX with the subfield LSB right-justified and unspecified bits set to zero.
- **R1 through RA** — These 10 addresses specify general-purpose, on-chip storage registers.
- **RB** — Register B is also used as the implied source for the XEC instruction.
- **RF** — Register F is also used as the implied destination for the XOR IMMEDIATE and AND IMMEDIATE instruction classes.



Enabled I/O Addresses

These three registers (RC RD, and RE) always contain the address of the most recently enabled I/O device for each of the three I/O banks. When register C, D, or E is the specified destination address, the destination data is sent to both the on-chip register and the corresponding bank on the DA bus. The relationship between the register addresses and I/O banks is shown below:

REGISTER	BANK
C	A
D	B
E	C

When these registers are specified as a destination address, the L field must be set to 0 (full 8-bit operation). Also, note that registers C, D, and E may not be used with the XMIT 5 or ADD IMMEDIATE 5 instructions.

Carry

Register 10 contains the Carry bit. Bit position 0 (LSB) is the Carry bit, and positions one through seven are always zero. The Carry bit is updated each time an ADD, ADD IMMEDIATE, or ADD WITH CARRY instruction is performed. When specifying address 10 as a destination, only bit 0 (the Carry bit) will be written to. Data written to the Carry bit will be the LSB of the right-rotated data after any specified operation.

When the Carry register is the explicit destination of any ADD instruction, it will contain the carry resulting from the add operation rather than the LSB of the sum. Carry can also be affected via the Return and Set Carry or Return and Clear Carry instructions.

Status Register

This address specifies the current condition of the 8X401 system. The status register may be either a source or destination; however, certain bits in the status register are read-only. Four status outputs are available on 8X401 pins. They are NZ (Not Equal to Zero), PS (Programmable Status), IR (Interrupts Receivable), and Carry (R10, bit 0). The IR pin goes high when the interrupt mask is clear and the stack is not full. The IR output is updated during the 4th quarter cycle. The following descriptions define the bits within the status register.

Bit 0: (IM) — This bit represents the Interrupt Mask control. When IM is set, the interrupt is inhibited. This bit is set automatically by a response from a standard or non-maskable interrupt, or RESET. IM can also be set or cleared by a write to the status register.

Bit 1: (NZ) — This bit is set whenever the ALU output data is not equal to zero after any of the following instructions: MOVE, ADD, AND, XOR, ADD IMMEDIATE, AND IMMEDIATE, XOR IMMEDIATE, or ADD WITH CARRY. NZ can also be written to directly when specified in the destination field. This operation will negate and take priority over the normal setting by the ALU output. NZ is not affected after an XMIT instruction, or after a write to R17.

Bit 2: (PS) — This is the Programmable Status bit. The contents are reflected on the PS output pin. This status is controlled entirely by the user program.

Bits 3 and 4: (UF0, UF1) — These two bits represent user flags and have no assigned functions. They can be used as 1-bit internal flags and are entirely under control of the user program.

Bit 5: (SI) — This bit reflects the state of the status input pin. This read-only bit is updated during the 4th quarter cycle.

Bits 6 and 7: (SE, SF) — These read-only bits indicate Stack Empty and Stack Full, respectively. The bits are updated during the 3rd quarter cycle within the instruction that alters the stack status.

INSTRUCTION WORD (See Table 3)

Operations Code Field — The 4-bit opcode specifies one of 16 classes of instructions. Some instructions require two additional subopcode fields, X and XS. Variations and interpretations are displayed in Table 2.

Source (S) and Destination (D) Fields — The 5-bit "S" and "D" fields specify the source and destination, respectively, for the operation that is defined by the opcode. The "S" and/or "D" fields specify an internal 8X401 register or a variable length field from an I/O device. Hexadecimal values and source/destination field assignments for all internal registers are shown in Table 1.

When RC-RE (banks A, B, or C, respectively) are specified as the destination, the data is output onto the DA bus using the specified bank. The data is also stored in the specified register and may be later accessed as source data.

Rotate (R) and Length (L) Fields — The R field is used in conjunction with the L field to define the desired data within a register or I/O device. The source data is right-rotated prior to ALU operations, such that the bit specified by the R field is right-justified. The L field specifies the number of bits of data to be used for the operation. After the ALU operation, the data is left-rotated back to the original position prior to merging the data in the destination register.

When the L field specification is 0 (indicating a full 8-bit operation), the left-rotate is suppressed. This allows byte rotate operations to be performed. The left-rotate is also suppressed when the destination is register 0. This is the AUX register and is used as the implied second operand in certain instructions.

It should also be noted that subfields are defined at the ends of a register; for example, bit positions 1, 0, 7, and 6 constitute a contiguous 4-bit subfield.

Data Field — The data field holds data that can be processed directly from the instruction word.

LEFT-ROTATE OVERRIDE BLOCK

Register addresses 18-1F are destination only and are used to independently control left rotation of data prior to storage in the destination. Specifying 18-1F as a destination causes the data to be returned to the source address.

In order to move a processed subfield within the same register but in different bit positions (the LSB of the contiguous subfield can vary), it is necessary to independently specify the LSB for both the source and destination. The order of operation is as follows:

- Register or I/O source data is right-rotated as specified by the "R" field. Along with the "L" field, the subfield data is defined.
- Subfield data is processed via the ALU.
- Data is left-rotated 0 – 7 bits, depending on the corresponding register addresses 18-1F as specified in the destination field rather than using the "R" field.
- After left-rotation the specified subfield is merged into those bits of the original source data. The unspecified bits of the original source data remain unchanged.
- Result is stored in the register address specified by the source field.

Note that the left-rotate is always inhibited if the "L" field is zero. Also, addresses 18-1F may not be used in the destination field for the XMIT or ADD IMMEDIATE instructions. The destination addresses and corresponding left-rotate values are shown in Table 2.

DA BUS CONTROL BLOCK

Register addresses 11 – 16 are used by the 8X401 to access I/O devices for either a source or destination specified within the instruction. Register addresses 13, 15, and 16 specify banks A, B, and C, respectively, whereas addresses 11, 12, and 14 specify bank pairs AB, CA, and BC, respectively (Table 4). One bank of each pair is known as the preferred bank. The preferred banks for

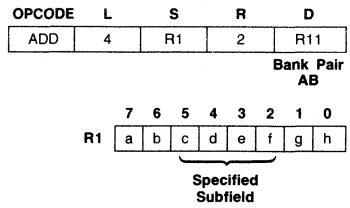
pairs AB, CA, and BC are banks A, C, and B, respectively. The first letter from each bank pair can serve as a mnemonic aid as to which bank is the preferred bank. Having a preferred bank is simply a method of determining which bank to read when an instruction would otherwise indicate that two banks should be read at once.

When used as a source, the appropriate I/O bank is enabled and data is read from the activated I/O device on that bank. The I/O device may have been activated by a previous address select instruction where registers C-E were the specified destination. If a bank pair (addresses 11, 12, or 14) is specified as a source, only data from the preferred bank of that pair will be read in.

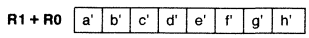
When addresses 11 – 16 are specified as the destination address, the destination data is sent to the DA bus. The Write Control (WC) signal goes high, indicating data (as opposed to an address) is on the DA bus and is to be written to the activated I/O device on the selected bank(s).

When addresses 11 – 16 are specified as the destination and the "L" field is not zero, the following statements apply: If the source is a register and the destination is a single bank, the bank will be read (or the preferred bank of a bank pair will be read) to obtain the data required to perform the merge operation. The result is that processed data from the specified subfield of the source register is returned to that selected field of the destination bank and any bits outside of the specified subfield will be loaded with unprocessed data from the I/O device just read. If the destination is a bank pair, the data from the procedure just described is sent to both banks. Below is an example of the outcome of an ADD instruction with Length = 4, Rotate = 2, Source = R1, and Destination = R11 (bank pair AB).

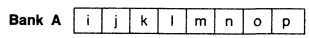
Example 1:



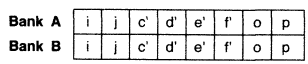
Source register R1. Note specified subfield from L = 4 and R = 2.



Add R1 with R0 (AUX). Result after left-rotate.



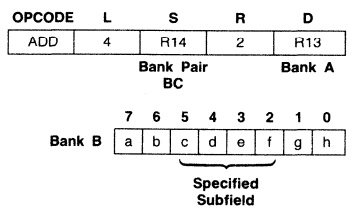
Read preferred bank A of AB.



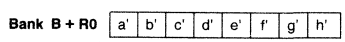
Result after merge. Data put out on both banks A and B.

If, however, the specified source is a bank or bank pair, any unspecified bits will contain unprocessed data from the source I/O device. Below is an example of the outcome with Source = R14 (bank pair BC) and Destination = R13 (bank A).

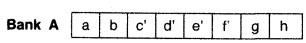
Example 2:



Read Preferred bank B of BC. Note specified subfield from L = 4 and R = 2.



Add bank B data with R0 (AUX). Result after left-rotate.



Result after merge. Unspecified bits contain original source data.

PROGRAM COUNTER STACK

The 8X401 stack is capable of saving up to four return addresses for subroutines and interrupts. Addresses are pushed onto the stack as a result of a call or a maskable or non-maskable interrupt. Addresses are popped from the stack as a result of an unconditional RETURN, a satisfied conditional RETURN, or a Pop Stack and Jump instruction. The status of the stack (whether empty, full, or neither) is available from the SE and SF flags in the status register and the Interrupts Receivable (IR) output pin.

INTERRUPTS

Interrupt ($\overline{INT}$)

The interrupt input is tested once each instruction cycle, during the fourth quarter cycle (see Figure 1). When the interrupt input is taken low and is enabled, the address of the next instruction is pushed onto the program counter stack.

Program flow is transferred to address 2 for the start of the service routine (Figure 2). This is accomplished by inserting a dummy instruction cycle after the interrupt is accepted.

The interrupt mask bit (R17, bit 0) is set automatically as part of the interrupt response.

The Interrupts Receivable (IR) pin indicates whether an interrupt applied at any point in time will be serviced. Interrupts are receivable when the interrupt mask is clear and the stack is not full (IM = 0 and SF = 0).

Non-Maskable Interrupt ($\overline{NMI}$)

The function of the non-maskable interrupt is similar to the standard interrupt, except that the interrupt receivable status has no effect on its operation and the address jumped to is 1 rather than 2 (Figure 2). Address 1 should contain an unconditional JUMP to the start of the NMI service routine. An NMI is triggered by a falling edge on the $\overline{NMI}$ input. The interrupt mask is set to prevent normal interrupts from interfering with the NMI service routine. Note that it may not always be possible to recover from an NMI, since the condition of the interrupt mask prior to the NMI is not known, and the NMI response may overflow the stack.

SLOW CLOCK REQUEST ($\overline{SCR}$)

This control input is sampled during the first quarter cycle of each instruction along with the instruction data. If the input is low, it will cause the current instruction to be executed at half of the normal clock rate. The purpose of this function is to facilitate accesses to I/O devices that cannot operate at the 8X401's

full speed, without the need to run the 8X401 continuously at half speed.

HALT

The HALT input is sampled during the first quarter cycle of each instruction. If the HALT input is low, the instruction cycle is not executed. The MCLK continues to operate normally (high every fourth quarter cycle), even though program execution has ceased. When the HALT input goes high, program execution will resume at the next falling edge of MCLK. The DA bus is also inactive during a HALT operation. Like MCLK, RWC continues to operate during a HALT operation.

RESET LOGIC

The RESET pin is used to initialize the 8X401. When RESET is low, the address outputs (A12 – A0) are high impedance, the stack pointer is set to the top of the stack (empty), MCLK is inhibited, RWC is low, and the interrupt mask (bit 0, register 17) is set.

When RESET is released, the address outputs all go low (program address 0). A dummy instruction cycle occurs to allow time to fetch the first instruction from program storage at address 0. Only MCLK, RWC, and the address bus are in operation during the dummy cycle. The first active instruction cycle will begin following the first MCLK after RESET is released. The instruction at address 0 should be an unconditional jump to the beginning of the main program (which may be proceeded by a power-up sequence to initialize the system (Figure 2).

If RESET is applied during program execution, its effect is immediate. That is, if MCLK is high, it may be prematurely terminated by RESET.

Table 1. Hexadecimal Addresses and Source/Destination Specification

ADDRESS	DESIGNATION	S	D	ADDRESS	DESTINATION	S	D
00	R0 (AUX) — General Purpose[1]	X	X	10	R10 — Carry[6]	X	X
01	R1 — General Purpose	X	X	11	R11 — Bank Access Command (Bank Pair AB)	X	x
02	R2 — General Purpose	X	X	12	R12 — Bank Access Command (Bank Pair CA)	X	x
03	R3 — General Purpose	X	X	13	R13 — Bank Access Command (Bank A)	X	x
04	R4 — General Purpose	X	X	14	R14 — Bank Access Command (Bank Pair BC)	X	x
05	R5 — General Purpose	X	X	15	R15 — Bank Access Command (Bank B)	X	x
06	R6 — General Purpose	X	X	16	R16 — Bank Access Command (Bank C)	X	x
07	R7 — General Purpose	X	X	17	R17 — Status[4]	X	X
08	R8 — General Purpose	X	X	18	R18 — Suppress Left-Rotate[5]		X
09	R9 — General Purpose	X	X	19	R19 — Left-Rotate 1 Place[5]		X
0A	RA — General Purpose	X	X	1A	R1A — Left-Rotate 2 Places[5]		X
0B	RB — General Purpose[2]	X	X	1B	R1B — Left-Rotate 3 Places[5]		X
0C	RC — Address Reg (Bank A)	X	X	1C	R1C — Left-Rotate 4 Places[5]		X
0D	RD — Address Reg (Bank B)	X	X	1D	R1D — Left-Rotate 5 Places[5]		X
0E	RE — Address Reg (Bank C)	X	X	1E	R1E — Left-Rotate 6 Places[5]		X
0F	RF — General Purpose[3]	X	X	1F	R1F — Left-Rotate 7 Places[5]		X

NOTES:
1. Also used as implied second operand for two operand instructions.
2. Also used as implied source for XEC instructions.
3. Also used as implied destination for XOR IMMEDIATE and AND IMMEDIATE instructions.
4. Certain bits in the status register are read-only. (See Status Register within text.)
5. The result is returned to the register address specified by the source field.
6. Carry register, bit 0 is the carry bit. Bits 1 – 7 are always set to zero.

Table 2. Various of Instruction Types

INSTRUCTION TYPE	VARIATION	OPCODE	X	XS	DESCRIPTION
MOVE	MOV	0000	–	–	MOVE
ADD	ADD	0001	–	–	ADD
	ADC	0101	–	–	ADD with CARRY
	AD8	1000	–	–	ADD IMMEDIATE 8
	AD5	1001	–	–	ADD IMMEDIATE 5
AND	AND	0010	–	–	AND
	AN8	1010	–	–	AND IMMEDIATE 8
	AN5	1011	–	–	AND IMMEDIATE 5
XOR	XOR	0011	–	–	Exclusive-OR
	XR8	1100	–	–	Exclusive-OR IMMEDIATE 8
	XR5	1101	–	–	Exclusive-OR IMMEDIATE 5
XEC	XEC	0100	–	–	EXECUTE
XMIT	XT8	0110	–	–	Transmit IMMEDIATE 8
	XT5	0111	–	–	Transmit IMMEDIATE 5
RETURN	RIF NS	1110	000	–	RETURN IF SI = 0
	RIF S	1110	001	–	RETURN IF SI = 1
	RIF NC	1110	010	–	RETURN IF CARRY = 0
	RIF C	1110	011	–	RETURN IF CARRY = 1
	RIF Z	1110	100	–	RETURN IF ALU = 0
	RIF NZ	1110	101	–	RETURN IF ALU ≠ 0
	PSJ	1110	110	–	POP STACK and JUMP
	RTN	1110	111	00	RETURN
	RCC	1110	111	10	RETURN and CLEAR CARRY
	RSC	1110	111	11	RETURN and SET CARRY
JUMP	JIF NS	1111	000	–	JUMP IF SI = 0
	JIF S	1111	001	–	JUMP IF SI = 1
	JIF NC	1111	010	–	JUMP IF CARRY = 0
	JIF C	1111	011	–	JUMP IF CARRY = 1
	JIF Z	1111	100	–	JUMP IF ALU = 0
	JIF NZ	1111	101	–	JUMP IF ALU ≠ 0
	JSR	1111	110	–	JUMP to SUBROUTINE
	JMP	1111	111	–	JUMP

Instruction Set Overview: The 8X401 instruction set is summarized in Table 2. Subsets of each instruction type are grouped together showing the variations of each instruction type. The hardware and software descriptions can be found in the data operations section.

Table 3. 8X401 Instruction Formats

1. Format for the MOVE, ADD, XOR, and ADD with Carry Instructions:

OPCODE	L	SOURCE	R	DESTINATION

2. Format for the XMIT 8 and ADD Immediate 8 Instruction:

OPCODE	L	DESTINATION	DATA8

3. Format for the XMIT 5 and ADD Immediate 5 Instruction:

OPCODE	L	DESTINATION	R	DATA5

4. Format for the AND Immediate 8, and XOR Immediate 8 Instruction:

OPCODE	L	SOURCE	DATA8

5. Format for the AND Immediate 5 and XOR Immediate 5 Instruction:

OPCODE	L	SOURCE	R	DATA5

6. Format for the JUMP, Subroutine Jump, Conditional Jump, and Conditional Return Instruction:

OPCODE	X	ADDRESS

7. Format for the Unconditional Return, Return and Set Carry, and Return and Clear Carry Instruction:

OPCODE	X	XS	(UNUSED

8. Format for the XEC Instruction:

OPCODE	L	ADDRESS

Table 4. I/O Acess Register

REGISTER	INPUT BANK	OUTPUT BANK
11	A	A & B
12	C	C & A
13	A	A
14	B	B & C
15	B	B
16	C	C

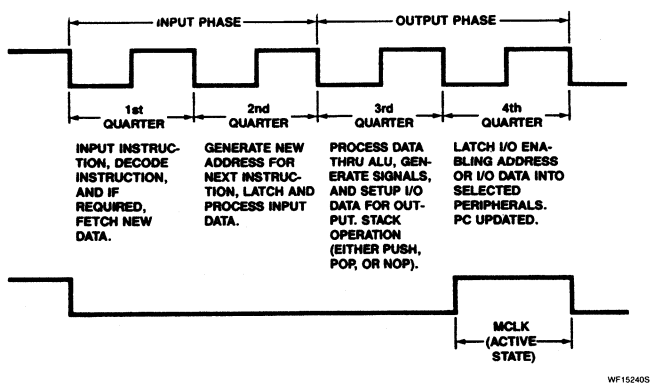


Figure 1. Instruction Cycle and MCLK with: Clock Input = 26.67MHz and Cycle Time = 150ns

DATA OPERATIONS OF THE 8X401 (See Tables 2 and 3)

MOVE, ADD, AND, XOR, ADD with CARRY

SOURCE	PRE-ALU	ALU	POST-ALU	DESTINATION
R0 – R17 as specified by "S" field of instruction.	Right rotate as specified by "R" field of instruction.	Perform appropriate ALU operation. (Note 1)	Left-rotate and mask as specified by the "R" and "L" fields of instruction.	R0 – R17 as specified by the "Destination" field of instruction. (Notes 2 & 3)

MOVE, ADD, AND, XOR, ADD with CARRY (Using left-rotate override R18 – R1F)

SOURCE	PRE-ALU	ALU	POST-ALU	DESTINATION
R0 – R17 as specified by "S" field of instruction.	Right-rotate as specified by "R" field of instruction.	Perform appropriate ALU operation. (Note 1)	Left-rotate and mask as specified by the "D" and "L" fields of instruction. (Note 4)	When R18 – R1F are specified in the "D" field, data is returned to the address specified in the "Source" field.

XMIT 8 (XT8), ADD IMMMEDIATE 8 (AD8), AND IMMEDIATE 8 (AN8), XOR IMMEDIATE 8 (XR8)

SOURCE	PRE-ALU	ALU	POST-ALU	DESTINATION
One to eight bit constants from data field of instruction and register/ address specified by either the source or destination field.	NOP.	Perform appropriate ALU operation. (Note 5)	Mask as specified by the "L" field.	**XT8, AD8:** Results are sent to register/address, starting from LSB, specified by the destination field of the instruction word. **AN8, XR8:** Results are sent to RF, starting from LSB.

XMIT 5 (XT5), ADD IMMMEDIATE 5 (AD5), AND IMMEDIATE 5 (AN5), XOR IMMEDIATE 5 (XR5)

SOURCE	PRE-ALU	ALU	POST-ALU	DESTINATION
One to five bit constants from data field of instruction and register/ address specified by either the source or destination.	Right-rotate as specified by "R" field, the data defined in either the source or destination field.	Perform appropriate ALU operation. (Note 6)	Left-rotate and mask as specified by the "R" and "L" fields.	**XT5, AD5:** Subfield is merged into register/ address specified by destination field. LSB of data field is sent to bit position of destination defined by "R" field. **AN5, XR5:** Subfield is merged into register/ address specified by source field. The "R" field specifies the LSB position of source and RF. Results are returned to RF.

ALL CONDITIONAL AND UNCONDITIONAL RETURNS

ADDRESS REGISTER (AR)	PROGRAM COUNTER (PC)	STACK
Conditional Returns: If condition is true. AR = address from top of stack, else load address from instruction word. **Unconditional Returns:** AR = address from top of stack.	PC = AR.	**Conditional Returns:** If condition is true. POP stack, else NOP. **Unconditional Returns:** POP stack.

ALL CONDITIONAL JUMPS, POP STACK AND JUMP (PSJ), and JUMP (JMP)

ADDRESS REGISTER	PROGRAM COUNTER	STACK
Conditional Jumps: If condition is true. AR = instruction word address, else AR = PC + 1. **PSJ and JMP:** AR = instruction word address.	PC = AR.	**Conditional Jumps:** NOP. **PSJ:** POP stack. **JMP:** NOP.

DATA OPERATIONS OF THE 8X401 (CONTINUED)

CALL (JSR), INTERRUPT ($\overline{INT}$), NON-MASKABLE INTERRUPT ($\overline{NMI}$)

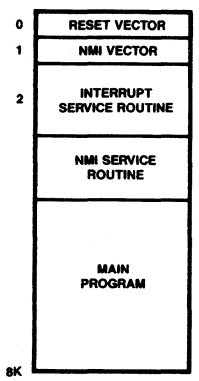
ADDRESS REGISTER	STACK	PROGRAM COUNTER
JSR: AR = instruction address. **$\overline{INT}$:** AR = address 2. **$\overline{NMI}$:** AR = address 1.	PUSH PC + 1. (Note 6)	PC = AR.

XEC

ADDRESS REGISTER	PROGRAM COUNTER	STACK
Right-most L bits of RB merged into corresponding bits in instruction address	PC not updated.	NOP.

NOTES:

1. **ALU Descriptions:**
   - MOVE: • No operation
   - ADD: • Source data ADDed to contents of auxiliary register (R0 – AUX). Carry bit set if carry is generated at MSB of selected data field. NZ status bit set if specified bits are not zero after ALU add.
   - AND: • Source data ANDed to contents of AUX register. NZ status bit updated accordingly.
   - XOR: • Source data Exclusive-ORed with contents of AUX register. NZ status bit set accordingly.
   - ADD with CARRY: • Sum is formed from source data. AUX register, and carry bit (register 10, bit 0). Carry and NZ status bits are set when appropriate.
2. Left-rotate is suppressed when destination is R0 (AUX).
3. When address registers RC, RD and RE are specified in the destination, source data will also go out on banks A, B, C, respectively. The L-field should be zero (a full 8-bit operation) to ensure duplication of the two outputs.
4. A left-rotate of 0 – 7 bits will correspond to R18 – R1F as specified in the "Destination" field of instruction word.
5. **ALU Descriptions:**
   - XMIT: • Input constants from the instruction word to specified destination. NZ flag is not updated when an XMIT is performed: however, NZ can be written to by an XMIT if R17 bit 1 is within the destination field.
   - ADD IMMEDIATE: • Instruction word data is ADDed to data specified by destination fied. The carry bit is set if a carry is generated at the MSB of the selected data field. NZ status bit is updated to reflect the value of "L" bits of data after the addition.
   - AND IMMEDIATE: • Instruction word data is ANDed to data specified by source field. Returning the destination data to RF allows the operation to be performed without destroying the original data field. This will facilitate testing of data for certain pre-defined values while still preserving the original data for other uses. NZ status bit updated accordingly. Unspecified bits in RF remain unchanged.
   - XOR IMMEDIATE: • Same as AND IMMEDIATE, except the logical operation performed is Exclusive-OR.
6. Note that the stack operation is show before the PC in the CALL and INTERRUPT formats. This is because the stack is actually in operation in cycle 3, and the PC is updated in cycle 4 (see Figure 1). In fact, for the Call (JSR) instruction and interrupt servicing, cycle order is important for the user to understand the current status of the PC. The other instructions are in reverse order for visual simplicity in keeping with block diagram flow, and cycle order is irrelevant.



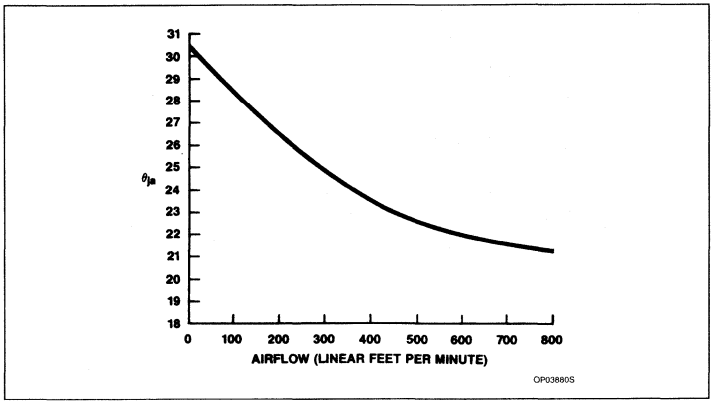
- **The RESET VECTOR, LOCATED AT ADDRESS 0, WILL BE AN UNCONDITIONAL JUMP TO THE BEGINNING OF THE MAIN PROGRAM.**
- **THE NON-MASKABLE INTERRUPT ($\overline{NMI}$) VECTOR, LOCATED AT ADDRESS 1, IS A JUMP TO THE NMI SERVICE ROUTINE.**
- **LOCATED AT ADDRESS 2 IS THE ROUTINE THAT SERVICES INTERRUPT ($\overline{INT}$) CALLS.**

AF03650S

Figure 2. Typical 8X401 System Memory Map

Thermal Junction Temperature vs Airflow

The ceramic package used for the 8X401 has no heat sink and a θ_{ja} rating of 30.5°C/W in still air. Currently, to ensure operation at 150ns, the junction temperature of the devices must be kept below 115°C. The maximum power dissipation at that junction temperature will be 2.4W so that airflow will be required for full commercial range operation. The θ_{ja} versus Airflow curve is drawn here:



DC ELECTRICAL CHARACTERISTICS Commercial Part $4.75V \leqslant V_{CC} \leqslant 5.25V$, $0°C \leqslant T_A \leqslant 70°C$[1]

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS			UNIT	COMMENTS
			Min	Typ	Max		
V_{CC}	Supply voltage		4.75	5.0	5.25	V	
V_{IH}	High level input voltage		2.0			V	
V_{IL}	Low level input voltage				0.8	V	
V_{OH}	High level output voltage	V_{CC} = Min, I_{OH} = −3mA	2.4			V	$\overline{DA0}$ through $\overline{DA7}$, MCLK, SC, WC, AB, BB, CB
		V_{CC} = Min, I_{OH} = −400μA	2.4			V	All others
V_{OL}	Low level output voltage	V_{CC} = Min, I_{OL} = 16mA			0.5	V	$\overline{DA0}$ through $\overline{DA7}$, MCLK, RWC, SC, WC, $\overline{A}$, $\overline{B}$, $\overline{C}$
		V_{CC} = Min, I_{OL} = 8mA			0.5	V	A0 through A12, PS, NZ, CY, IR
V_{IC}	Input clamp voltage	V_{CC} = Min, I_{IN} = −10mA			−1.5	V	
I_{IH}	High level input current	V_{CC} = Max, V_I = 2.7			20	μA	
I_{IL}	Low level input current	V_{CC} = Max, V_I = 0.4V			−400	μA	
I_{OZH}	Off-state output current, high level voltage applied	V_{CC} = Max, V_O = 2.7V			50	μA	$\overline{DA0}$ through $\overline{DA7}$
I_{OZL}	Off-state output current low level voltage applied	V_{CC} = Max, V_O = 0.4V			−400	μA	$\overline{DA0}$ through $\overline{DA7}$
I_{OS}	Short circuit output current[2]	V_{CC} = Max, V_O = 0V	−30		−140	mA	
I_{CC}	Supply Current	V_{CC} = Max			500	mA	T_A = 0°C; Cold start[3]
					430	mA	T_J = 115°C

NOTES:
1. 64 Pin CDIP, Airflow required for full Commercial operation. (A plastic 64 Pin DIP with internal heat sink is currently under development which will not have this requirement.) See above for thermal characteristics.
2. Not more than one output should be tested at a time.
3. Guaranteed by Correlation to I_{CC} measured at 25°C.

AC ELECTRICAL CHARACTERISTICS (V_{CC} = 5V +5%, o°C ≤ T_A ≤ 70°C)[1]

SYMBOL	PARAMETER	150ns CYCLE			> 150ns CYCLE			UNIT	COMMENTS
		Min	Typ	Max	Min	Typ	Max		
t_{PC}	Processor cycle time	150						ns	
t_{CP}	Clock pulse period	37.5						ns	
t_{CH}	Clock pulse high time	15						ns	See Note 2
t_{CL}	Clock pulse low time	15						ns	
t_{MCL}	CP1 to MCLK low			30				ns	See Note 2
t_{MCH}	CP4 to MCLK high			40				ns	
t_W	MCLK pulse width	25	31	38	T4Q – 12		T4Q	ns	
t_{RWCL}	CP1 to RWC low		34	40				ns	
t_{RWCH}	CP3 to RWC high			40				ns	
t_{RWCW}	RWC pulse width	65	75	80	T3Q + T4Q – 10		T3Q + T4Q + 5	ns	
t_{AS}	CP2 to address stable			52				ns	
t_{MAS}	MCLK to address stable			62			T1Q + 24	ns	
t_{IS}	Instruction setup to CP1	0						ns	See Note 3
t_{MIS}	Instruction setup to MCLK	25						ns	
t_{IH}	Instruction hold from CP2	20						ns	
t_{MIH}	Instruction hold from MLCK	25			T1Q – 12			ns	
t_{SCH}	CP3 to SC rising edge			45				ns	
t_{MSCH}	MCLK to SC rising edge			95			T1Q + T2Q + 20	ns	
t_{WCH}	CP3 to WC rising edge			55				ns	
t_{MWCH}	MCLK to WC rising edge			105			T1Q + T2Q + 20	ns	
t_{WL}	CP1 to SC/WC falling edge			35				ns	See Note 4
t_{MWL}	MCLK to SC/WC falling edge	0						ns	
t_{IBSL}	CP1 to input phase bank signal falling edge			60				ns	
t_{MIBSL}	MCLK to input phase bank signal falling edge			33				ns	
t_{IBSH}	CP3 to input phase bank signal rising edge			45				ns	
t_{MIBSH}	MCLK to input phase bank signal rising edge			95			T1Q + T2Q + 20	ns	
t_{OBSL}	CP3 to output phase bank signal falling edge			53				ns	
t_{MOBSL}	MCLK to output phase bank signal falling edge			105			T1Q + T2Q + 30	ns	
t_{OBSH}	CP1 to output phase bank signal rising edge			46				ns	
t_{MOBSH}	MCLK to output phase bank signal rising edge	0		20				ns	
t_{IDS}	Input data setup to CP3	–3						ns	
t_{MIDS}	Input data setup to MCLK	–50			25 – T1Q – T2Q			ns	
t_{IDH}	Input data hold from CP3	28						ns	
t_{MIDH}	Input data hold from MCLK	78			T1Q + T2Q + 3			ns	
t_{ODH}	Output data hold from CP1	35		55				ns	
t_{MODH}	Output data hold from MCLK	10		25				ns	

AC ELECTRICAL CHARACTERISTICS (CONTINUED) (V_{CC} = 5V ±5%, 0°C ≤ T_A ≤ 70°C)[1]

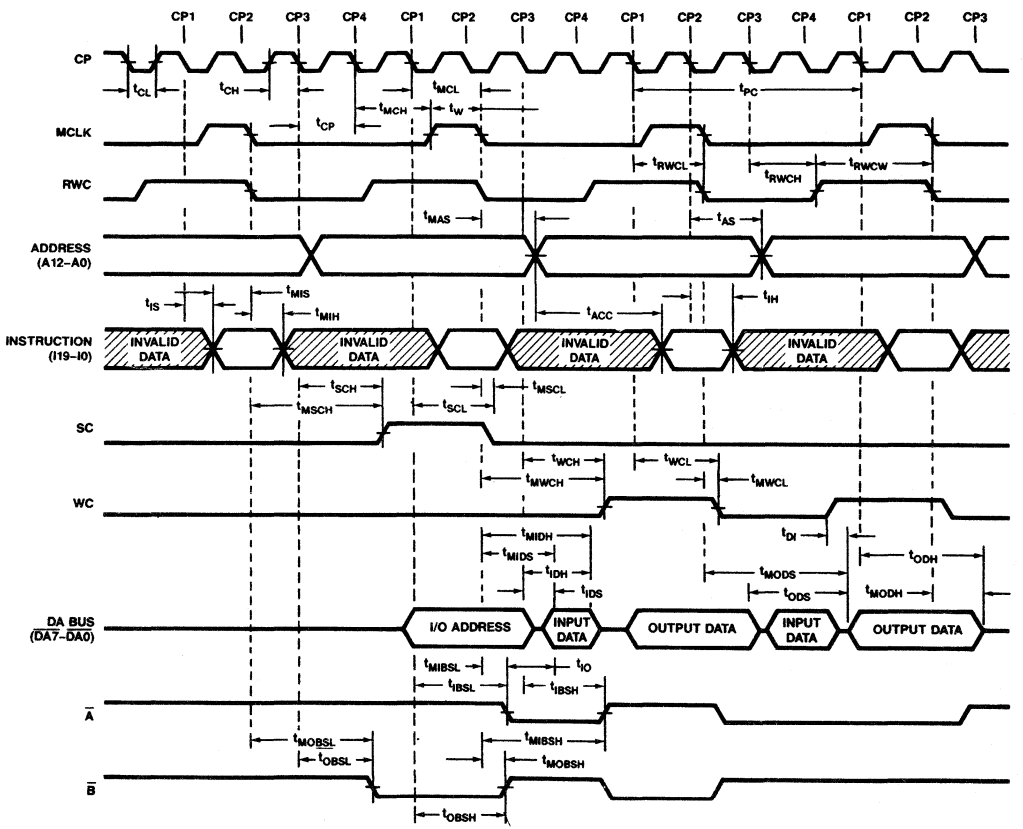
SYMBOL	PARAMETER	150ns CYCLE			> 150ns CYCLE			UNIT	COMMENTS
		Min	Typ	Max	Min	Typ	Max		
t_{ODS}	CP3 to output data stable			70				ns	
t_{MODS}	MCLK to output data stable			120			T1Q + T2Q + 45	ns	
t_{DI}	SC/WC rising edge to output driver turn on	18						ns	
t_{HS}	Halt setup to CP2	0						ns	
t_{MHS}	Halt setup to MCLK	−10						ns	
t_{HH}	Halt hold from CP2	50						ns	
t_{MHH}	Halt hold from MCLK	60			T1Q + 22			ns	
t_{SIS}	Status input setup to CP1	10						ns	
t_{MSIS}	Status input setup to MCLK	40						ns	
t_{SIH}	Status input hold from CP1	20						ns	
t_{MSIH}	Status input hold from MCLK	0						ns	
t_{SCRS}	SCR setup to CP1	0						ns	
t_{MSCRS}	SCR setup to MCLK	25						ns	
t_{SCRH}	SCR hold from CP2 (Slow CP2)	20						ns	See Diagram for CP2
t_{MSCRH}	SCR hold from MCLK	63			ST1Q − 12			ns	Slow T1Q
t_{INTS}	INT setup to CP1	10						ns	
t_{MINTS}	INT setup to MCLK	40						ns	
t_{INTH}	INT hold from CP1	10						ns	
t_{MINTH}	INT hold from MCLK	−10						ns	
t_{CYU}	CP4 to CY update			60				ns	
t_{MCYU}	MCLK to CY update			−10			28 − T4Q	ns	
t_{NZU}	CP4 to NZ update			60				ns	
t_{MNZU}	MCLK to NZ update			−5			33 − T4Q	ns	
t_{IRU}	CP4 to IR update			75				ns	
t_{MIRU}	MCLK to IR update			20			58 − T4Q	ns	
t_{PSU}	CP4 to PS update			60				ns	
t_{MPSU}	MCLK to PS update			−10			28 − T4Q	ns	
t_{ACC}	Program memory access time (address stable to valid instruction)			60		T2Q + T3Q + T4Q − 52			ns
t_{IO}	I/O port output enable time (bank signal to valid data on bus)			24			T1Q + T2Q − 51	ns	
t_{RW}	Reset pulse width	150			t_{PC}			ns	
t_{NMIW}	NMI pulse width	50						ns	
t_{NMIS}	NMI setup to CP2	15						ns	See Note 5
t_{MNMIS}	NMI setup to MCLK	10						ns	See Note 5

NOTES:

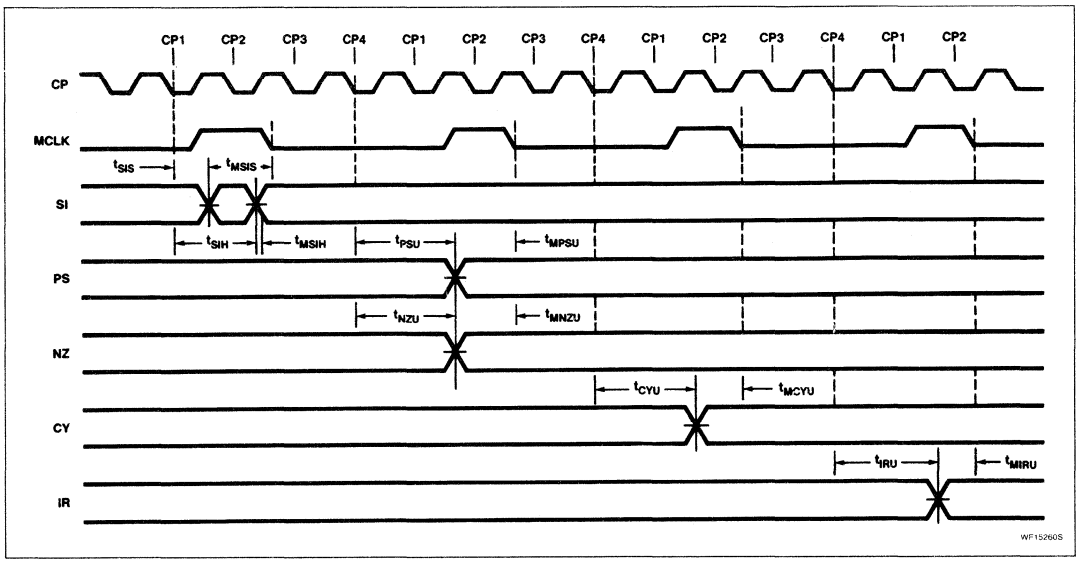
1. Inputs swing between 0V and 3V. All outputs are measured at 1.5V with loading as specified in the test circuits.
2. CP1, CP2, CP3, and CP4 refer to the clock pulse that causes the first, second, third, and fourth 8X401 quarter cycles, respectively. Parameters referenced to MCLK, CP1, CP2, CP3, and CP4 are measured to the falling edge of those signals. T1Q, T2Q, T3Q, and T4Q represent time intervals for the first, second, third, and fourth 8X401 quarter cycles, respectively. Duty cycle can be from 40% to 60%.
3. Instructions must be setup before CP1.
4. t_{WL} represents t_{SCL} and t_{WCL}. t_{MWL} represents t_{MSCL} and t_{MWCL}.
5. This guarantees NMI is serviced in the current cycle.

TEST CIRCUIT

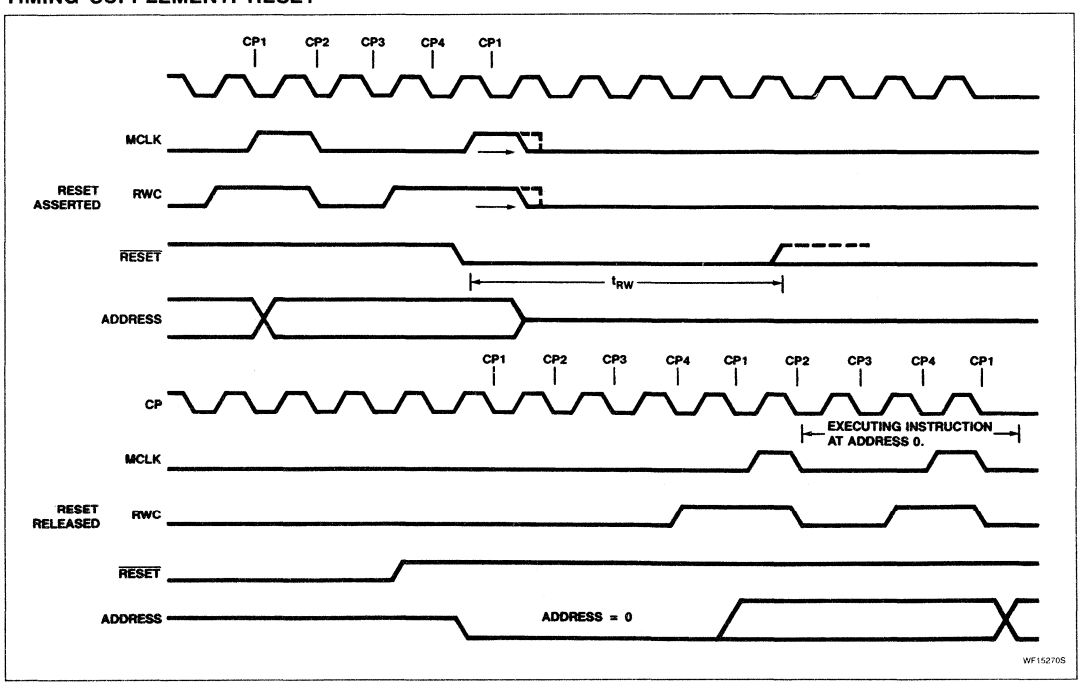
MAIN TIMING DIAGRAM



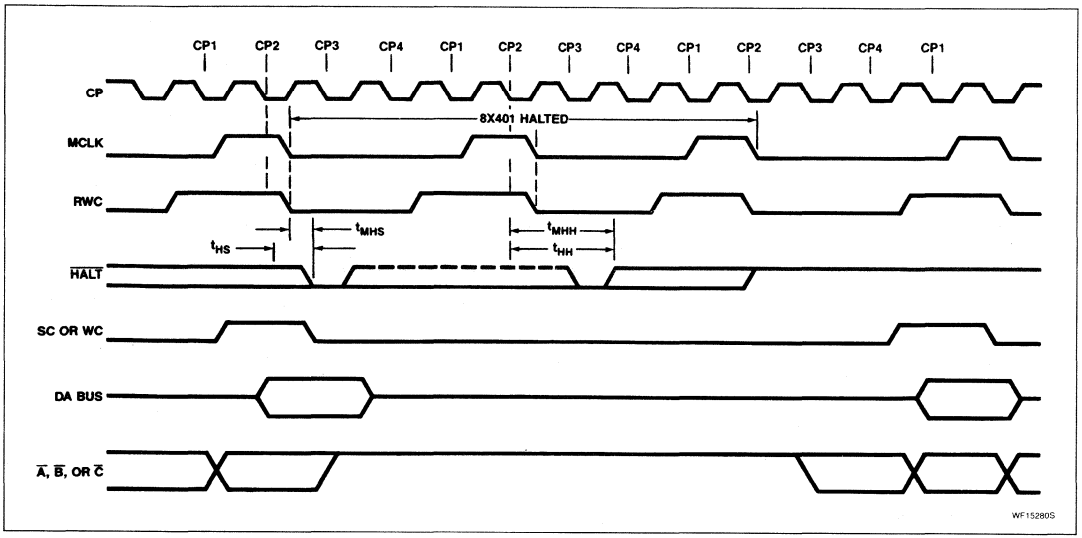
TIMING SUPPLEMENT: STATUS



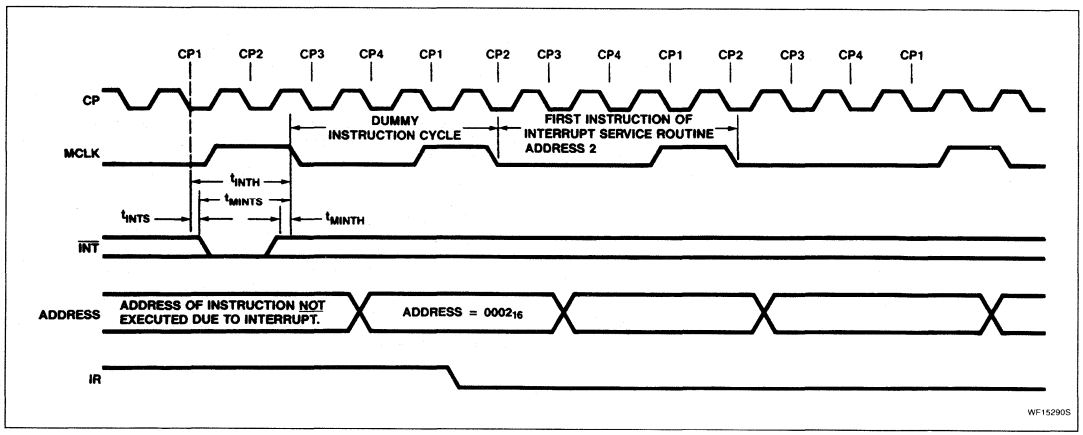
TIMING SUPPLEMENT: $\overline{\text{RESET}}$



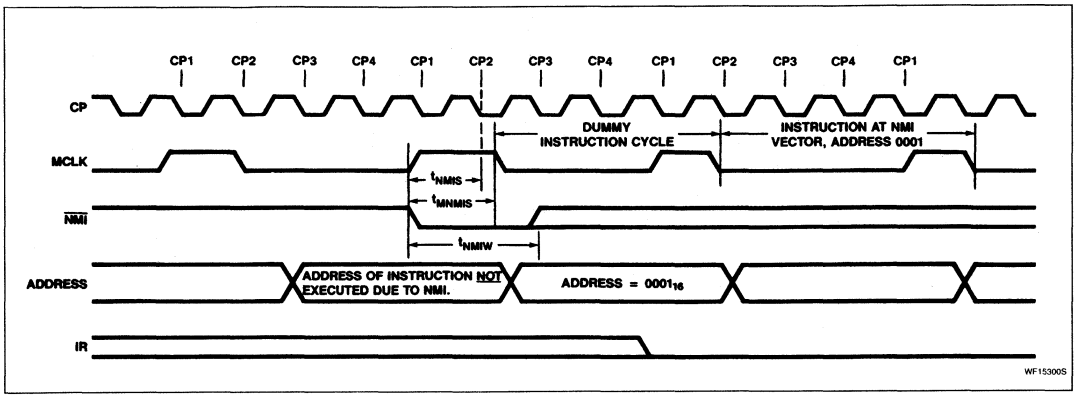
TIMING SUPPLEMENT: $\overline{\text{HALT}}$



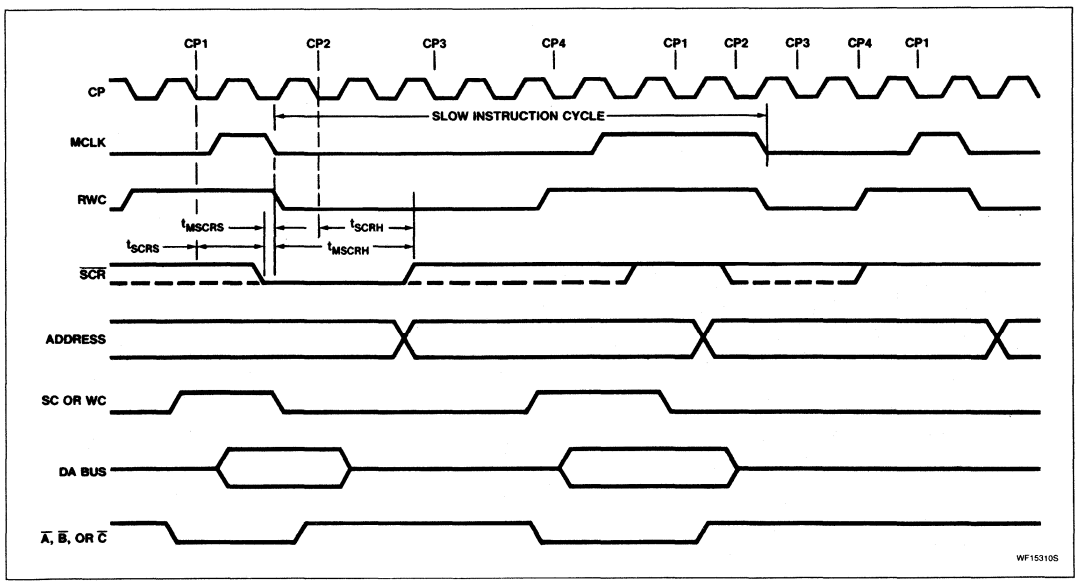
TIMING SUPPLEMENT: $\overline{\text{INT}}$



TIMING SUPPLEMENT: $\overline{NMI}$



TIMING SUPPLEMENT: $\overline{SCR}$



Signetics

8X450
RAM

Product Specification

Microprocessor Products

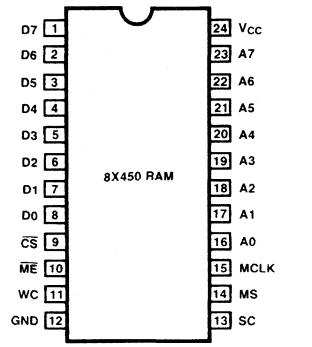
DESCRIPTION

The 8X450 is a 256-byte RAM designed primarily as a working storage element for 8X305 based systems. The 8X450 can be used as a dedicated RAM or in groups of RAMs to form a customized memory stack.

FEATURES

- **256-byte RAM**
- **Two modes of operation**
- **Paged RAM capability**
- **On-chip address latches**
- **Directly bus compatible with the 8X305 microcontroller**
- **24-pin slim line package**
- **Single + 5-volt supply**

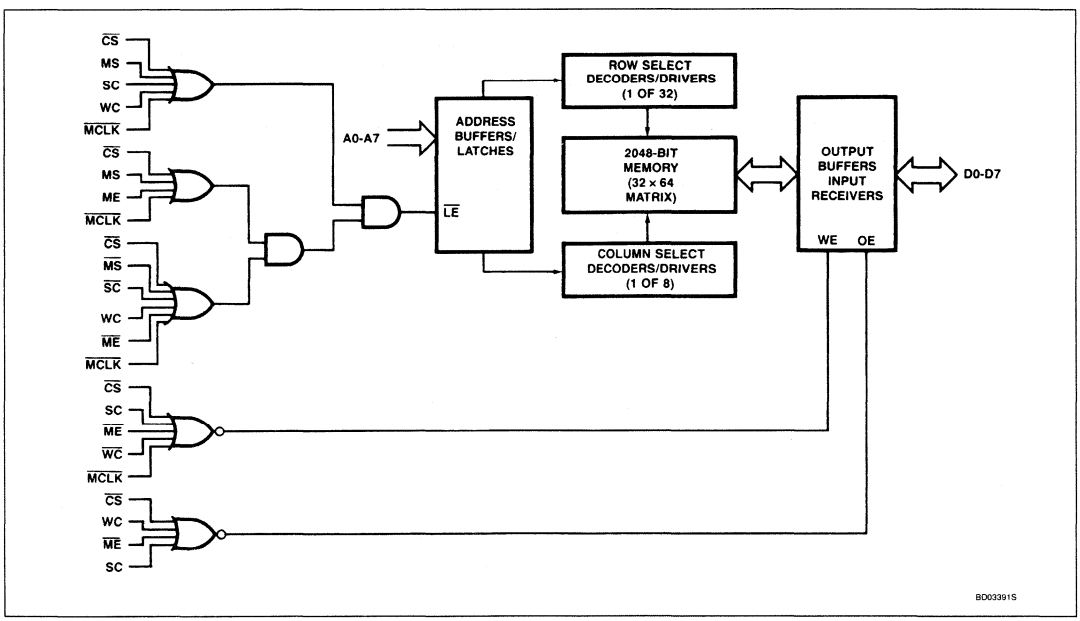
PIN CONFIGURATION



ORDERING INFORMATION

DESCRIPTION	ORDER CODE
24-Pin Ceramic DIP 400 Mil Wide	N8X450I

8X450 BLOCK DIAGRAM



FUNCTIONAL OPERATION

The 8X450 can be used in either of two operating modes — conventional DA select or fast select. Each mode supports either single or multiple RAM applications. The mode of operation is selected by the MS pin.

Conventional $\overline{DA}$ Select (MS = 1)

When MS is tied high, the conventional $\overline{DA}$ addressing mode is selected; in this mode, the 8X450 data and address lines are tied together. For applications that use a single RAM, the chip select ($\overline{CS}$) input is grounded. To prevent bus contention in applications that use multiple RAMs, the $\overline{CS}$ input is driven by an external source. Examples of both configurations (single and multiple RAMs) are shown in the Application Diagrams at the rear of this data sheet. Control signals and operating conditions for conventional $\overline{DA}$ addressing are summarized in Table 1.

Fast Select Operations (MS = 0)

When MS is tied low, the fast select addressing mode is invoked; in this mode, the 8X450 address lines are driven by an external source and the 8X450 data lines are tied to the 8X305 $\overline{IV}$ bus. The $\overline{CS}$ input is treated exactly the same as in conventional $\overline{DA}$ addressing; examples of single and multiple RAM configurations are shown at the rear of this data sheet. Control signals and operating conditions for fast select addressing are summarized in Table 2.

PIN CONFIGURATION AND DECRIPTIONS

PIN NO.	IDENTIFIER	FUNCTION
1 – 8	D7 – D0	Bidirectional data bus, D0 is LSB
9	$\overline{CS}$	Chip Select, active low
10	$\overline{ME}$	Master Enable, active low
11	WC	Write Control, active high
12	GND	Ground
13	SC	Select Control, active high
14	MS	Mode Select
15	MCLK	Master Clock, active high
16 – 23	A0 – A7	Address lines, A0 is LSB
24	V_{CC}	Power supply

Table 1. Conventional $\overline{DA}$ Select (MS = 1)

SC	WC	$\overline{ME}$	MCLK	FUNCTION
0	0	0	0	Read from RAM
0	0	1	X	None
0	1	0	1	Write data to RAM
0	1	1	1	None
1	0	0	1	Conventional $\overline{DA}$ select (write address to RAM)
1	0	1	1	None

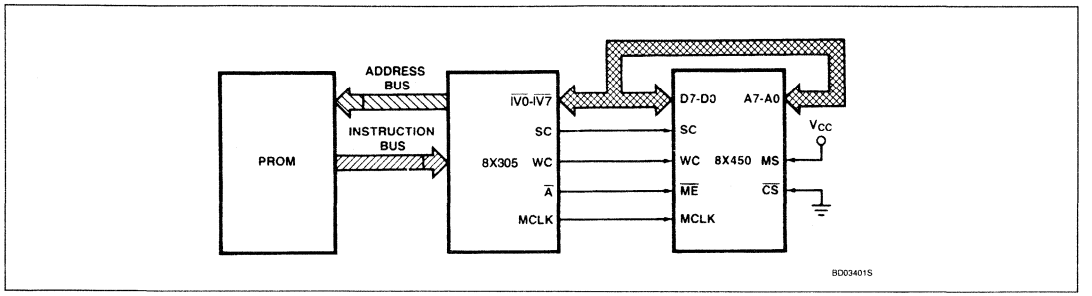
NOTES:
1. Other states are not generated by the 8X305 and are not defined for operation in this mode.
2. Table 1 is valid only if the chip select ($\overline{CS}$) is low; if $\overline{CS}$ is high, the 8X450 is disabled.

Table 2. Fast Select (MS = 0)

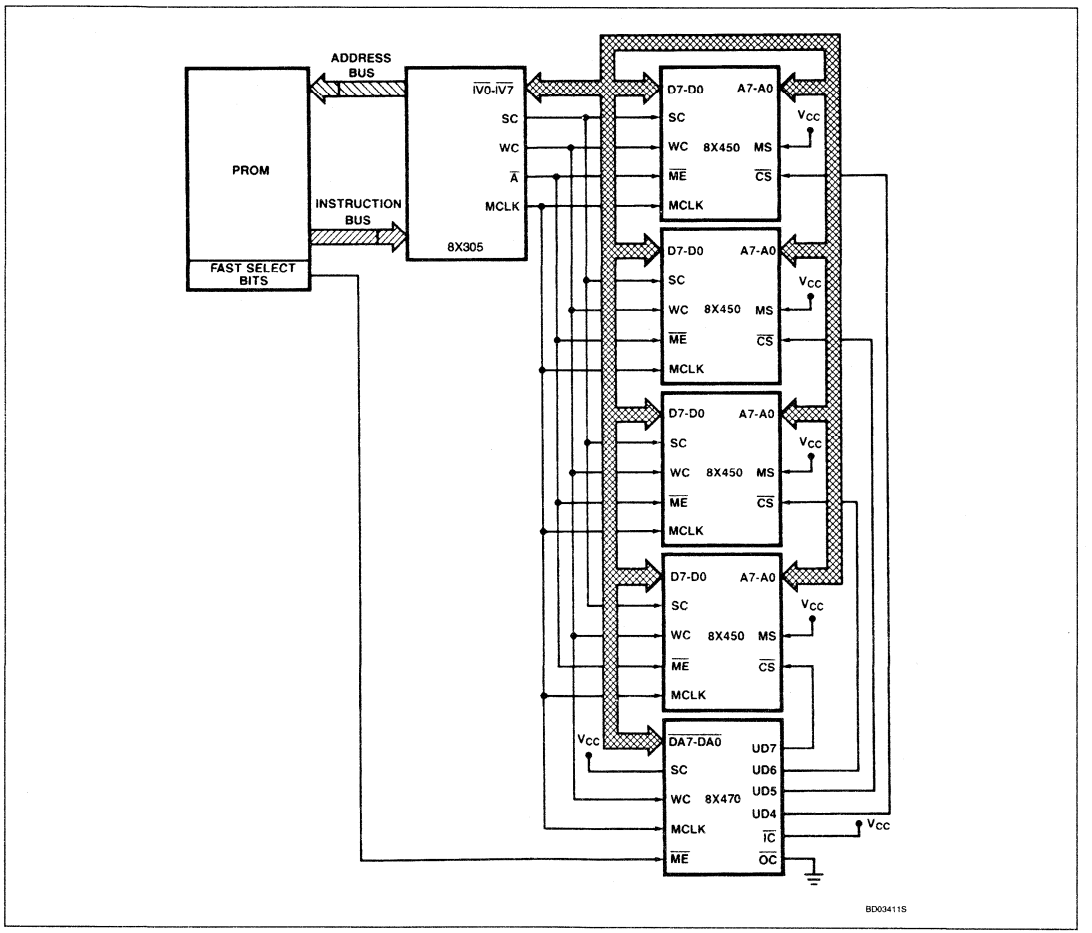
SC	WC	$\overline{ME}$	MCLK	FUNCTION
0	0	0	0	Read from RAM
0	0	1	0	None
X	X	1	1	Write address to RAM
0	1	0	1	Write data to RAM

NOTES:
1. Other states are not generated by the 8X305 and are not defined for operation in this mode.
2. Table 2 is valid only if the chip select ($\overline{CS}$) is low. If $\overline{CS}$ is high, the 8X450 is disabled.

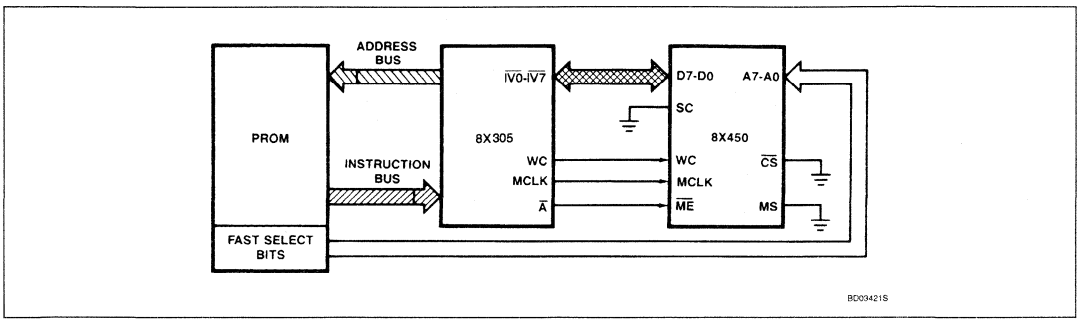
SINGLE RAM CONFIGURATION (CONVENTIONAL $\overline{DA}$ SELECT)



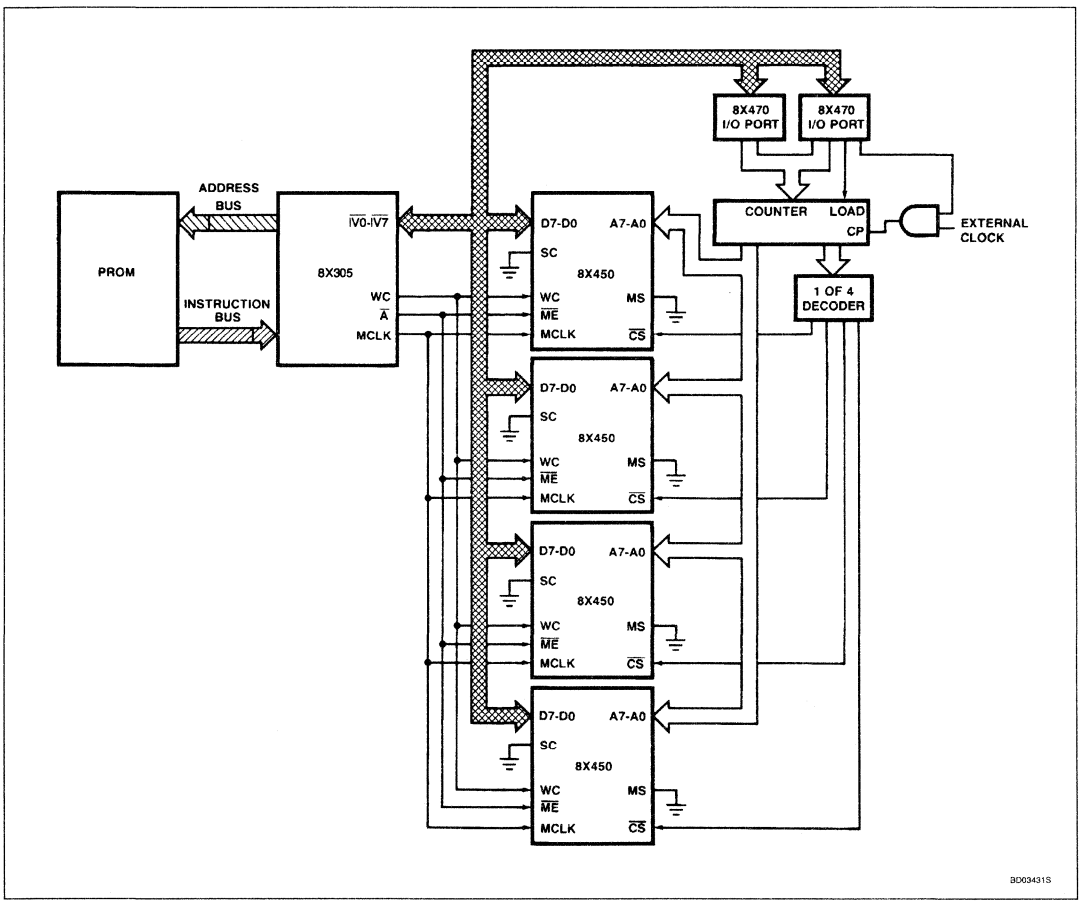
MULTIPLE RAM CONFIGURATION (CONVENTIONAL $\overline{DA}$ SELECT)



SINGLE RAM CONFIGURATION (FAST SELECT)



MULTIPLE RAM CONFIGURATION (FAST SELECT)



ABSOLUTE MAXIMUM RATINGS

SYMBOL	PARAMETER	RATING	UNIT
V_{CC}	Power supply voltage	+7	V_{DC}
V_{IN}	Input voltage	+5.5	V_{DC}
V_O	Output off-state voltage	+5.5	V_{DC}
T_{STG}	Storage temperature range	−65 to +150	°C

DC ELECTRICAL CHARACTERISTICS (V_{CC} = 5V +5%, 0°C ≤ T_A ≤ 70°C)

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS			UNIT
			Min	Typ	Max	
V_{CC}	Supply voltage		4.75	5.00	5.25	V
V_{IL}	Low level input voltage				0.8	V
V_{IH}	High level input voltage		2.0			V
V_{OL}	Low level output voltage	V_{CC} = Min, I_{OL} = 16mA			0.5	V
V_{OH}	High level output voltage	V_{CC} = Min, I_{OH} = −3mA	2.7			V
V_{IC}	Input clamp voltage	V_{CC} = Min, I_{IN} = −10mA			−1.5	V
I_{IL}	Low level input current	V_{CC} = Max, V_I = 0.4V			−0.4	mA
I_{IH}	High level input current	V_{CC} = Max, V_I = 2.7V			20	μA
I_{OZL}	High-Z output current (Low Level)	V_{CC} = Max, V_O = 0.4V			−0.4	mA
I_{OZH}	High-Z output current (High Level)	V_{CC} = Max, V_O = 2.7V			50	μA
I_{OS}	Short circuit output current	V_{CC} = Max + 0.5, V_O = 0.5V	−30		−140	mA
I_{CC}	Supply current	V_{CC} = Max			220	mA

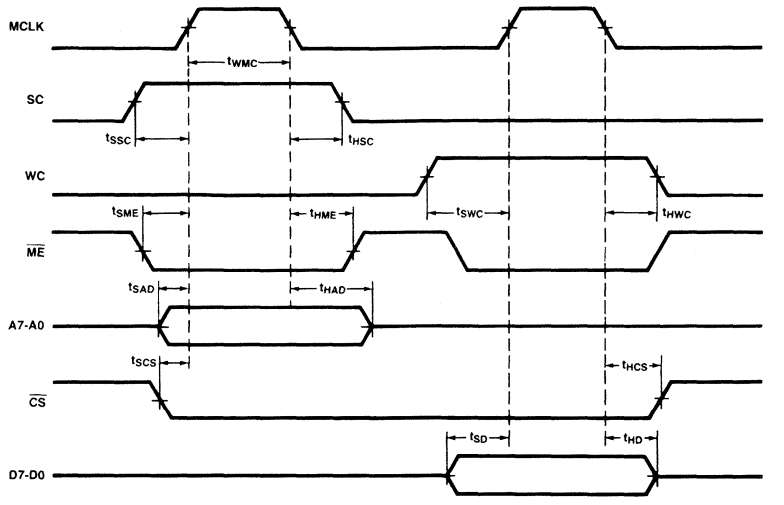
NOTE:
*Not more than one output should be tested at a time.

AC ELECTRICAL CHARACTERISTICS (V_{CC} = 5V +5%, 0°C ≤ T_A ≤ 70°C)

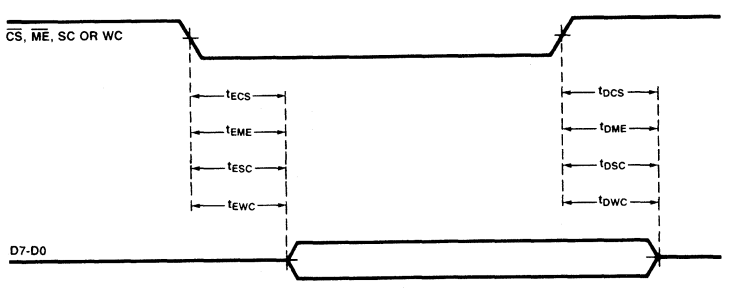
SYMBOL	PARAMETER	REFERENCES		TEST CONDITIONS	LIMITS		UNIT
		From	To		Min	Max	
Pulse Widths							
t_{WMC}	MCLK pulse width	↑ MCLK	↓ MCLK		35		ns
Setup Times							
t_{SCS}	Setup time for $\overline{CS}$	↓ $\overline{CS}$	↑ MCLK		0		ns
t_{SSC}	Setup time for SC	↑ SC	↑ MCLK		15		ns
t_{SWC}	Setup time for WC	↑ WC	↑ MCLK		15		ns
t_{SME}	Setup time for $\overline{ME}$	↑ $\overline{ME}$	↑ MCLK		0		ns
t_{SAD}	Setup time for address	A7 – A0	↑ MCLK		0		ns
t_{SD}	Setup time for data	$\overline{DA}$ Bus Active	↑ MCLK		0		ns
Hold Times							
t_{HCS}[3]	Hold time for $\overline{CS}$	↓ MCLK	↑ $\overline{CS}$		0		ns
t_{HSC}	Hold time for SC	↓ MCLK	↓ SC		0		ns
t_{HWC}	Hold time for WC	↓ MCLK	↓ WC		0		ns
t_{HME}	Hold time for $\overline{ME}$	↓ MCLK	↑ $\overline{ME}$		0		ns
t_{HAD}	Hold time for address	↓ MCLK	A7 – A0		10		ns
t_{HD}	Hold time for data	↓ MCLK	$\overline{DA}$ Bus Tri-Stated		5		ns
Enable Times							
t_{ECS}	Enable time for $\overline{CS}$	↓ $\overline{CS}$	$\overline{DA}$ Bus Active	$\overline{ME}$, SC, WC, MCLK = Low		30	ns
t_{EME}	Enable time for $\overline{ME}$	↓ $\overline{ME}$	$\overline{DA}$ Bus Active	$\overline{CS}$, SC, WC, MCLK = Low		30	ns
t_{ESC}	Enable time for SC	↓ SC	$\overline{DA}$ Bus Active	$\overline{CS}$, $\overline{ME}$, WC, MCLK = Low		30	ns
t_{EWC}	Enable time for WC	↓ WC	$\overline{DA}$ Bus Active	$\overline{CS}$, $\overline{ME}$, SC, MCLK = Low		30	ns
Disable Times							
t_{DCS}	Disable time for $\overline{CS}$	↑ $\overline{CS}$	$\overline{DA}$ Bus Tri-Stated	$\overline{ME}$, SC, WC, MCLK = Low		40	ns
t_{DME}	Disable time for $\overline{ME}$	↑ $\overline{ME}$	$\overline{DA}$ Bus Tri-Stated	$\overline{CS}$, SC, WC, MCLK = Low		40	ns
t_{DSC}	Disable time for SC	↑ SC	$\overline{DA}$ Bus Tri-Stated	$\overline{CS}$, $\overline{ME}$, WC, MCLK = Low		40	ns
t_{DWC}	Disable time for WC	↑ WC	$\overline{DA}$ Bus Tri-Stated	$\overline{CS}$, $\overline{ME}$, SC, MCLK = Low		40	ns

NOTES:
1. All inputs swing between 0V and 3V.
2. All outputs are measured at 1.5V with loading specified in TEST CIRCUIT.
3. $\overline{CS}$ must be active for at least one complete 8X305 instruction cycle.

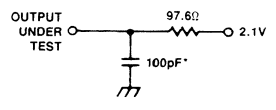
8X450 TIMING



ENABLE AND DISABLE TIMING DIAGRAM



TEST CIRCUIT



NOTE:
*Includes test fixture capacitance.

Signetics

8X470
I/O Port

Product Specification

Microprocessor Products

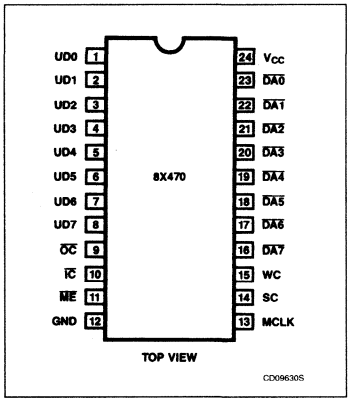
DESCRIPTION

The 8X470 I/O Port (Figure 1) is a fuse-programmable device designed to interface a user system with a Signetics 8X401 Microcontroller. The 8X470 has three internal registers — a data register and two status registers. The data register is shared between the two systems, while each system has its own dedicated status register. The 8X470 can be programmed to provide a bidirectional or unidirectional user interface with the user data entry being either synchronous or asynchronous with the 8X401.

FEATURES

- **Two independent 8-bit busses**
- **Fast polling**
- **Directly bus-compatible with the 8X401, 8X305 Microcontrollers**
- **256 Data Addresses**
- **Separate Controller and User status registers**
- **16 Status Addresses**
- **Positionable Controller Status Register output**
- **Bidirectional or unidirectional I/O capability**
- **Synchronous or asynchronous user data entry**
- **Conventional $\overline{DA}$ or Fast select addressing techniques**
- **24-pin slimline package**
- **Single +5V supply**

PIN CONFIGURATION



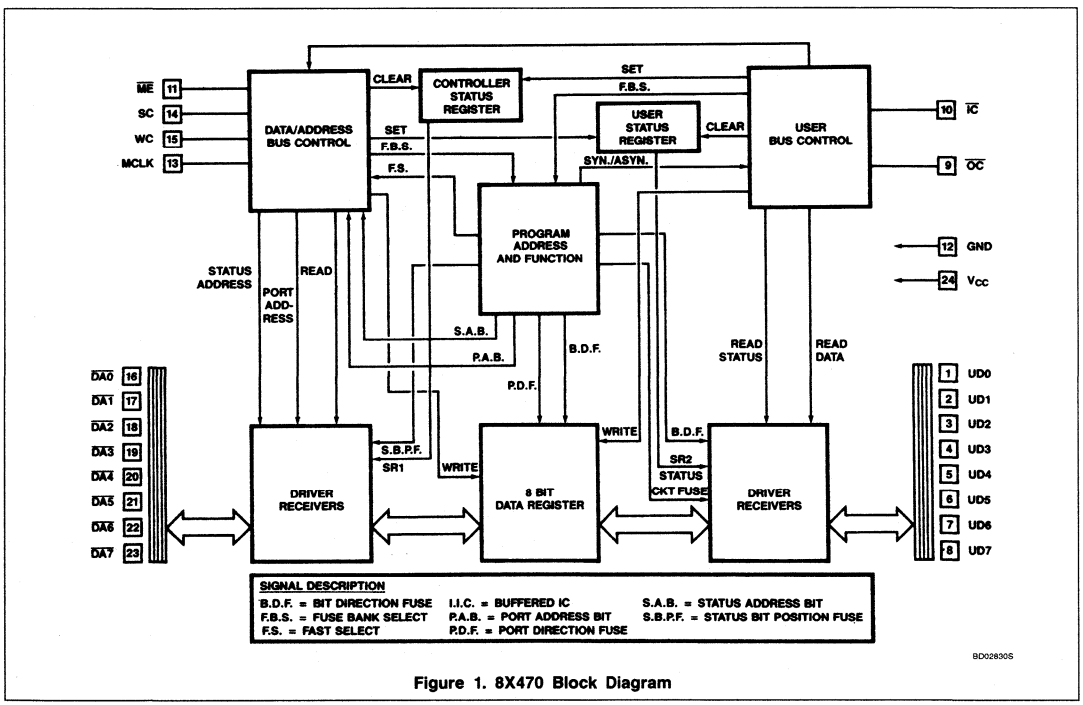


Figure 1. 8X470 Block Diagram

PIN DESCRIPTION

PIN	IDENTIFIER	FUNCTION
1 – 8	UD0 – UD7	User port data lines. Active high.
9	$\overline{OC}$	**Output Control:** An active low signal which controls reading into the I/O port from the UD bus.
10	$\overline{IC}$	**Input Control:** An active low signal which controls writing into the I/O port from the UD bus.
11	$\overline{ME}$	**Master Enable:** The $\overline{ME}$ pin depends upon the addressing mode for its function. Conventional $\overline{DA}$ select: $\overline{ME}$ is an active low signal which allows read or write or address operations from the $\overline{DA}$ bus. Fast select: $\overline{ME}$ is an active high signal which allows the controller status register to be read on the $\overline{DA}$ bus.
12	GND	**Ground.**
13	MCLK	**Master Clock:** Input from the 8X401.
14	SC	**Select Command:** The SC pin depends upon the addressing mode for its function. Conventional $\overline{DA}$ select: An active high signal which compares the 8X401 $\overline{DA}$ bus and the address. An address match selects the 8X470. Fast select: SC is an active high signal which outputs the contents of the data register on the $\overline{DA}$ bus while clearing the control status register.
15	WC	**Write Command:** An active high signal which controls the writing of data into the I/O port from the 8X401 $\overline{DA}$ bus.
16 – 23	$\overline{DA7}$ – $\overline{DA0}$	Controller Data/Address ($\overline{DA}$) lines. Active low.
24	V_{CC}	+5V power supply.

In a typical system configuration (Figure 2), the 8X470 connects to the 8X401 Microcontroller via the Data/Address ($\overline{DA}$) bus; the 8X470 is linked to the user system via the User Data (UD) bus. To facilitate interbus communications and to implement handshaking operations, a dedicated status register is assigned to each bus. The status registers can be polled at full speed to optimize throughput and to prevent errors in data transmissions. The 8X470 can be addressed from the $\overline{DA}$ bus in either of two ways — by the conventional microcontroller software-select cycle or by fast select methods.

$\overline{DA}$ BUS INTERFACE

The 8X470 interfaces to the 8X401 Microcontroller through an 8-bit inverted three-state Data/Address ($\overline{DA}$) bus and four control signals (SC, WC, $\overline{ME}$ and MCLK). The status registers are accessible through the $\overline{DA}$ bus interface. For conventional select operation, both the data and controller status register addresses can be fuse programmed. Also, the controller status can be programmed to appear at any or all $\overline{DA}$ bit locations. For time critical operations, the 8X470 can be fuse programmed for fast select operations.

UD BUS INTERFACE

The 8X470 interfaces to a user system through an 8-bit UD bus and two control signals ($\overline{IC}$ and $\overline{OC}$). If the user and the 8X401 attempt to input data to the I/O port at the same time, the user will have priority. Both the data register and the user status register are accessible through the UD bus interface. The 8X470 UD interface can be fuse programmed for synchronous or asynchronous data entry.

Data Entry

Data entry into the 8X470 from the user bus can be fuse programmed to be either synchronous or asynchronous. Synchronous data input to the data latches occurs when MCLK is high. Asynchronous data input is independent of MCLK. For further details, see the $\overline{IC}$ fuse bank section.

MODE A

The user data lines are bidirectional or input only. See the $\overline{IC}$ and SC fuse bank sections for further details.

MODE B

The user data lines are output or input only. See the $\overline{IC}$ and SC fuse bank sections for further details.

STATUS REGISTERS

The 8X470 has two status registers; the User and Controller status registers. These registers can be used for handshaking between

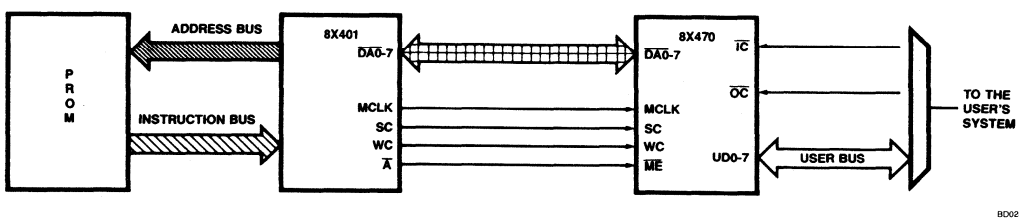


Figure 2. Typical System Configuration

the 8X401 and the User Data Bus. A register is set by a write from one bus and cleared by reading the data register from the opposite bus.

User Status Register

The User Status register is set by any write from the $\overline{DA}$ bus and cleared by reading the data register from the UD bus. The user status register is available on UD7 only. The output of the user status is controlled by $\overline{IC}$ and $\overline{OC}$ user port control lines. See the $\overline{IC}$ bank for further details.

Controller Status Register

The Controller Status register is set by any write from the UD bus and cleared by reading the data register from the $\overline{DA}$ bus. The Controller Status register can be accessed by either a fast select or a conventional $\overline{DA}$ select. For conventional select methods, the controller status register address can be fuse programmed to be any one of the addresses $F0 - FF_{16}$. The programming of this address is discussed in the $\overline{ME}$ and $\overline{IC}$ fuse bank sections. The Controller Status register can also be programmed to appear at any or all $\overline{DA}$ bit locations, enhancing the polling process of multiple 8X470 I/O ports.

POLLING

The Controller Status register address and output location can be fuse programmed to allow multiple 8X470 I/O ports to be polled in a single cycle. The 8X470 I/O ports to be polled are programmed with a common Controller Status address, with each status register output available on a different $\overline{DA}$ line to prevent a bus contention. The 8X401 Microcontroller can poll a maximum of eight 8X470 I/O ports in a single cycle, obtaining up to eight bits of status information. The 8X401 Microcontroller's subfield manipulation capability can then be used to isolate a single 8X470 status bit, simplifying the polling procedure and improving the overall system throughput. Figure 3 shows a typical configuration of eight 8X470 I/O ports polled in a single cycle.

ADDRESSING MODES

The 8X470 can be addressed by a microcontroller through either Fast Select or Conventional $\overline{DA}$ select techniques.

Conventional $\overline{DA}$ Select

The 8X470 must be preselected through a conventional $\overline{DA}$ select cycle before either the data or status registers can be accessed by a Microcontroller. During a conventional $\overline{DA}$ select cycle, the 8X470 compares the address on the $\overline{DA}$ bus to the data and status addresses. If either address matches the $\overline{DA}$ address, the appropriate 8X470 latch is set, enabling the port for a data transfer or a polling cycle. Table 1 shows the conventional select operation of the 8X470 $\overline{DA}$ interface.

Fast Select

A fast select cycle is ideal for time critical applications which require port selection and data transfer in a single clock cycle. Additional bits in the 8X401 instruction word are used to execute a fast select cycle. It should be noted that the 8X470 SC input is not driven by the 8X401 SC output and could introduce a minor operational limitation. For an I/O address select instruction, the 8X401 will output a $\overline{DA}$ address and assert SC. However, a fast select configured 8X470 will also drive the $\overline{DA}$ bus, and a bus contention will result. To avoid this contention, the 8X470 should not be fast selected during any I/O address select instruction. Figure 4 shows a typical fast select configuration. Table 2 shows the fast select operation of the 8X470 $\overline{DA}$ bus interface operation.

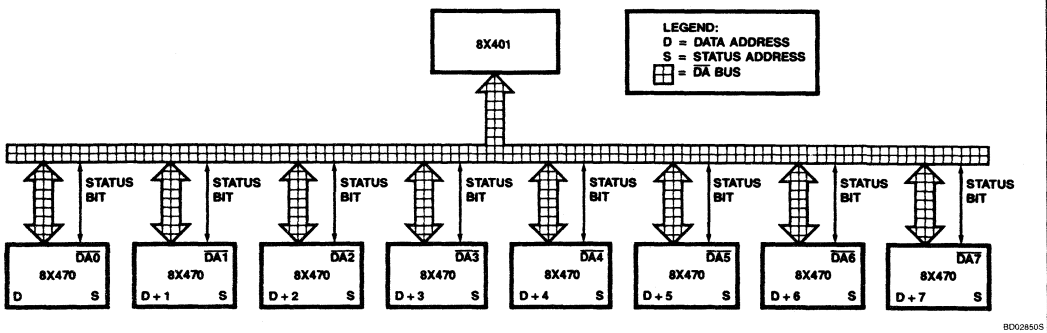


Figure 3. Typical Polling Configuration

Table 1. Conventional $\overline{DA}$ Operation

$\overline{ME}$	SC	WC	MCLK	ADDRESS LATCH	STATUS LATCH	FUNCTION
L	L	L	X	SET	CLEAR	Output data, clear CONTROLLER STATUS
L**	L	H	H	SET	CLEAR	Input 8X401 data, set USER STATUS
L	H	L	H	X	X	Compare enable address
H	X	X	X	X	X	High impedance
L	L	L	X	CLEAR	SET	Output CONTROLLER STATUS
L	L	H	X	CLEAR	SET	High impedance
L	L	H	L	X	X	High impedance
L	H	L	L	SET	X	High impedance
L	L	L	X	CLEAR	CLEAR	High impedance

NOTE:
**If the user and the 8X401 attempt to input data to the I/O port at the same time, the user will overwrite the 8X401 (user priority).

Table 2. Fast Select Operation

$\overline{ME}$	SC	WC	MCLK	FUNCTION
L	H	H	H	Input 8X401 data, set USER STATUS
L	H	L	X	Output data, clear CONTROLLER STATUS
H	L	L	X	Output CONTROLLER STATUS
H	H	X	X	Inactive
L	L	X	X	Inactive

Table 3. 8X470 Fuse Mnemonics

BANK	PROGRAMMABLE FUSES	PROGRAMMED FUNCTION
SC	UD0 – UD7	Input Only
WC	UD0 – UD7	Controller Data Address (CDA0 – CDA7)
$\overline{ME}$	UD0 – UD7	Controller Status Location (CSL0 – CLS7)
$\overline{IC}$	UD0 – UD3	Controller Status Address (CSA0 – CSA3)
$\overline{IC}$	UD4	Fast Select
$\overline{IC}$	UD5	Bidirectional Disable
$\overline{IC}$	UD6	Asynchronous Input

Fuses

The 8X470 has four programmable fuse banks, namely the SC, WC, $\overline{ME}$ and $\overline{IC}$ banks. Each bank is selected by its namesake pin, i.e. the SC pin selects the SC fuse bank, the WC pin selects the WC fuse bank, etc. Each bank has eight fuses, except the $\overline{IC}$ bank, which only has seven. These fuse banks govern different modes of operation, and by selecting the appropriate fuse and fuse bank, the 8X470 can be configured for the modes of operation as mentioned earlier. (See Figure 5 for an illustration of the fuse banks.) The mnemonics for the fuses are listed in Table 3.

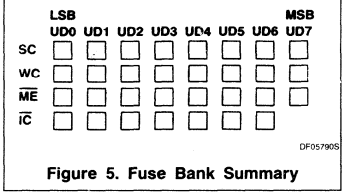


Figure 5. Fuse Bank Summary

$\overline{IC}$ BANK

CSA0 – CSA3

The first four fuses of the $\overline{IC}$ bank (UD0 – UD3) program the four least significant bits of the Controller Status address (CSA0 – 3). The four most significant bits of the Controller Status address are always a 1. An unprogrammed (unblown) CSA fuse is a logic 1 from the $\overline{DA}$ bus, and a programmed (blown) CSA fuse is a logic 0 from the $\overline{DA}$ bus. The Controller Status register can be assigned any one of the addresses F0 – FF (00 – 0F on the $\overline{DA}$ bus). In an unprogrammed part, the status register will be assigned address F0.

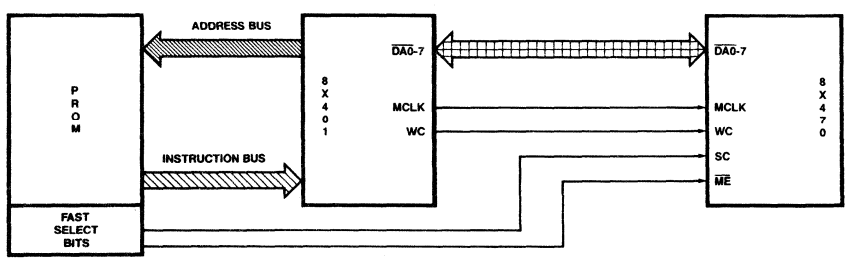


Figure 4. Typical Fast Select Configuration

Fast Select Fuse

The Fast select fuse programs the addressing mode; i.e., Fast or Conventional $\overline{DA}$ select. In an unprogrammed part, Conventional $\overline{DA}$ select is selected.

Bidirectional Disable Fuse

The bidirectional disable fuse selects either Mode A or Mode B operation. In an unprogrammed part, Mode A is selected. Modes A and B are defined more fully in the SC fuse bank.

Asynchronous Input Fuse

The asynchronous input fuse selects the user data entry mode; synchronous or asynchronous. In an unprogrammed part, data is entered synchronously with MCLK.

See Table 4 for a summary of the $\overline{IC}$ bank.

SC Bank

The SC bank has two possible configurations, Mode A and Mode B, as selected by the bidirectional disable fuse. Mode A allows bidirectional and input only UD ports; Mode B allows output only and input only UD ports. Input only UD ports are output only from the $\overline{DA}$ bus, all other types of UD ports (bidirectional and output only) are bidirectional from the $\overline{DA}$ bus at the same bit location. (See Figure 6.)

Mode A

Mode A is selected when the bidirectional disable fuse is unblown, configuring the unprogrammed SC bank for bidirectional operations. (See Table 5.) The user would then blow the needed input only fuses to disable the bidirectional capability of that UD line, making it input only. This line is input only from the UD bus and output only from the $\overline{DA}$ bus. (See Example 1.) Table 6 gives the Mode A operation of the 8X470 user interface.

Table 5. Mode A Bus Operation

INPUT ONLY FUSE "N"	UD "N"	$\overline{DA}$ "N"
Unblown	B	B
Blown	I	O

I = Input only
O = Output only
B = Bidirectional

Table 4. $\overline{IC}$ Fuse Bank Summary

UD PIN	FUNCTION	UNBLOWN FUSE	BLOWN FUSE
UD0 – UD3	Controller Status Address	Logic 1 from the $\overline{DA}$ bus	Logic 0 from the $\overline{DA}$ bus
UD4	Addressing mode	Conventional $\overline{DA}$ Select	Fast Select
UD5	Bidirectional disable	Mode A	Mode B
UD6	Data entry	Synchronous	Asynchronous

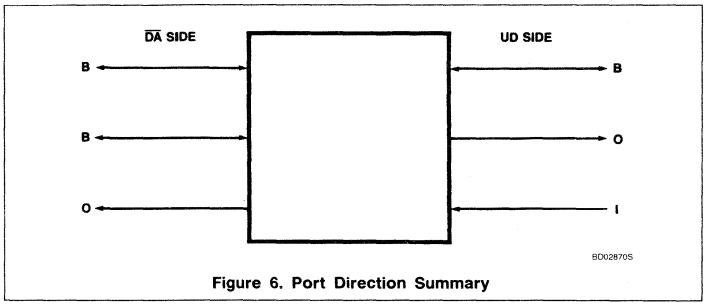
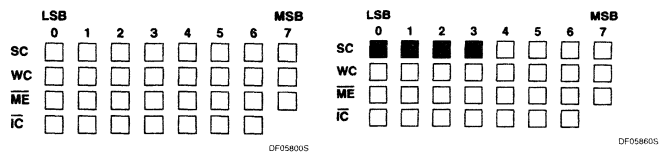


Figure 6. Port Direction Summary

Example 1. Mode A Unprogrammed and Programmed Configurations



BUS SIDE	BIT LOCATION 0	1	2	3	4	5	6	7
UD	B	B	B	B	B	B	B	B
$\overline{DA}$	B	B	B	B	B	B	B	B

BUS SIDE	BIT LOCATION 0	1	2	3	4	5	6	7
UD	I	I	I	I	B	B	B	B
$\overline{DA}$	O	O	O	O	B	B	B	B

Table 6. Mode A Operation (See Note)

BIDIRECTIONAL			INPUT ONLY		
$\overline{OC}$	$\overline{IC}$	Function	$\overline{OC}$	$\overline{IC}$	Function
L	L	High impedance	L	L	High impedance
L	H	Output data, clear user status	L	H	High impedance (UD7 = V_{OH})
H	L	Input user data, set controller status	H	L	Input user data, set controller status
H	H	High impedance (UD7 = User Status)	H	H	High impedance (UD7 = User Status)

NOTE:
This table should only be used if the bidirectional fuse is unblown. For UD lines where the input only fuse is blown, use the input only side; else use the bidirectional side.

Mode B

Mode B is selected when bidirectional disable fuse is blown, configuring the unprogrammed SC bank for output only operation from the UD bus. (See Table 7.) The user would then blow the needed input only fuses to disable the output only capability of that UD line, making it input only. The line is input only from the UD bus and output only from the $\overline{DA}$ bus. (See Example 2). Table 8 gives the Mode B operation of the 8X470 user interface.

Table 7. Mode B Bus Operation

INPUT ONLY FUSE "N"	UD "N"	$\overline{DA}$ "N"
Unblown	O	B
Blown	I	O

I = Input only
O = Output only
B = Bidirectional

WC Bank

The WC fuses (CDA0 – CDA7) are used to program the Data Address for Conventional $\overline{DA}$ select operations. An unprogrammed WC fuse requires a logic 1 at its $\overline{DA}$ bit location to produce an address match. A blown fuse requires a logic 0 at its $\overline{DA}$ bit location to produce an address match. The user can then program the data address to be any one of the addresses 00 – FF. An unprogrammed 8X470 recognizes a $\overline{DA}$ address of FF (00 within the 8X401) as a data address match. See Example 3.

Example 3. Controller Data Register Address

BUS SIDE	LSB 0	1	2	3	4	5	6	MSB 7
UD	□	■	■	■	■	■	■	□
$\overline{DA}$	The controller status register output is fixed at $\overline{DA0}$ and $\overline{DA7}$.							

DF06160S

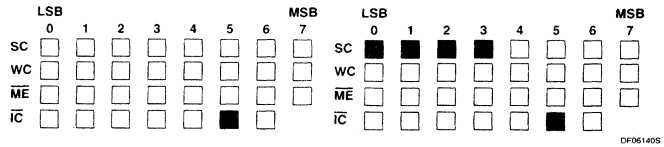
$\overline{ME}$

The $\overline{ME}$ fuse (CSL0 – CSL7) bank positions the output of the Controller Status bit at any unblown UD fuse at the same bit location on the $\overline{DA}$ side. In an unprogrammed part, the Controller Status output will appear on all eight $\overline{DA}$ lines. See Example 4.

PROGRAMMING

Flowchart 1 summarizes the 8X470 programmable features by their function and programming configuration. Flowchart 1 can be used before programming as a guide for mapping the fuses that need to be blown, to configure an 8X470 I/O port for a particular configuration. After the features have been mapped, the programming procedure can begin. Each bank is programmed in a similar manner, with one exception, the $\overline{IC}$ fuse bank. The $\overline{IC}$ fuse bank has two added considerations; when being programmed, any input to UD7 should be removed, and the fuse at UD7 should not be programmed. Fuses can be verified when all programming as been completed or just after a fuse has been programmed. During verification, a blown fuse acts as a high impedance and an unblown fuse acts as a VOL.

Example 2. Mode B Unprogrammed and Programmed Configurations



BUS SIDE	BIT LOCATION 0	1	2	3	4	5	6	7
UD	O	O	O	O	O	O	O	O
$\overline{DA}$	B	B	B	B	B	B	B	B

BUS SIDE	BIT LOCATION 0	1	2	3	4	5	6	7
UD	I	I	I	I	O	O	O	O
$\overline{DA}$	O	O	O	O	B	B	B	B

Table 8. Mode B Operation (See Note)

OUTPUT ONLY			INPUT ONLY		
$\overline{OC}$	$\overline{IC}$	Function	$\overline{OC}$	$\overline{IC}$	Function
L	L	Output data, clear User Status	L	L	Input user data, set Controller Status
L	H	Output data, clear User Status	L	H	High impedance (UD7 = V_{OH})
H	L	High impedance (UD7 = V_{OH})	H	L	Input user data, set Controller Status
H	H	High impedance (UD7 = User Status)	H	H	High impedance (UD7 = User Status)

NOTE:
Table 7 should only be used if the bidirectional fuse has been blown. For UD lines where the input only fuse is blown, use the input only side; else use the output only side.

Example 4. Controller Status Output Location

BUS SIDE	LSB 0	1	2	3	4	5	6	MSB 7
UD	■	□	■	■	□	□	■	■
$\overline{DA}$	0	1	0	0	1	1	0	0
	Data Register Address: CD (Hex)*							

DF06170S

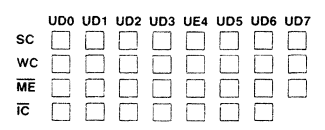
Thus, programming the 8X470 I/O port can be broken down into three stages:

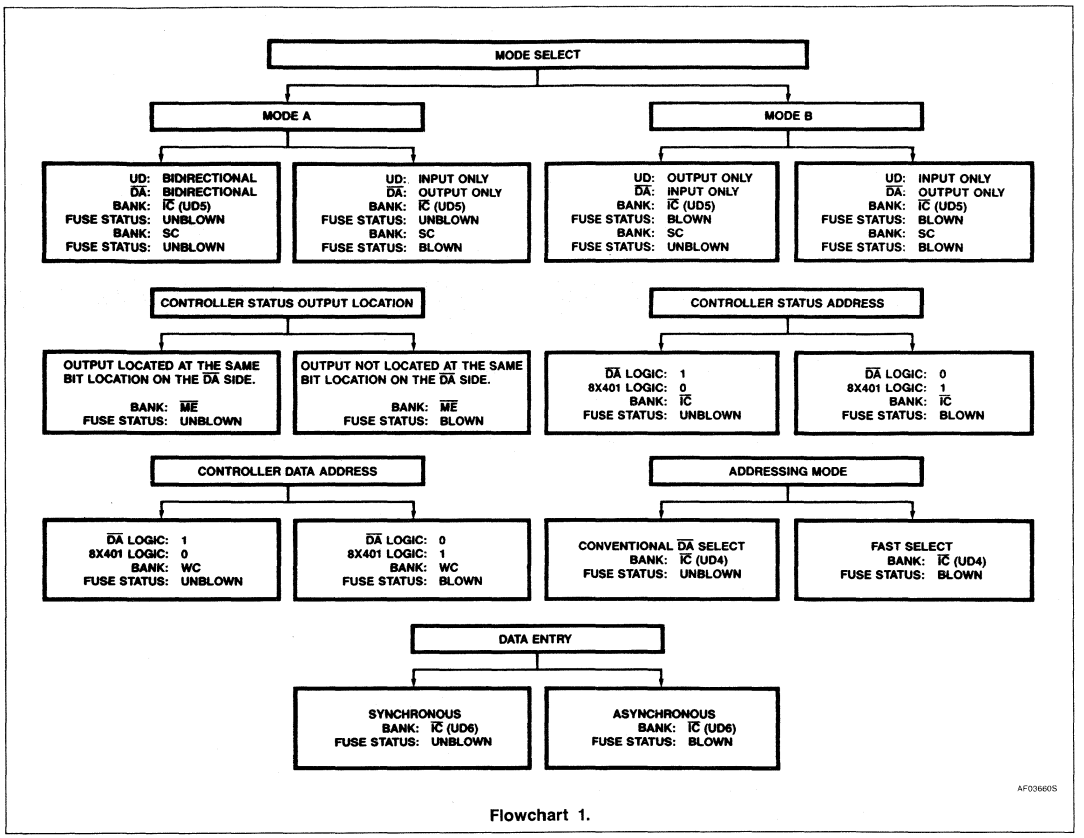
(1) Pre-Programming
   a. Decide what features the 8X470 is to have.
   b. Use Flowchart 1 to determine which fuses need to be blown.
   c. Map the fuses to be blown onto a diagram, like the one shown below, by darkening the squares where fuses are to be blown.

(2) Programming
   Use "map" to blow fuses using the programming procedure.

(3) Verify
   Verify that the fuses blown correspond to the map, using the verification procedure.

For an example, see Programming Example.





Flowchart 1.

Programming Sequence

V_f = 10V

1) V_{CC} = 5V, $\overline{ME}$, SC, $\overline{IC}$ = V_{OL} (initial conditions).
2) $\overline{IC}$ = $\overline{OC}$ = V_{OL}.
3) UD0 – UD7 = V_{OL}.
4) Raise WC to V_f (programming data address).
5) Raise V_{CC} to V_f.
6) Raise UD lines 0 – 7 one at a time where a fuse is to be blown.
7) Lower V_{CC} to 5V.
8) Raise $\overline{OC}$ to V_f.
9) Verify blown fuses on UD0 – UD7. These act as open collector outputs during verification. An intact fuse is indicated by a V_{OL}, a blown fuse by a high impedance.
10) Lower $\overline{OC}$ to V_{OL}.
11) Lower WC to V_{OL}.
12) Raise $\overline{ME}$ to V_f (programming status bit position).
13) Repeat steps 5 through 10 above.
14) Lower $\overline{ME}$ to V_{OL}.
15) Raise SC to V_f (programming data bit direction).
16) Repeat steps 5 through 10 above.
17) Lower SC to V_{OL}.
18) Remove any input from UD7.
19) Raise $\overline{OC}$ to V_{OH}.
20) Raise $\overline{IC}$ to V_f (programming status address and miscellaneous functions).
21) Repeat steps 5 through 10 above, but do not raise UD7 in step 6.
22) Lower $\overline{IC}$ to V_{OL}.
23) Lower $\overline{OC}$ to V_{OL}.
24) Remove power (V_{CC} = 0V).

Programming Verification Sequence

V_f = 10V

1) V_{CC} = 5V, $\overline{ME}$, SC, $\overline{IC}$ = V_{OL} (initial conditions).
2) $\overline{IC}$ = $\overline{OC}$ = V_{OL}.
3) UD0 – UD7 = V_{OL}.
4) Raise WC to V_f (verifying data address).
5) Raise $\overline{OC}$ to V_f.
6) Verify blown fuses on UD0 – UD7. These act as open collector outputs during verification. An intact fuse is indicated by a V_{OL} and a blown fuse by a high impedance.
7) Lower $\overline{OC}$ to V_{OL}.
8) Lower WC to V_{OL}.
9) Raise $\overline{ME}$ to V_f (verifying status bit position).
10) Repeat steps 5 through 7 above.
11) Lower $\overline{ME}$ to V_{OL}.
12) Raise SC to V_f (verifying bit direction).
13) Repeat steps 5 through 7 above.
14) Lower SC to V_{OL}.
15) Raise $\overline{IC}$ to V_f (verifying status address and miscellaneous functions).
16) Repeat steps 5 through 7 above.
17) Lower $\overline{IC}$ to V_{OL}.
18) Remove power (V_{CC} = 0V).

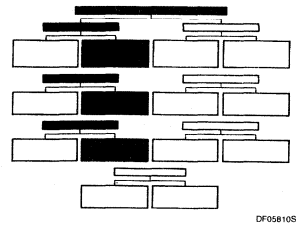
Pre-programming Example

Stage 1 — Pre-programming

(a) Features Desired

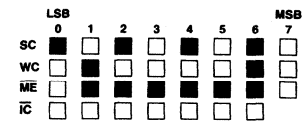
	Conventional $\overline{DA}$
Addressing Mode:	select
Controller Data address:	42 (within the 8X401)
Controller Status address:	F0 (within the 8X401)
Controller Status Output Location:	$\overline{DA0}$, $\overline{DA7}$
Mode of Operation:	A
	UD0, UD2, UD4,
Input only UD lines:	UD6
Data Entry:	Synchronous

(b)The shaded areas of Flowchart 1 indicate the selections from pat 'a' which require a blown fuse.



DF05810S

(c) Map



DF05820S

Programming Example — Stages 2 and 3

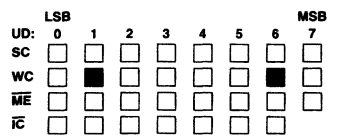
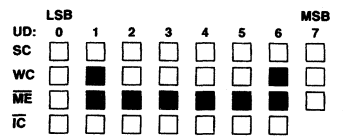
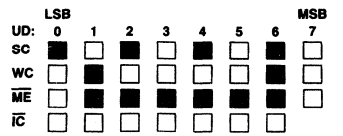
INITIAL CONDITIONS (V_f = 10V)	8X470 CONFIGURATION AFTER PROGRAMMING SEQUENCE
1. V_{CC} = 5V, $\overline{ME}$, SC, $\overline{IC}$ = V_{OL} (initial conditions). 2. $\overline{IC}$ = $\overline{OC}$ = V_{OL}. 3. UD0 – UD7 = V_{OL}.	
DATA ADDRESS 4. Raise WC to V_f. 5. Raise V_{CC} to V_f. 6. Raise UD1 to V_{OH} to blow a fuse. Repeat of UD6. 7. Lower V_{CC} to 5V. 8. Raise $\overline{OC}$ to V_f. 9. Verify that the fuses at UD1 and UD6 are blown. A blown fuse acts like a high impedance and an intact fuse acts like a V_{OL}. 10. Lower $\overline{OC}$ to V_{OL}. 11. Lower WC to V_{OL}.	
CONTROLLER OUTPUT LOCATION 12. Raise $\overline{ME}$ to V_f. 13. Repeat steps 5 – 10 above for UD1 – UD6. 14. Lower $\overline{ME}$ to V_{OL}.	
DATA BIT DIRECTION 15. Raise SC to V_f. 16. Repeat steps 5 – 10 above for UD0, UD2, UD4, UD6. 17. Lower SC to V_{OL}.	

Table 9. Programming Examples

CONFIGURATION LSB MSB UD Bits: 0 1 2 3 4 5 6 7	INPUT/OUTPUT CAPABILITY Bit Location LSB MSB 0 1 2 3 4 5 6 7	DATA ENTRY	ADDRESSING MODE	CONTROLLER STATUS POSITION	CONTROLLER STATUS ADDRESS (within the 8x401)	CONTROLLER DATA ADDRESS (within the 8x401)	MODE OF OPERATION
Example 1 SC □ □ □ □ □ □ □ □ WC □ □ □ □ □ □ □ □ $\overline{ME}$ □ □ □ □ □ □ □ □ $\overline{IC}$ □ □ □ □ □ □ □ □	UD: B B B B B B B B $\overline{DA}$: B B B B B B B B	Synchronous	Conventional $\overline{DA}$ Select	$\overline{DA0}$ – $\overline{DA7}$	F0 (Hex)	00 (Hex)	A
Example 2 SC □ □ □ □ □ □ □ □ WC □ □ □ □ □ □ □ □ $\overline{ME}$ □ ■ ■ ■ ■ ■ ■ □ $\overline{IC}$ □ □ □ □ ■ □ □	UD: B B B B B B B B $\overline{DA}$: B B B B B B B B	Synchronous	Fast Select	$\overline{DA0}$, $\overline{DA7}$	X	X	A
Example 3 SC ■ □ ■ □ ■ □ ■ □ WC □ □ □ □ □ □ □ □ $\overline{ME}$ □ ■ ■ ■ ■ ■ ■ □ $\overline{IC}$ □ □ □ □ ■ □ □	UD: I B I B I B I B $\overline{DA}$: O B O B O B O B	Synchronous	Fast Select	$\overline{DA0}$, $\overline{DA7}$	X	X	A
Example 4 SC □ □ □ □ □ □ □ □ WC □ ■ □ □ □ □ ■ □ $\overline{ME}$ □ ■ ■ ■ ■ ■ ■ □ $\overline{IC}$ □ □ □ □ □ ■ ■	UD: O O O O O O O O $\overline{DA}$: B B B B B B B B	X	Conventional $\overline{DA}$ Select	$\overline{DA0}$, $\overline{DA7}$	F0 (Hex)	42 (Hex)	B
Example 5 SC ■ ■ ■ ■ □ □ □ □ WC □ ■ □ □ □ □ ■ □ $\overline{ME}$ □ ■ ■ ■ ■ ■ ■ □ $\overline{IC}$ □ □ □ □ □ ■ ■	UD: I I I I O O O O $\overline{DA}$: O O O O B B B B	Asynchronous	Conventional $\overline{DA}$ Select	$\overline{DA0}$, $\overline{DA7}$	F0 (Hex)	42 (Hex)	B

ABSOLUTE MAXIMUM RATINGS

SYMBOL	PARAMETER	RATING	UNIT
V_{CC}	Power supply voltage	+7	V_{DC}
V_{IN}	Input voltage	+5.5	V_{DC}
V_O	Off-state voltage	+5.5	V_{DC}
T_{STG}	Storage temperature range	−65 to +150	°C

ORDERING INFORMATION

DESCRIPTION	ORDER CODE
24-Pin Plastic DIP 400 Mil Wide	N8X470N

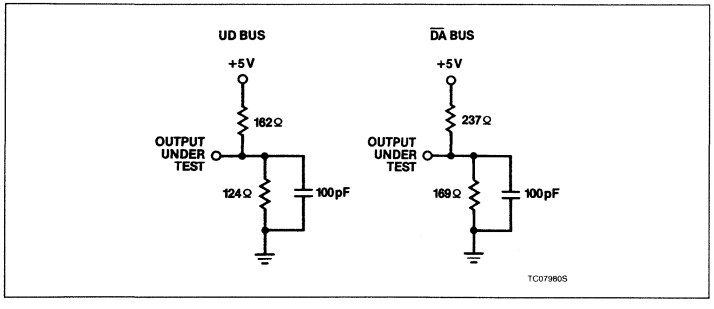
DC ELECTRICAL CHARACTERISTICS (V_{CC} = 5V +5%; 0°C ≤ T_A ≤ 70°C)

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS			UNIT
			Min	Typ	Max	
V_{IL}	Low level input voltage	V_{CC} = Min			0.8	V
V_{IH}	High level input voltage	V_{CC} = Max	2.0			V
V_{OL}	Low level output voltage	V_{CC} = Min I_{OL} = 16 mA from $\overline{DA}$ bus I_{OL} = 24 mA from UD bus			0.55	V
V_{OH}	High level output voltage	V_{CC} = Min, I_{OH} = −3mA	2.5			V
V_{CL}	Input clamp voltage	V_{CC} = Min, I_{IN} = −10mA			−1.5	V
I_{IL}	Low level input current	V_{CC} = Max, V_I = 0.4V			−0.4	mA
I_{IL}	Low level input current for $\overline{IC}$, $\overline{ME}$, SC	V_{CC} = Max, V_I = 0.4V			−0.7	mA
I_{IH}	High level input current	V_{CC} = Max, V_I = 2.7V			20	μA
I_{OZL}	High-Z output current (low level)	V_{CC} = Max, V_O = 0.4V			−0.4	mA
I_{OZH}	High-Z output current (high level)	V_{CC} = Max, V_O = 2.7V			50	μA
I_{OS}	Short circuit output current	V_{CC} = Min	−30		−140	mA
I_{CC}	Supply current	V_{CC} = Max			220	mA
V_{CC}	Supply voltage		4.75	5.00	5.25	V

DC FUSING CHARACTERISTICS

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS			UNIT
			Min	Typ	Max	
V_F	Fusing voltage	Applies to V_{CC}, SC, WC, $\overline{ME}$ and $\overline{IC}$ only	9.5		11.5	V
I_{FB}	Fuse bank current	$\overline{ME}$ or SC or WC, $\overline{IC}$ = 11.5V (Only one bank at a time can equal 11.5V)				mA
I_{CCF}	Fusing current	V_{CC} = 11.5V, $\overline{ME}$ or SC or WC or $\overline{IC}$ = 11.5V (Only one bank at a time can equal 11.5 V)				mA

TEST LOADING CIRCUITS

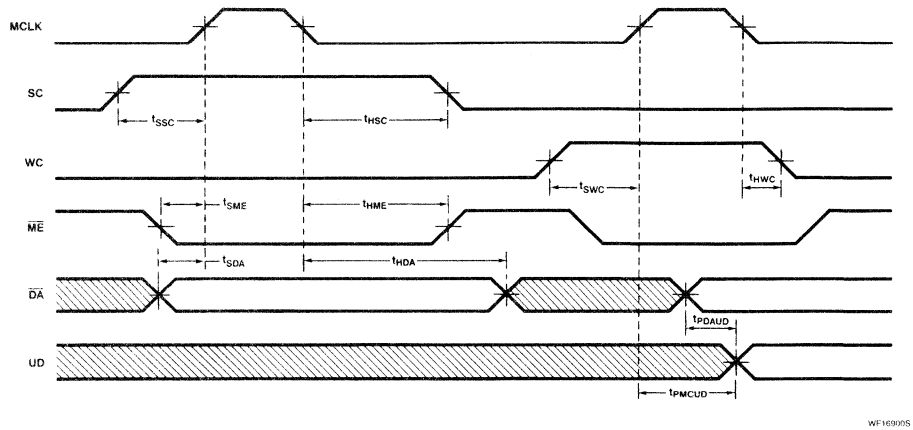


AC ELECTRICAL CHARACTERISTICS (V_{CC} = 5V +5%; 0°C ≤ T_A ≤ 70°C) (See Note 1)

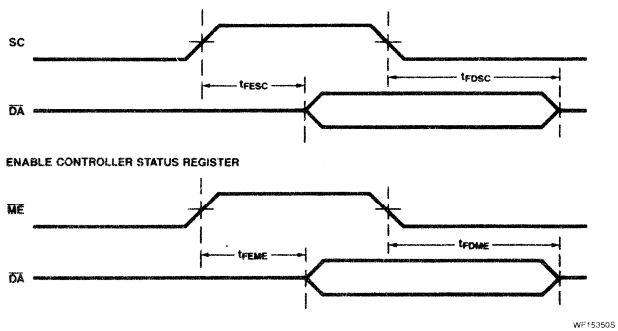
SYMBOL	PARAMETER	REFERENCES		TEST CONDITIONS	LIMITS		UNIT
		From	To		Min	Max	
Pulse Widths:							
t_{WMC}	MCLK pulse width			CS = MS = Low	25		ns
t_{WIC}	$\overline{IC}/\overline{OC}$ pulse width			$\overline{ME}$ = High	25		ns
Propagation Delays:							
t_{PUDDA}	Propagation delay from UD to $\overline{DA}$	UD	$\overline{DA}$			35	ns
t_{PICDA}	Propagation delay from $\overline{IC}$ to $\overline{DA}$	$\overline{IC}$ ↓	$\overline{DA}$			35	ns
t_{PMCDA}	Propagation delay from MCLK to $\overline{DA}$	MCLK ↑	$\overline{DA}$			35	ns
t_{PDAUD}	Propagation delay from $\overline{DA}$ to UD	$\overline{DA}$	UD			35	ns
t_{PMCUD}	Propagation delay from MCLK to UD	MCLK ↑	UD			35	ns
Setup Times:							
t_{SSC}	Setup time for SC	SC ↑	MCLK ↑		5		ns
t_{SWC}	Setup time for WC	WC ↑	MCLK ↑		5		ns
t_{SME}	Setup time for $\overline{ME}$	$\overline{ME}$ ↓	MCLK ↑		5		ns
t_{SDA}	Setup time for $\overline{DA}$	$\overline{DA}$	MCLK ↑		0		ns
t_{SIC}	Setup time for $\overline{IC}/\overline{OC}$	$\overline{IC}$ ↓	MCLK ↑		0		ns
t_{SUDIC}	Setup time to enable $\overline{IC}$	UD	$\overline{IC}$ ↓		0		ns
Hold Times:							
t_{HSC}	Hold time for SC	MCLK ↓	SC ↓		0		ns
t_{HWC}	Hold time for WC	MCLK ↓	WC ↓		0		ns
t_{HME}	Hold time for $\overline{ME}$	MCLK ↓	$\overline{ME}$ ↑		0		ns
t_{HIC}	Hold time for $\overline{IC}/\overline{OC}$	MCLK ↓	$\overline{IC}$ ↑		0		ns
t_{HUDIC}	Hold time for between $\overline{IC}$ and UD	$\overline{IC}$ ↑	UD		15		ns
t_{HDA}	Hold time for $\overline{DA}$ bus	MCLK ↓	DA		10		ns
Enable Times:							
t_{EIC}	Enable time for $\overline{IC}$	$\overline{IC}$ ↑	UD output			24	ns
t_{EME}	Enable time for $\overline{ME}$	$\overline{ME}$ ↓	$\overline{DA}$	SC = WC = Low		24	ns
t_{ESC}	Enable time for SC	SC ↓	$\overline{DA}$	WC = $\overline{ME}$ = Low		24	ns
t_{EWC}	Enable time for WC	WC ↓	$\overline{DA}$	SC = $\overline{ME}$ Low		24	ns
t_{ESIC}	Enable time for user status	$\overline{IC}$ ↑	UD7			24	ns
t_{FESC}	Enable time for data in the fast select mode	SC ↑	$\overline{DA}$			24	ns
t_{FEME}	Enable time for the controller status in the fast select mode	$\overline{ME}$ ↑	$\overline{DA}$			24	ns
Disable Times: (see note 2)							
t_{DIC}	Disable time for $\overline{IC}$	$\overline{IC}$ ↓	UD bus tri-stated			45	ns
t_{DME}	Disable time for $\overline{ME}$	$\overline{ME}$ ↑	$\overline{DA}$ bus tri-stated			40	ns
t_{DSC}	Disable time for SC	SC ↑	$\overline{DA}$ bus tri-stated			40	ns
t_{DWC}	Disable time for WC	WC ↑	$\overline{DA}$ bus tri-stated			40	ns
t_{DSIC}	Disable time for user status	$\overline{IC}$ ↓	UD7 tri-stated			40	ns
t_{FDSC}	Disable time for data in the fast select mode	SC ↓	$\overline{DA}$ bus tri-stated			40	ns
t_{FDME}	Disable time for controller status in the fast select mode	$\overline{ME}$ ↓	$\overline{DA}$ bus tri-stated			40	ns

NOTES:
1. Input levels swing from 0V to 3V with Outputs measured at 1.5V.
2. These parameters are measured with a capacitive loading of 100pF.

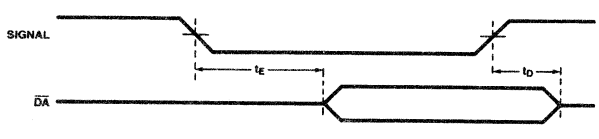
8X470 Timing for the Conventional DA Select Addressing Mode



8X470 Timing for the Fast Select Addressing Mode



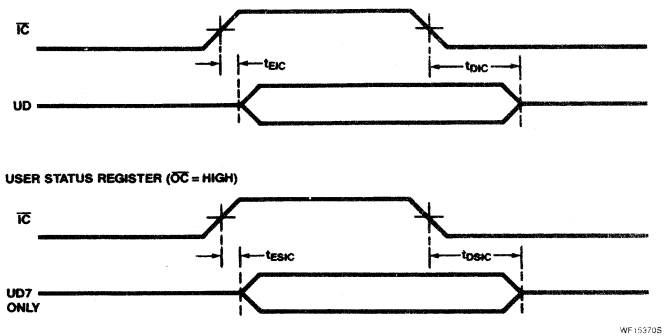
Enable and Disable Timing at the DA Ports



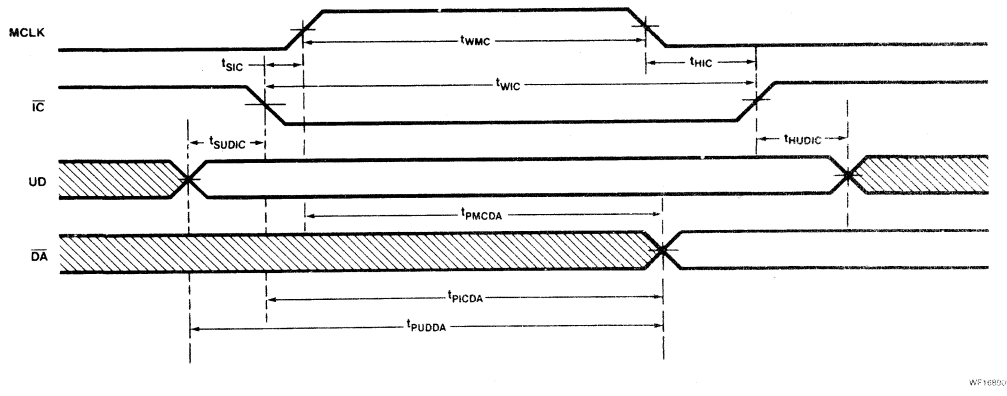
Enable and Disable Signal Summary

SIGNAL	PARAMETERS		CONDITIONS	AFFECT
	Enable	Disable		
$\overline{ME}$	t_{EME}	t_{DME}	SC = WC = LOW	READ
WC	t_{EWC}	t_{DWC}	SC = $\overline{ME}$ = LOW	READ
SC	t_{ESC}	t_{DSC}	WC = $\overline{ME}$ = LOW	READ

Enable and Disable Timing at the UD Ports



8X470 Timing-Propagation Delays, Pulse Widths and Hold Times



8X470 Fuse Coding Sheet

This coding sheet is intended to assist in determining the fuse coding of an 8X470 Programmable I/O Port for a particular application. Use of this sheet requires some prior knowledge of the 8X470 fuse – programmable functions. Refer to the 8X470 Data Sheet for more information. To produce a fuse code, simply follow the steps below. Some of the information will start out in binary form, and must later be converted to hexadecimal. The chart at the right is provided to simplify the conversion process.

1. For conventional (DA bus) port selection, choose a Controller Data Address in the range of 0 to FF hexadecimal. Write the address in the space to the right. If Fast Select will be used, enter 00.
2. Pick the Controller Status Locations. Write a 1 in those positions where status should NOT appear and 0 in the other positions.
3. Select any bit positions that are to be programmed as input only. Write a 1 in any positions that should be input only, 0 in the other positions.
4. For conventional (DA bus) port selection, choose the Controller Status Address in the range of F0 to FF hexadecimal. Write the bottom digit (0 to F) of the address in the space to the right. If Fast Select mode will be used, enter a 0.
5. If the 8X470 will be used in the Fast Select mode (as opposed to the conventional select mode), enter a 1 in the space to the right, otherwise enter a 0.
6. If the 8X470 will be used in the directional mode (Mode B) rather than the bidirectional mode (Mode A), enter a 1 in the space to the right, otherwise enter a 0.
7. If it is desired that data be written to the 8X470 from the User Bus asynchronously with respect to MCLK, enter a 1 in the space to the right, otherwise a 0 should be entered.
8. Copy the data from lines 4 through 7 into the boxes to the right. As noted, the first bit is always a zero. The four bits plus the single hex digit form one byte of programming data.
9. Copy the data from lines 1 to 3 and 8 into the spaces on the right, converting any binary data to hexadecimal. This is the data that is required to order or field-program an 8X470.

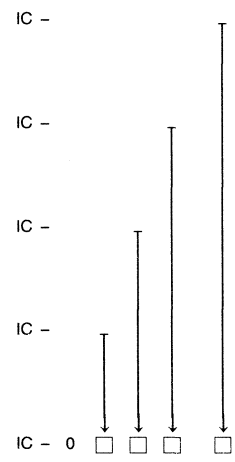
Binary to Hex Conversion

0000 — 0	1000 — 8
0001 — 1	1001 — 9
0010 — 2	1010 — A
0011 — 3	1011 — B
0100 — 4	1100 — C
0101 — 5	1101 — D
0110 — 6	1110 — E
0111 — 7	1111 — F

WC – ____

	7	6	5	4	3	2	1	0
ME –	–	–	–	–	–	–	–	–

	7	6	5	4	3	2	1	0
SC –	–	–	–	–	–	–	–	–



Fuse Code:

WC		ME		SC		IC
___	–	___	–	___	–	___

8-BIT LATCHED BIDIRECTIONAL I/O PORT

Originally published by Signetics January 1984

FEATURES

- **Dual bidirectional ports**
- **Independent port operation (User-port priority for data entry)**
- **User data input synchronous**
- **At power-up, User-port outputs are high and Microprocessor-port outputs are high-Z**
- **Three-state TTL outputs for high-drive capabilities**
- **Directly compatible with 8X300 Microcontroller**
- **Single +5V supply**

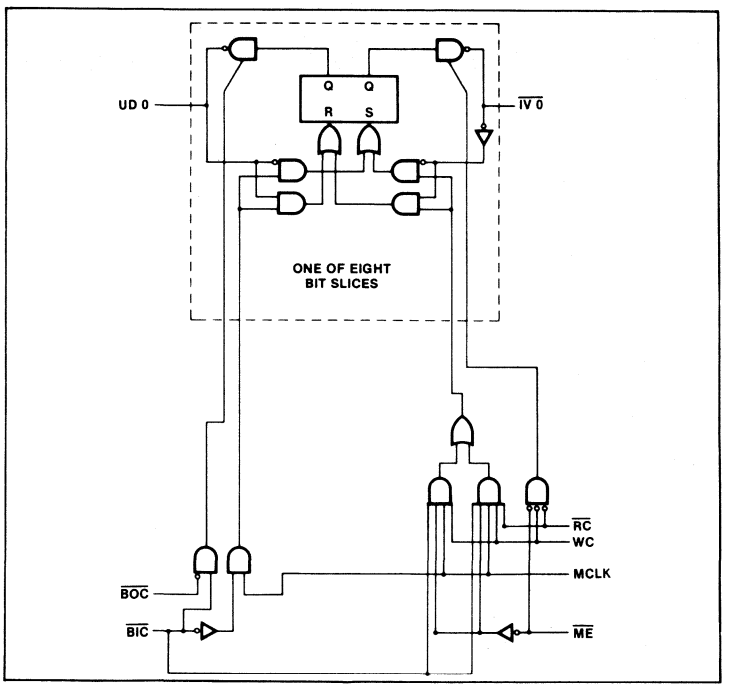
PRODUCT DESCRIPTION

The 8T31 is an 8-bit bidirectional data register designed to function as Input/Output interface elements in microprocessor systems.

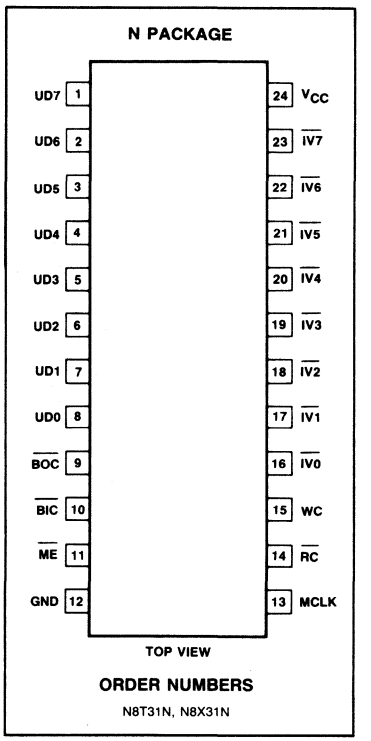
Each part contains eight clocked data latches that are accessible from either a *microprocessor* port or a *user* port. Separate I/O control is provided for each port. The two ports operate independently, except that when both are attempting to input data into the data latches, the User port (UD0-UD7) has priority. The master enable (ME) signal enables or disables the microprocessor bus regardless of the state of the other inputs but has no effect on the user bus.

A unique feature of these parts is their ability to start up in a predetermined state. If the clock is maintained at a level of less than 0.8 volts until the power supply reaches 3.5 volts, all bits of the user port will wakeup at a "logic 1" level and those of the microprocessor port will wakeup in the high-impedance state.

LOGIC DIAGRAM



PIN CONFIGURATION



ORDER NUMBERS

N8T31N, N8X31N

PIN DESIGNATION

PIN	SYMBOL	NAME AND FUNCTION	TYPE
1-8	UD0-UD7:	User Data I/O Lines. Bidirectional data lines to communicate with user's equipment.	Acitve high three-state
16-23	$\overline{IV0}$-$\overline{IV7}$:	Microprocessor Bus. Bidirectional data lines to communicate with controlling digital system.	Active low three-state
10	$\overline{BIC}$:	Input Control. User input to control writing into the I/O Port from the user data lines.	Active low
9	$\overline{BOC}$:	Output Control. User input to control reading from the I/O Port onto the user data lines.	Active low
11	$\overline{ME}$:	Master Enable. System input to enable or disable all other system inputs and outputs. It has no effect on user inputs and outputs.	Active low
15	WC:	Write Command. When WC is high, stores contents of $\overline{IV0}$-$\overline{IV7}$ as data.	Active high
14	$\overline{RC}$	Read Command. When RC is low, data is presented on $\overline{IV0}$-$\overline{IV7}$.	Active low
13	MCLK:	Master Clock. Input to strobe data into the latches. See function tables for details.	Active high
24	V_{CC}:	5V power connection.	
12	GND:	Ground.	

Table 1. USER PORT CONTROL FUNCTION

$\overline{BIC}$	$\overline{BOC}$	MCLK	USER DATA BUS FUNCTION
H	L	X	Output Data
L	X	H	Input Data
H	H	X	Inactive

H = High Level L = Low Level X = Don't care

Table 2. MICROPROCESSOR PORT CONTROL FUNCTION

$\overline{ME}$	$\overline{RC}$	WC	MCLK	$\overline{BIC}$	MICROPROCESSOR BUS FUNCTION
L	L	L	X	X	Output Data
L	X	H	H	H	Input Data
X	H	L	X	X	Inactive
X	X	H	X	L	Inactive
H	X	X	X	X	Inactive

USER DATA BUS CONTROL

The activity of the user data bus is controlled by the $\overline{BIC}$ and $\overline{BOC}$ inputs as shown in Table 1.

The user data input is a synchronous function with MCLK. A low level on the $\overline{BIC}$ input allows data on the user data bus to be written into the data latches only if MCLK is at a high level. A low level on the BIC input allows data on the user data bus to be latched regardless of the level of the MCLK input.

To avoid conflicts at the data latches, input from the microprocessor port is inhibited when BIC is at a low level. Under all other conditions the 2 ports operate independently.

MICROPROCESSOR BUS CONTROL

As is shown in Table 2, the activity of the microprocessor port is controlled by the $\overline{ME}$, RC, WC and $\overline{BIC}$ inputs, as well as the state of an internal status latch. $\overline{BIC}$ is included to show user port priority over the microprocessor port for data input.

BUS OPERATION

Data written into the 8T31 from one port will appear inverted when read from the other port. Data written into the 8T31 from one port will not be inverted when read from the same port.

DC ELECTRICAL CHARACTERISTICS

V_{CC} = 5V ± 5%, 0°C ≤ T_A ≤ 70°C unless otherwise specified.

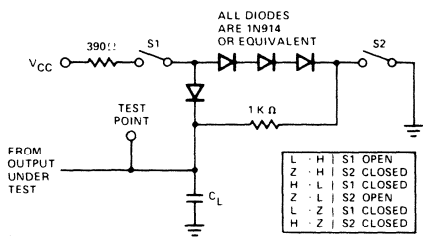
PARAMETER		TEST CONDITIONS	LIMITS			UNIT
			Min	Typ	Max	
Input voltage:						V
V_{IH}	High		2.0			
V_{IL}	Low				.8	
V_{IC}	Clamp	I_I = 5mA			-1	
Output voltage:		V_{CC} = 4.75V				V
V_{OH}	High		2.4			
V_{OL}	Low				.55	
Input current[1]:		V_{CC} = 5.25V				μA
I_{IH}	High	V_{IH} = 5.25V		<10	100	
I_{IL}	Low	V_{IL} = .5V		-350	-550	
Output current[2]:						mA
I_{OS}	Short circuit	V_{CC} = 4.75V				
	UD bus		10			
	IV bus		20			
I_{CC}	VCC supply current	V_{CC} = 5.25V		100	150	mA

NOTES

1. The input current includes the three-state/open collector leakage current of the output driver on the data lines.
2. Only one output may be shorted at a time.

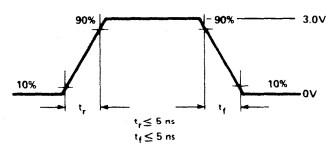
PARAMETER MEASUREMENT INFORMATION

LOAD CIRCUIT FOR THREE-STATE OUTPUTS

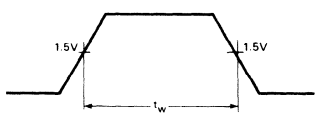


NOTE: C_L includes fixture capacitance.

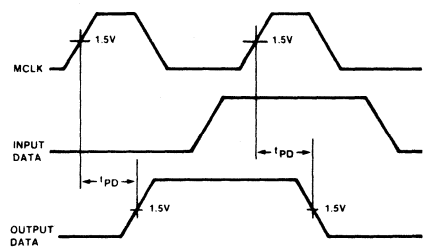
INPUT WAVEFORM



CLOCK PULSE WIDTH



DATA DELAY TIMES
Clock Referenced



AC ELECTRICAL CHARACTERISTICS $0°C \leq T_A \leq 70°C$, $V_{CC} = 5V \pm 5\%$

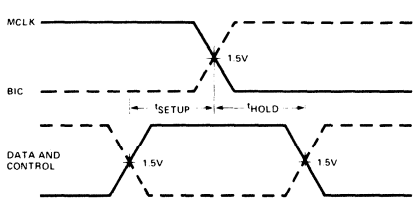
PARAMETERS		INPUT	TEST CONDITIONS	LIMITS			UNIT
				Min	Typ	Max	
		UD X			25	38	ns
t_{PD}	User data relay[1]	MCLK	C_L = 50pF		45	61	ns
t_{OE}	User output enable	$\overline{BOC}$	C_L = 50pF	18	26	47	ns
		$\overline{BIC}$		18	28	35	ns
t_{OD}	User output disable	BOC	C_L = 50pF	16	23	33	ns
		$\overline{IV}$ X			38	53	ns
t_{PD}	µP data delay[1]	MCLK	C_L = 50pF		48	61	ns
		$\overline{ME}$					
t_{OE}	µP output enable	$\overline{RC}$	C_L = 50pF	14	19	25	ns
		WC					
		$\overline{ME}$					
t_{OD}	µP output disable	$\overline{RC}$	C_L = 50pF	13	17	32	ns
		WC					
t_W	Minimum pulse width	MCLK		40			ns
		$\overline{UD}$ X[3]		15			
		$\overline{BIC}$		25			
		$\overline{IV}$ X		55			
t_{SETUP}	Minimum setup time[2]	$\overline{ME}$		30			ns
		$\overline{RC}$		30			
		WC		30			
		$\overline{UD}$ X[3]		25			
		$\overline{BIC}$		10			
		$\overline{IV}$ X		10			
t_{HOLD}	Minimum hold time[2]	$\overline{ME}$		5			ns
		$\overline{RC}$		5			
		WC		5			

NOTES

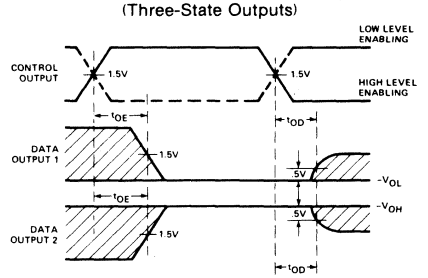
1. Data delays referenced to the clock are valid only if the input data is stable at the arrival of the clock and the hold time requirement is met.
2. Set up and hold times given are for "normal" operation. $\overline{BIC}$ setup and hold times are for a user write operation. $\overline{RC}$ setup and hold times are for an I/O Port select operation. $\overline{ME}$ and WC setup and hold times are for a microprocessor bus write operation.
3. Times are referenced to MCLK.

VOLTAGE WAVEFORMS

SETUP AND HOLD TIMES



OUTPUT ENABLE AND DISABLE TIMES
(Three-State Outputs)



WAVEFORM #1 IS FOR AN OUTPUT WITH INTERNAL CONDITIONS SUCH THAT THE OUTPUT IS LOW WHEN THE THREE-STATE DRIVER IS ENABLED. WAVEFORM #2 IS FOR THE OPPOSITE CONDITION.

8-BIT LATCHED ADDRESSABLE BIDIRECTIONAL I/O PORT

Originally published by Signetics January 1984

FEATURES

- **Independent port operation (user-port priority for data entry)**
- **User data input available as synchronous (8T32) or as asynchronous (8T36)**
- **User data bus available with three-state (8T32, 8T36)**
- **At power-up, user-port outputs are high and microprocessor-port outputs are high-z; status latch (from address compare) is also cleared at power-up**
- **Three-state TTL outputs for high-drive capabilities**
- **Directly compatible with 8X300 microcontroller**
- **Single +5V supply**

PRODUCT IDENTITY

8T32— Three-state, field-programmable (addresses 0-255), synchronous user port.

8T36—Three-state, field-programmable (addresses 0-255), asynchronous user port

PRODUCT DESCRIPTION

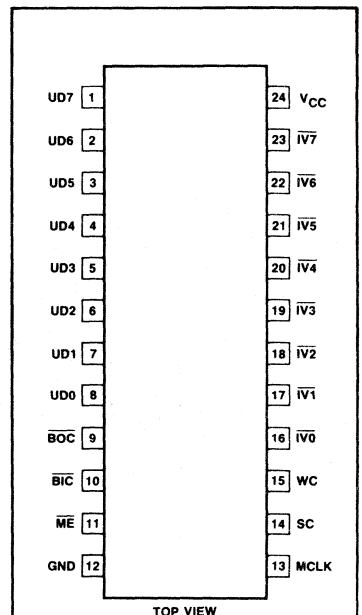
8T32/8T36. Each of these I/O Bytes is an addressable and bi-directional register designed for use as an interface element in any system with TTL-compatible buses. (*Note.* Since these I/O Bytes are frequently used with the 8X300 Microcontroller and its associated Interface Vector bus, the 8T32-8T36 family of parts are commonly called IV Bytes.) Each I/O Byte contains eight identical data latches (Bits 0 through 7); the latches are accessed from either of two 8-bit ports—one port connecting to the microprocessor (8X300) and the other port connecting to the user device.

Separate controls are provided for each port and the two ports operate independently, except when both attempt to input data at the same time; in this case, the user port bus has priority.

The address of each I/O Byte is field-programmable and the microprocessor port is accessed when a valid address is received; the user port is accessible at all times. A selected Byte is automatically deselected when the address of another I/O Byte is sensed on the address/data bus. A Master Enable (ME) input is available for use as a ninth address bit, allowing direct access to 512 I/O Bytes without address decoding.

A unique feature of these parts is their ability to start up in a predetermined state. If the clock is maintained at a level of less than 0.8 volts until the power supply reaches 3.5 volts, all bits of the user port will wakeup at a "logic 1" level and those of the microprocessor port will wakeup in the high-impedance state.

PIN CONFIGURATION



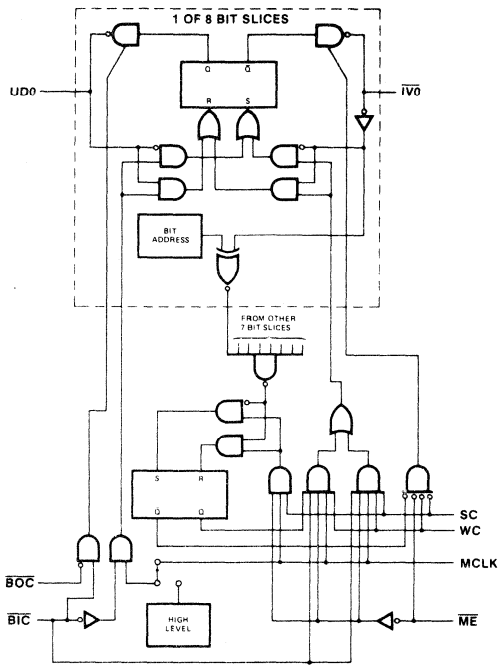
TOP VIEW

ORDER NUMBERS

CONTACT LOCAL SALES REPRESENATIVE

A stock of 8T32s and 8T36s with addresses "1" through "10" are maintained in inventory; with a longer lead time, a small quantity of address "11" through "50" are also available.

TYPICAL BLOCK DIAGRAM



*Switch indicates synchronous/asynchronous user write option. Switch shown for synchronous version.

PIN DESCRIPTION

PIN	SYMBOL	NAME AND FUNCTION	TYPE
1-8	UD0-UD7:	User Data I/O Lines. Bidirectional data lines to communicate with user's equipment. Either tri-state or open collector outputs are available.	Active high
16-23	$\overline{IV0}$-$\overline{IV7}$:	Microprocessor Bus. Bidirectional data lines to communicate with controlling digital system (microprocessor).	Active low three-state
10	$\overline{BIC}$:	Input Control. User input to control writing into the I/O Port from the user data lines.	Active low
9	$\overline{BOC}$:	Output Control. User input to control reading from the I/O Port onto the user data lines.	Active low
11	$\overline{ME}$:	Master Enable. System input to enable or disable all other system inputs and outputs. It has no effect on user inputs and outputs.	Active low
15	WC:	Write Command. When WC is high and SC is low, I/O Port, if selected, stores contents of $\overline{IV0}$-$\overline{IV7}$ as data.	Active high
14	SC:	Select Command. When SC is high and WC is low, data on $\overline{IV0}$-$\overline{IV7}$ is interpreted as an address. I/O Port selects itself if its address is identical to μP bus data; it de-selects itself otherwise.	Active high
13	MCLK:	Master Clock. Input to strobe data into the latches. See function tables for details.	Active high
24	VCC:	5V power connection.	
12	GND:	Ground.	

Table 1. USER PORT CONTROL FUNCTION

$\overline{BIC}$	$\overline{BOC}$	MCLK	USER DATA BUS FUNCTION	
			8T32	8T36
H	L	X	Output Data	Output Data
L	X	H	Input Data	Input Data
L	X	L	Inactive	Input Data
H	H	X	Inactive	Inactive

H = High Level L = Low Level X = Don't care

Table 2. MICROPROCESSOR PORT CONTROL FUNCTION

$\overline{ME}$	SC	WC	MCLK	$\overline{BIC}$	STATUS LATCH	I/O PORT FUNCTION
L	L	L	X	X	SET	Output Data
L	L	H	H	H	SET	Input Data
L	H	L	H	X	X	Input Address
L	H	H	H	L	X	Input Address
L	H	H	H	H	X	Input Data and Address
L	X	H	L	X	X	Inactive
L	H	X	L	X	X	Inactive
L	L	H	H	L	X	Inactive
L	L	X	X	X	Not Set	Inactive
H	X	X	X	X	X	Inactive

USER DATA BUS

The activity of the user data bus is controlled by the $\overline{BIC}$ and $\overline{BOC}$ inputs as shown in Table 1.

For the 8T32, user data input is a synchronous function with MCLK. A low level on the $\overline{BIC}$ input allows data on the user data bus to be written into the data latches only if MCLK is at a high level. For the 8T36, user data input is an asynchronous function. A low level on the $\overline{BIC}$ input allows data on the user data bus to be latched regardless of the level of the MCLK input. Note that when the 8T36, is used with the 8X300 Microcontroller, care must be taken to insure that the Microprocessor bus is stable when it is being read by the 8X300 Microcontroller.

To avoid conflicts at the Data Latches, input from the Microprocessor Port is inhibited when $\overline{BIC}$ is at a low level. Under all other conditions the two ports operate independently

MICROPROCESSOR BUS CONTROL

As is shown in Table 2, the activity of the microprocessor port is controlled by the $\overline{ME}$, SC, WC, and $\overline{BIC}$ inputs, as well as the state of an internal status latch. $\overline{BIC}$ is included to show user port priority over the microprocessor port for data input.

Each I/O Port's status latch stores the result of the most recent I/O Port select; it is set when the I/O Port's internal address matches the Microprocessor Bus. It is cleared when an address that differs from the internal address is presented on the Microprocessor Bus.

In normal operation, the state of the status latch acts like a master enable; the microprocessor port can transfer data only when the status latch is set.

When SC and WC are both high, data on the Microprocessor Bus is accepted as data, whether or not the I/O Port was selected. The data is also interpreted as an address. The I/O Port sets its select status if its address matches the data read when SC and WC were both high; it resets its select status otherwise.

BUS OPERATION

Data written into the I/O Port from one port will appear inverted when read from the other port. Data written into the I/O Port from one port will not be inverted when read from the same port.

AC ELECTRICAL CHARACTERISTICS $0°C \leq T_A \leq 70°C$, $V_{CC} = 5V \pm 5\%$

PARAMETER	INPUT	TEST CONDITION	LIMITS			UNIT
			Min	Typ	Max	
t_{PD} User data delay (Note 1)	UD X	$C_L = 50pF$		25	38	ns
	MCLK*			45	61	
	$\overline{BIC}$†			40	55	
t_{OE} User output enable	$\overline{BOC}$	$C_L = 50pF$	18	26	47	ns
t_{OD} User output disable	$\overline{BIC}$	$C_L = 50pF$	18	28	35	ns
	$\overline{BOC}$		16	23	33	
t_{PD} µP data delay (Note 1)	$\overline{IV\ X}$	$C_L = 50pF$		38	53	ns
	MCLK			48	61	
t_{OE} µP output enable	$\overline{ME}$	$C_L = 50pF$	14	19	25	ns
	SC					
	WC					
t_{OD} µP output disable	$\overline{ME}$	$C_L = 50pF$	13	17	32	ns
	SC					
	WC					
t_W Minimum pulse width	MCLK		40			ns
	$\overline{BIC}$†		35			
t_{SETUP} Minimum setup time	UD X□	(Note 2)	15			ns
	$\overline{BIC}$*		25			
	$\overline{IV\ X}$		55			
	$\overline{ME}$		30			
	SC		30			
	WC		30			
t_{HOLD} Minimum hold time	UD X□	(Note 2)	25			ns
	$\overline{BIC}$*		10			
	$\overline{IV\ X}$		10			
	$\overline{ME}$		5			
	SC		5			
	WC		5			

• Applies for 8T32.
† Applies for 8T36.
□ Times are referenced to MCLK for 8T32, and are referenced to BIC for 8T36.

NOTES:

1. Data delays referenced to the clock are valid only if the input data is stable at the arrival of the clock and the hold time requirement is met.
2. Set up and hold times given are for "normal" operation. $\overline{BIC}$ setup and hold times are for a user write operation. SC setup and hold times are for I/O Port select operation. $\overline{ME}$ setup and hold times are for both IV write and select operations.

DC ELECTRICAL CHARACTERISTICS $0°C \leq T_A \leq 70°C$, $V_{CC} = 5V \pm 5\%$

PARAMETER		TEST CONDITIONS	LIMITS			UNITS
			Min	Typ	Max	
V_{IH}	High-level input voltage		2.0		5.5	V
V_{IL}	Low-level input voltage		-1.0		.8	V
V_{CL}	Input clamp voltage	$I_I = -5mA$			-1.0	V
I_{IH}	High-level input current[1]	$V_{CC} = 5.25V$ $V_{IH} = 5.25V$		<10	100	μA
I_{IL}	Low level input current[1]	$V_{CC} = 5.25V$ $V_{IL} = .5V$		-350	-550	μA
V_{OL}	Low-level output voltage	$V_{CC} = 4.75V$ $I_{OL} = 16mA$			.55	V
V_{OH}	High-level output voltage	$V_{CC} = 4.75V$ $I_{OH} = -3.2mA$	2.4			V
I_{OS}	Short-circuit output current[2] UD bus IV bus	 $V_{CC} = 4.75V$ $V_{CC} = 4.75V$	 10 20			 mA mA
I_{CC}	Supply current	$V_{CC} = 5.25V$		100	150	mA

NOTES

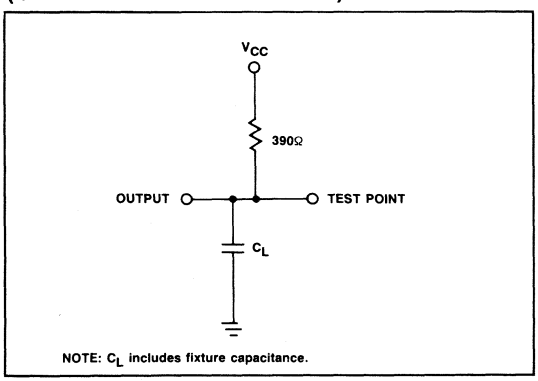
1. The input current includes the Three-state/Open Collector leakage current of the output driver on the data lines.
2. Only one output may be shorted at a time.
3. These limits do not apply during address programming.

Absolute Maximum Ratings:

Supply voltage[3] 7V

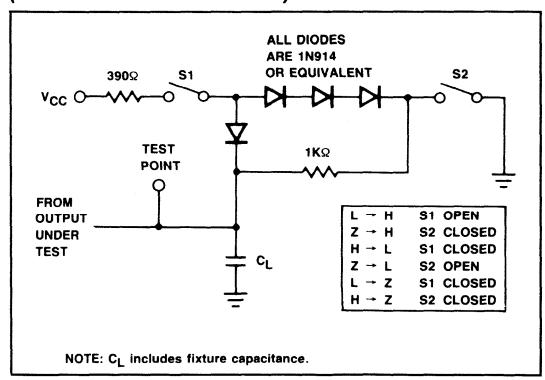
Input voltage[3] 5.5V

TEST LOAD CIRCUIT (OPEN COLLECTOR OUTPUTS)



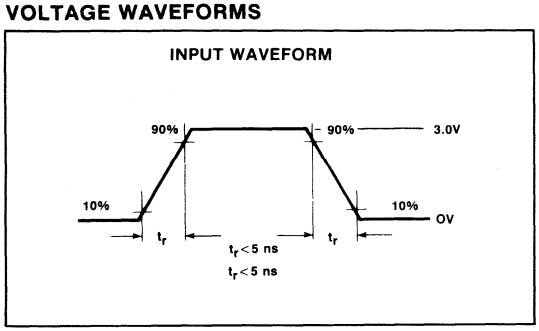
NOTE: C_L includes fixture capacitance.

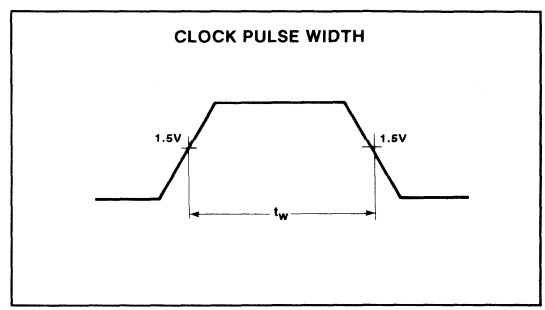
TEST LOAD CIRCUIT (THREE-STATE OUTPUTS)



NOTE: C_L includes fixture capacitance.

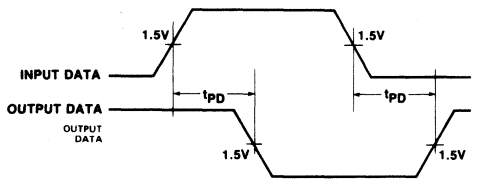
VOLTAGE WAVEFORMS



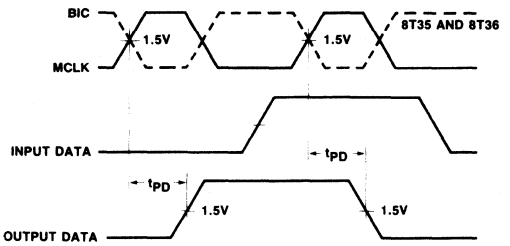


VOLTAGE WAVEFORMS (Cont'd)

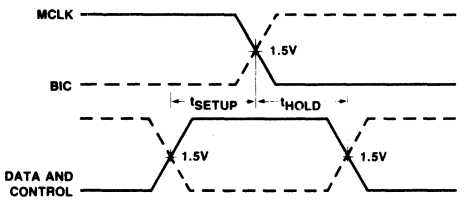
DATA DELAY TIMES
Input Data Reference



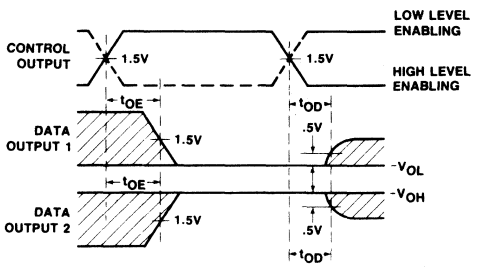
DATA DELAY TIMES
Clock Referenced



SETUP AND HOLD TIMES



OUTPUT ENABLE AND DISABLE TIMES
(Three-State Outputs)



Waveform #1 is for an output with internal conditions such that the output is Low when the three-state driver is enabled. Waveform #2 is for the opposite condition.

ADDRESS PROGRAMMING

The I/O Port is manufactured such that an address of all high levels (>2V) on the Microprocessor Bus inputs matches the Port's internal address. To program a bit so a low-level input (<0.8V) matches, the following procedure should be used:

1. Set all control inputs to their inactive state ($\overline{BIC}$ = $\overline{BOC}$ = $\overline{ME}$ = V_{CC}, SC = WC = MCLK = GND). Leave all Microprocessor Bus I/O pins open.
2. Raise V_{CC} to 7.75V ± .25V.
3. After V_{CC} has stabilized, apply a single programming pulse to the user data bus bit where a low-level match is desired. The voltage should be limited to 18V; the current should be limited 75mA. Apply the pulse as shown in Figure 1.
4. Return V_{CC} to 0V. (Note 1).
5. Repeat this procedure for each bit where a low-level match is desired.
6. Verify that the proper address is programmed by setting the Port's status latch ($\overline{IV0}$-$\overline{IV7}$ = desired address, $\overline{ME}$ = WC = L, SC = MCLK = H). If the proper address has been programmed, data presented at the μP bus will appear inverted on the user bus outputs. (Use normal V_{CC} and input voltage for verification.)

After the desired address has been programmed, a second procedure must be followed to isolate the address circuitry. The procedure is:

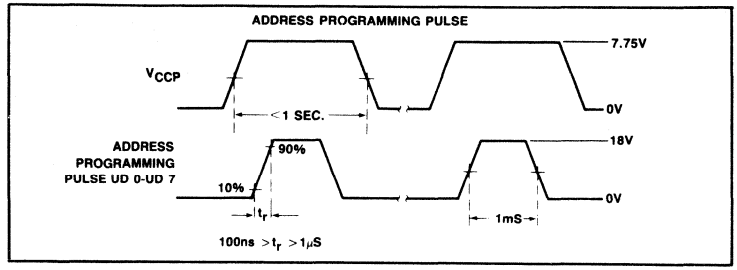


Figure 1

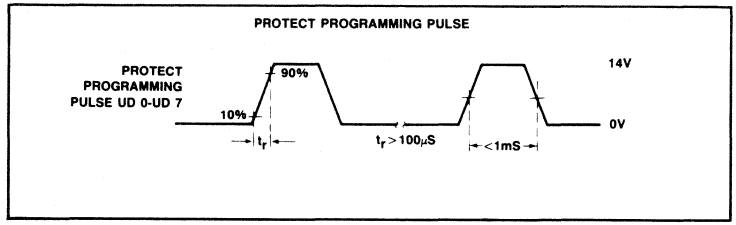


Figure 2

1. Set V_{CC} and all control inputs to 0V. (V_{CC} = $\overline{BIC}$ = $\overline{BOC}$ = $\overline{ME}$ = SC = WC = MCLK = 0V). Leave all Microprocessor Bus I/O pins open.
2. Apply a protect programming pulse to every user data bus pin, one at a time. The voltage should be limited to 14V; the current should be limited to 150mA. Apply the pulse as shown in Figure 2.
3. Verify that the address circuitry is isolated by applying 7V to each user data bus pin and measuring less than 1mA of input current. The conditions should be the same as in step 1 above. The rise time on the verification voltage must be slower than 100μs.

PROGRAMMING SPECIFICATIONS[1]

PARAMETER		TEST CONDITIONS	LIMITS			UNITS
			Min	Typ	Max	
V_{CCP}	Programming supply voltage					
	Address		7.5		8.0	V
	Protect			0		V
I_{CCP}	Programming supply current	V_{CCP} = 8.0V			250	mA
	Max time V_{CCP}>5.25V				1.0	s
	Programming voltage					
	Address		17.5		18.5	V
	Protect		13.5		14.0	V
	Programming current					
	Address				75	mA
	Protect				150	mA
	Programming pulse rise time					
	Address		.1		1	μs
	Protect		100			μs
	Programming pulse width		.5		1	ms

NOTE

1. If all programming can be done in less than 1 second, V_{CC} may remain at 7.75V for the entire programming cycle.

APPLICATIONS

Figure 3 shows some of the various ways to use the I/O Port in a system. By controlling the $\overline{BIC}$ and $\overline{BOC}$ lines, the device may be used for the input and output of data, control, and status signals. I/O Port 1 functions bidirectionally for data transfer and I/O Port 2 provides a similar function for discrete status and control lines. I/O Ports 3 and 4 serve as dedicated output and input ports, respectively.

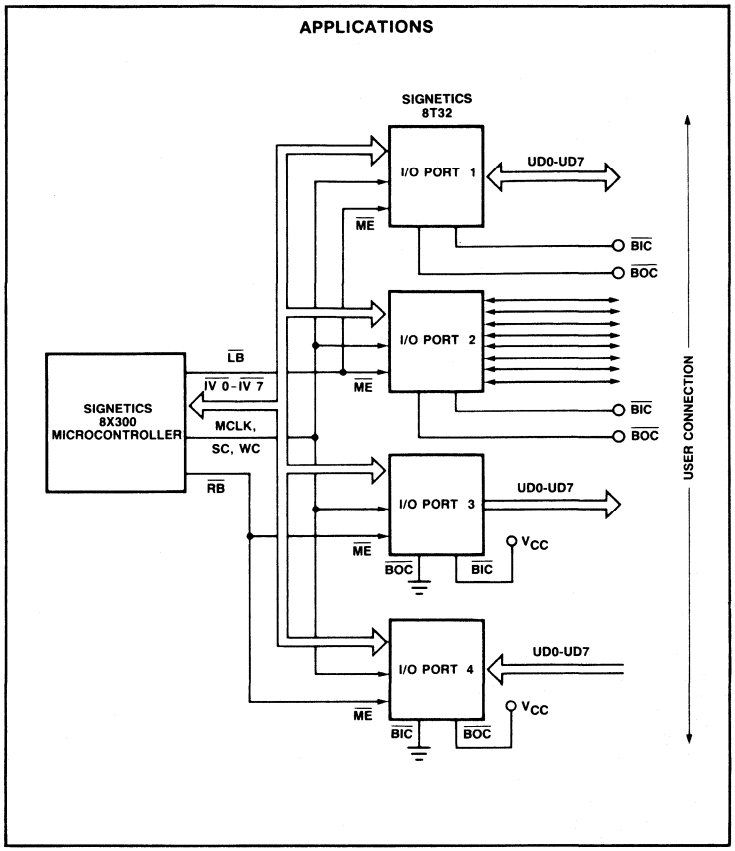


Figure 3

Product support

PRODUCT SUPPORT

Originally published by Signetics January 1984

SUPPORT FACILITIES

The 8X300 Family is strongly supported with Development Systems, Support Software, Applications, Training and Documentation. Together, this support provides the user with a powerful set of tools to evaluate, design, debug, and implement a simple or complex system.

DEVELOPMENT SYSTEMS

EZ-PRO (Manufactured by American Automation)
- Universal Development System
- Relocating macroassembler
- Full speed in-circuit emulator/debugger
- Maximum memory support

8X305 ICEPACK (Manufactured by Sigen Corp.)
- Full speed in-circuit emulator/debugger
- RS-232 interface to CP/M or Intellec
- 4K word memory/8-bit extended microcode
- Low cost

Signetics 8X305 Prototyping System
- Single board module
- RS-232 interface
- Resident monitor
- 256 to 4096 words of Writable Control Storage
- Minimum cost

SOFTWARE

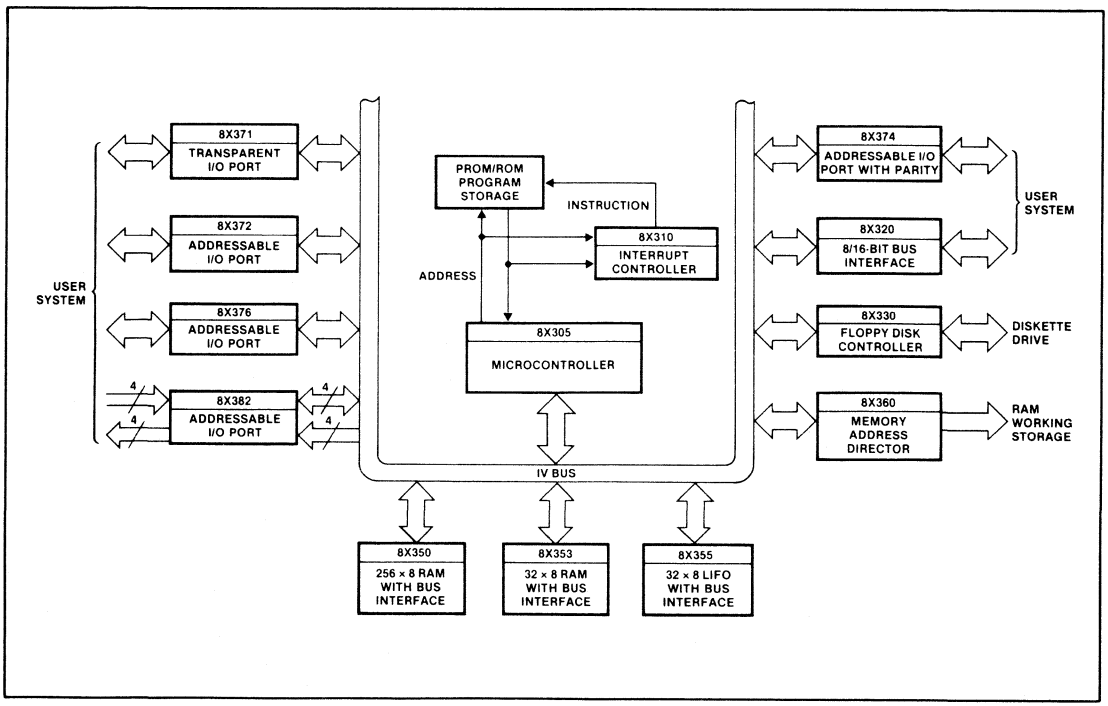
MCCAP CrossAssembler
- Full function macroassembler
  - — Free format source code
  - — Symbolic address assignment
  - — Nested macro support
  - — Cross-reference table
  - — Supports extended microcode
- Multiple output formats
- Available in two versions
  - — FORTRAN source code
  - — Intellec/ISIS object code

APPLICATIONS SUPPORT

Field applications engineers
Product applications engineers
Application notes
- Floppy disk controller
- ECC
- Hard disk controller
- Local network interfacing

TRAINING AND DOCUMENTATION

- Videocassette training course
- Designer's seminar
- 8X305 User's Manual
- 8X300 Family Product Capabilities Manual
- MCCAP Programming Manual
- Full complement of Data Sheets



PREFACE

Originally published by Signetics January 1984

8X305 Prototyping System

This document provides the information required to understand, set up and operate the 8X305 Prototyping System. It is recommended that the user become thoroughly familiar with the 8X305 MicroController and the high speed peripherals that support the device. For this purpose, the following documents are recommended:

- 8X305 Users Manual — comprehensive functional detail, interface characteristics, hardware and software design data, and systems information for the 8X305 MicroController.
- 8X300 Family Product Capabilities Manual — overview of the 8X300 Family of parts, including application information on the 8X305 MicroController and its support devices.
- MCCAP Manual — complete description of the powerful MicroController Cross-Assembler Program for the 8X300 and 8X305.
- Data Sheets — electrical and functional characteristics for each member of the 8X300 Family and related parts.
  - 8X305 MicroController
  - 8X310 Interrupt Control Coprocessor
  - 8X320 Bus Interface Register Array
  - 8X330 Floppy Disk Formatter/Controller
  - 8X338 Local Area Network Controller
  - 8X350 256 Byte Bipolar RAM
  - 8X360 Memory Address Director
  - 8X371 Latched 8-bit Bidirectional I/O Port
  - 8X372/8X376 Addressable 8-bit Bidirectional I/O Port
  - 8X374 Addressable 8-bit Bidirectional I/O Port with Parity
  - 8X382 Addressable 4-IN/4-OUT I/O Port
  - 8X60 FIFO RAM Controller

INTRODUCTION

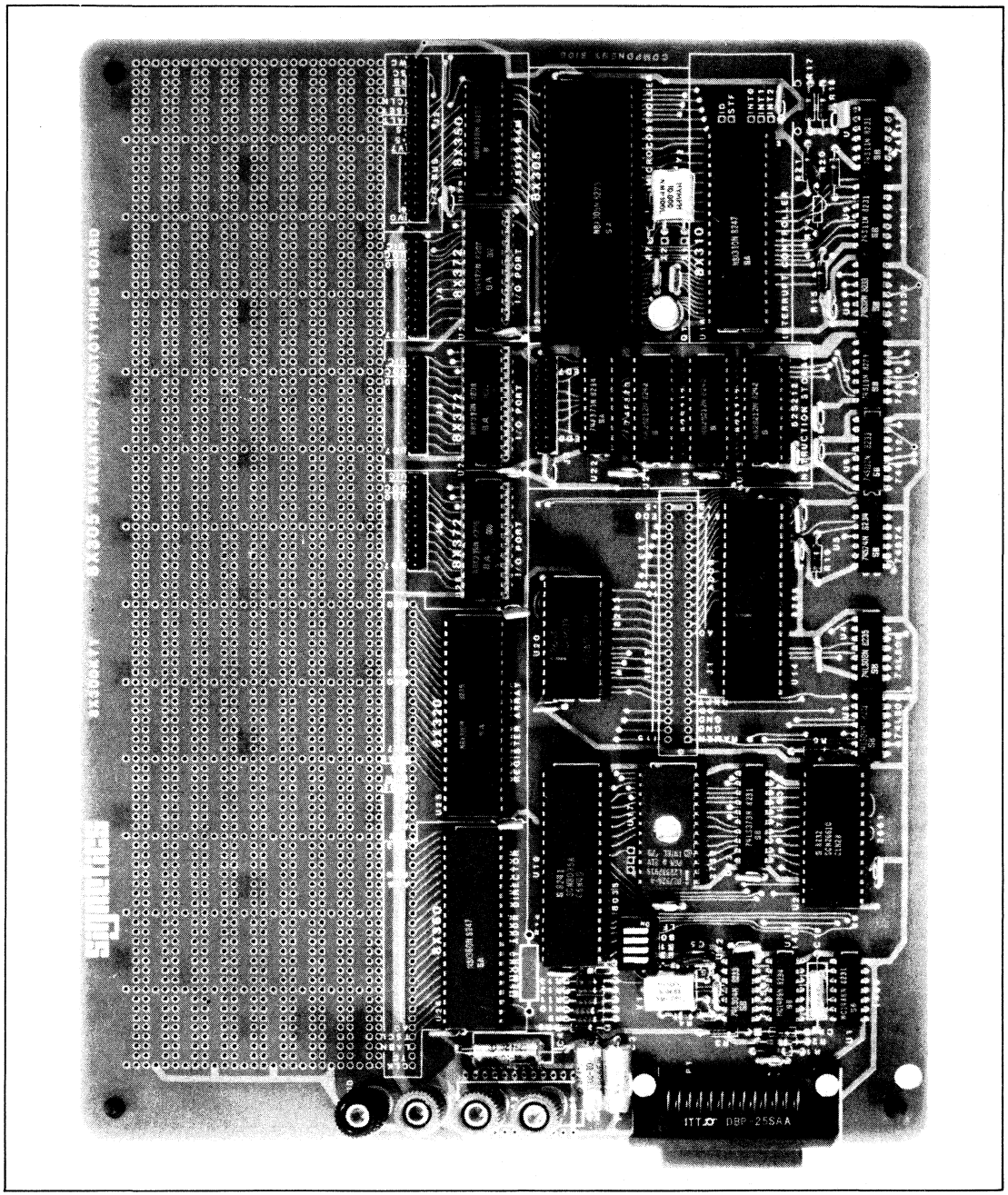


Figure 1-1. 8X305 Prototyping System

1.1 SYSTEM DESCRIPTION

The 8X305 Prototyping System is a powerful design support tool that aids the engineer in the evaluation, design, and prototyping of systems based on the Signetics 8X305 MicroController and its family of support devices. Its advanced features permit the development of both 8X305 firmware and application circuitry. The prototyping systems capabilities are adequate to serve as a complete development system for simple systems and provide a low-cost tool for evaluating portions of more complex designs.

1.2 ARCHITECTURAL OVERVIEW

As shown in Figure 1-1, the 8X305 Prototyping System consists of a single printed circuit board that includes an 8X305 MicroController and various 8X305 Family peripheral devices. The 8X305's microprogram resides in Writeable Control Storage (WCS). A Monitor Processor controls operation of the 8X305 by loading the WCS, activating the Run/ Step logic, and directly placing instructions onto the 8X305's Instruction bus. The Monitor Processor also controls the User Interface, which is through a standard RS-232 connector. The remainder of the board is occupied by power connections and a large wire-wrap area for prototyping of user-developed circuits. A complete discussion of the operation and interrelationship of these functions is contained in later chapters.

1.2.1 USER INTERFACE

The User Interface of the 8X305 Prototyping System is accomplished through a standard RS-232 connector. Data rates from 110 to 19,200 baud are switch selected by the user. The monitor program contained in the system controls all user communication, which is accomplished interactively through a straightforward and user-friendly syntax. While the operation of the system requires only a low-cost "dumb" terminal, it can be connected to a host computer to support more advanced developments. Commands are included to support up and down loading of programs in such applications.

1.2.2 MONITOR PROCESSOR AND RUN/STEP LOGIC

Operation of the system is controlled by the Monitor Processor, which is implemented using an 8035 Microprocessor. The Monitor Processor is responsible for the following functions:

- User interaction
- Loading Writeable Control Storage
- Loading and reading 8X305 registers and I/O devices
- Control of Run/Step logic

Programming for the Monitor Processor is supplied by Signetics and is contained in a PROM.

1.2.3 WRITEABLE CONTROL STORAGE

8X305 MicroController programs are executed from a Writeable Control Storage (WCS) that is contained on the board. The prototyping system is supplied with 256 words of instruction memory. An expansion module is available to support address space requirements of up to 4096 words. The WCS is sufficiently fast to permit full speed operation of the 8X305.

Writeable Control Storage words are 25 bits wide to support advanced microprogramming requirements. Sixteen of these bits contain actual 8X305 instructions. Eight of the remaining bits are used to support "Extended Microcode" designs as described in the 8X305 Users Manual. The 25th bit, transparent to the user, is set by the Monitor Processor to control breakpoints.

Since the three-bus architecture of the 8X305 does not permit the MicroController to modify its own program memory, the WCS is loaded by the Monitor Processor.

1.2.4 8X305 MICROCONTROLLER AND PERIPHERALS

An 8X305 MicroController and various 8X300 Family peripheral devices are included in the prototyping system.

The Instruction and Program Address busses of the 8X305 are connected to Writeable Control Storage (WCS) as well as an 8X310 Interrupt Control Coprocessor (ICC). The interrupt and status pins of the 8X310 are available to the user for use in prototyping real-time or other interrupt driven systems.

The 8X305's IV bus is connected to the following 8X300 Family peripheral devices:

(1) 8X320 Bus Interface Register Array
(1) 8X350 256 Byte Bipolar RAM
(1) 8X360 Memory Address Director
(3) 8X372 Addressable 8-bit Bidirectional I/O Ports

IV bus data and control signal connections are also available to the user to permit attachment of other devices or user developed logic. User interface connections to the 8X320, 8X360, and 8X372's are available adjacent to the wire-wrap area to permit prototyping of various 8X305 based designs.

SYSTEM SETUP

2.1 POWER CONNECTIONS

> *CAUTION*
>
> *Power connections must be properly made; otherwise, component damage will result.*

Power connections to the Prototyping System are made through four binding posts located on the left hand edge of the board. These are labeled GROUND, +5, —12, and +12. Without options or user circuitry in the wire-wrap area, the current drawn from each power source is as follows:

+5 VDC — less than 2.5 amperes
+12 VDC — less than 20 milliamperes
—12 VDC — less than 20 milliamperes

Additional current must be supplied for any options or user circuitry added to the board. At the user's option, a DC-to-DC converter (converts +5 VDC to ±12 VDC) can be installed in the space allotted to the —12 VDC and +12 VDC binding posts. Refer to the parts list in Appendix B for the manufacturer and part number of the recommended device.

2.2 RS-232 INTERFACE

An RS-232 connector is provided in the lower left-hand corner of the system for interconnection to the user's CRT terminal, and is connected as shown in Table 2-1.

Table 2-1 RS-232 CONNECTOR

Pin No.	Signal	Description
1	GND	Ground
2	TXD	Transmit Data(in)
3	RXD	Receive Data(out)
4	RTS	Request to Send(in)
5	CTS	Clear to Send(out)
6	DSR	Data Set Ready(out)
7	GND	Ground
8	CAD	Carrier Detect(out)

RS-232 specifications call for data communications equipment (DCE), such as this system, to be connected to data terminal equipment (DTE), such as the user's CRT terminal. When connecting this system to another DCE, such as a host computer, be sure to interchange TXD (pin 2) with RXD (pin 3), and RTS (pin 4) with CTS (pin 5). This can be done on the cable or it can be accomplished logically by using a Null Modem. Due to the variety of interpretations of RS-232, some terminals may not immediately work with the Prototyping System. If difficulties are encountered, first reduce the connections to three signals: TXD, RXD, and GND. Then if necessary, interchange TXD and RXD signals.

2.3 BAUD RATE SELECTION

Any of the eight baud rates listed in Table 2-2 may be selected by proper setting of switches B2, B1 and B0 located near the RS-232 connector.

Table 2-2 BAUD RATE

B2	B1	B0	Baud Rate
0	0	0	19200
0	0	1	9600
0	1	0	4800
0	1	1	2400
1	0	0	1200
1	0	1	600
1	1	0	300
1	1	1	110

(0 = Switch off, 1 = Switch on)

For hard-copy printing terminals, an optional line-feed feature may be selected to avoid over-strike of characters after a backspace on error. This feature is selected by setting the LF switch located next to switch B0 to the ON position.

2.4 EXTERNAL OSCILLATOR CONNECTIONS

> *CAUTION*
>
> *To prevent possible damage to the crystal, never apply an external oscillator signal to X1 or X2 input with crystal Y2 connected to the circuit.*

The Prototyping System is supplied with a 10 MHz crystal; therefore it operates the 8X305 MicroController at its full rated speed of 200 nanoseconds per instruction. The crystal may be changed by the user to any frequency from 4 MHz to 10 MHz. Alternatively, an external oscillator may be connected to the X1 and X2 inputs of the 8X305, as described in the 8X305 Users Manual. The external oscillator may operate at any frequency between 0.2 MHz and 10 MHz. Note that the 8X305 is capable of running at frequencies lower than 0.2 MHz, but the Prototyping System's Monitor Processor expects the 8X305 to be finished executing an instruction within 10 microseconds, thereby imposing a lower limit of 0.2 MHz. Tie points for X1 and X2 are located next to crystal Y2 and the 8X305. Be sure to disconnect crystal Y2 before connecting the external oscillator inputs to X1 and X2.

2.5 INHIBIT JUMPER FOR 8X310 ROM DISABLE

The 8X310 Interrupt Control Coprocessor connects to the Instruction and Address buses of the 8X305. It accomplishes interrupt and subroutine control by disabling the control storage that contains the 8X305's microprogram and placing specific JMP instructions onto the instruction bus. To permit the Prototyping System to operate either with or without an 8X310 in the circuit, a jumper is incorporated into the ROM Disable (RD) circuitry.

When the 8X310 Interrupt Control Coprocessor is physically present in location U16, the ROM Disable Inhibit Jumper located at R14 next to the 8X310 must not be present so that the 8X310 can disable WCS and avoid bus contention problems. When the 8X310 is not present, this jumper must be connected to the board at location R14 to permanently enable the Writeable Control Store RAMs.

SYSTEM OPERATION

3.1 POWER UP AND DIAGNOSTICS

When power is applied to the Prototyping System, resident diagnostic programs are executed to test the Micro-Controller and Writeable Control Store (WCS). The following message is then printed:

```
8X305 PROTOTYPING SYSTEM
REV n
SYSTEM CHECK ON
*
```

where "n" equals the current revision level.

If the Prototyping System is functioning improperly, either "MEMORY ERROR" or "8X305 ERROR" messages will be printed. Then the "prompt" character (*) will be printed and the user may examine WCS memory or 8X305 functions to diagnose the problem.

3.2 MONITOR PROGRAM

With the printing of the "prompt" character (*), system control passes to the monitor program. The user may then enter any of the ten monitor commands:

I **INPUT 8X305 instructions into memory:**
Accepts a starting WCS memory address and then permits the entering of an 8X305 instruction in mnemonic form and an extension instruction in octal notation.

n **Register examine/change, where "n" = register number:**
Displays the register number and its contents and allows a new value to be substituted.

(R0 may be accessed by its alternate name AUX by entering "A" or, R10 may be accessed by its alternate name OVF by entering "0".)

G **GO execute user's 8X305 program:**
Accepts a starting memory address and executes at full speed until a breakpoint or keyboard entry is reached. Then the contents of the registers and the next executable instruction are displayed, and single stepping can proceed.

S **STEP, single step user's program:**
Accepts a starting memory address and displays the contents of the registers and the instruction at that memory address. The instruction is executed by hitting the space bar; and successive instructions by successively hitting the space bar.

M **MEMORY and breakpoint examine/change:**
Accepts a starting WCS memory address and then displays the contents of that location in mnemonic form and indicates a possible breakpoint by an exclamation point. A breakpoint at this location may be set by typing an exclamation point or cleared by typing a backspace. A new 8X305 instruction and extension instruction is then entered by typing an I, as with the INPUT command.

L **LB, left-bank examine/change:**
Accepts a bank address (or the currently enabled address, if a blank is entered) and displays the contents of that left-bank address (0-377 octal). A new value may be entered to substitute for the original content.

R **RB, right-bank examine/change:**
Operates the same as the LB command except action is upon the right-bank.

D **DUMP memory contents to terminal:**
Accepts a starting WCS memory address and an ending address, and then dumps the memory contents from the start address to the end address onto the RS-232 port in ASCII HEX QUOTE format as described in Section 3.6.

F **FILL memory contents from terminal:**
Accepts a starting WCS memory address and fills memory in ASCII HEX QUOTE format as described in Section 3.6.

X **XCODE, temporarily set extended microcode latch:**
Accepts a new extension code value to be temporarily set in the extended microcode latch for control of user circuitry. The content of the extended microcode section of WCS is not changed by this command.

Although the user need only enter a single letter command, the monitor will respond by typing the whole command name as indicated in capitals above.

Any of the commands may be aborted before completion by typing an ASCII "control-C" character. While entering any number (sequence of octal digits), corrections may be made by entering a backspace character and then entering the correct number.

3.3 COMMAND SYNTAX DIAGRAMS

System commands and monitor responses are defined in the syntax diagrams in Figure 3-1a, b and c.

3.4 SAMPLE USAGE

The sample program in Figure 3-3 is shown to give the user an idea of a typical session and includes most commands used by the Prototyping System. A short program is input to WCS via the terminal Keyboard that continually increments R6 of the 8X305 and writes each new (incremented) value to location 133 of the 8X350 on the Right-Bank of the IV Bus and to I/O Port 001 on the Left-Bank. Since extended microcode is not needed in this example, none was entered as indicated by "/000".

3.5 BREAKPOINTS

Breakpoints are designed to halt the 8X305 just prior to the execution of an instruction on which a breakpoint has been specified. If the GO command is issued to start at an address that has a breakpoint set, the system will not stop on that address immediately. If the 8X305 should access that address again then a breakpoint stop will occur.

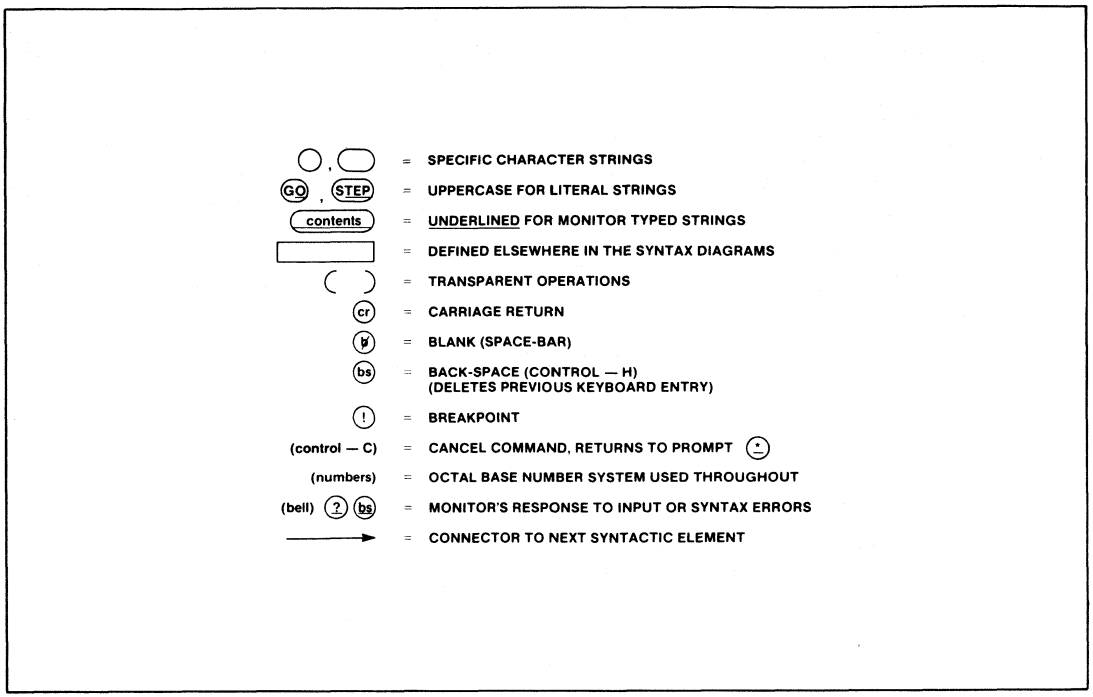


Figure 3-1a. Syntax Definitions

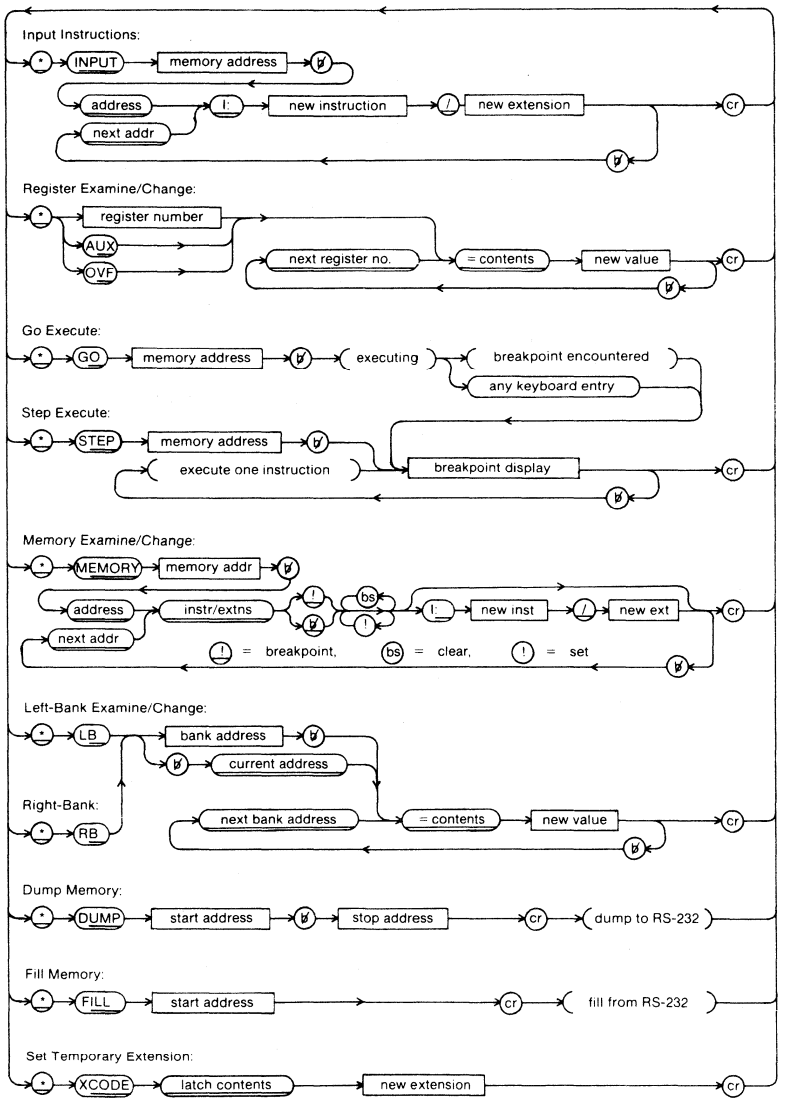
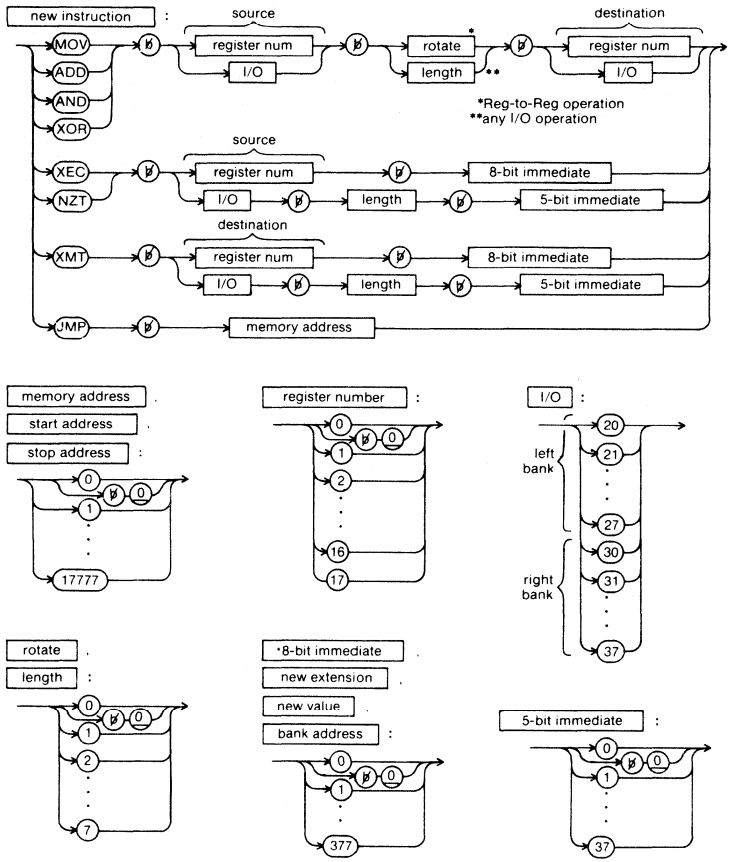


Figure 3-1b. Syntax Flow



note: — First two lines are column headers, printed each breakpoint or each 20 single steps.
— Third line is register contents.
— Fourth line is breakpoint: address — instruction/extension. This is the next instruction to be executed in single step.

Figure 3-1c. Syntax Flow

```
*
* DUMP 00000 00014
DUMPING HIGH BYTES ....
(STX)
00'C0'24'B9'C0'AC'68'74'01'10'70'11'F1'
(ETX)
DUMPING LOW BYTES ....
(STX)
00'00'08'BF'10'FF'FF'96'06'FF'7D'40'D7'
(ETX)
DUMPING EXT BYTES ....
(STX)
00'00'00'00'00'50'00'00'00'00'00'00'00'
(ETX)
*
*
```

Figure 3-2. Example of ASCII HEX QUOTE Format

3.6 ASCII HEX QUOTE FORMAT

Memory contents to and from the terminal during the DUMP and FILL commands are in an object code format commonly supported by PROM programming hardware known as "ASCII HEX QUOTE" format. Each block of eight-bit wide data is preceeded by an STX (ASCII Start of Text) character. Then each 8-bit byte is represented by 2 ASCII HEX characters (0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F) followed by a single quote ('). As many bytes as necessary may be transmitted until an ETX (ASCII End of Text) character terminates the block. Three such blocks are transmitted for any specified memory range: high byte of 8X305 instruction, low byte of 8X305 instruction, and extended microcode byte. Figure 3-2 shows an example of ASCII HEX QUOTE format.

3.7 XEC (EXECUTE) INSTRUCTIONS

When stepping through a user program and an 8X305 XEC instruction is encountered, the system will display an "XR" after the next instruction to indicate an XEC range of address.

Note that stepping must start on or before an XEC instruction for the program flow to be correct. If while running a program, a breakpoint or stop occurs on an instruction that is the target of an XEC instruction, an "XR" will not be displayed and program flow will not be correct if single stepping is then begun.

It is valid to nest any number of XEC instructions. Single stepping will work properly provided that the user start on or before the first XEC instruction.

3.8 8X310 INTERRUPT CONTROL COPROCESSOR CONSIDERATIONS

Certain 8X305 "NOP" instructions have specific meanings to the 8X310 as indicated in the data sheet. When using an 8X310 in the system, the user must start running or stepping from an instruction that is not an 8X310 instruction. It is valid, however, to start running and encounter a breakpoint or stop on an 8X310 instruction and then continue single stepping the program.

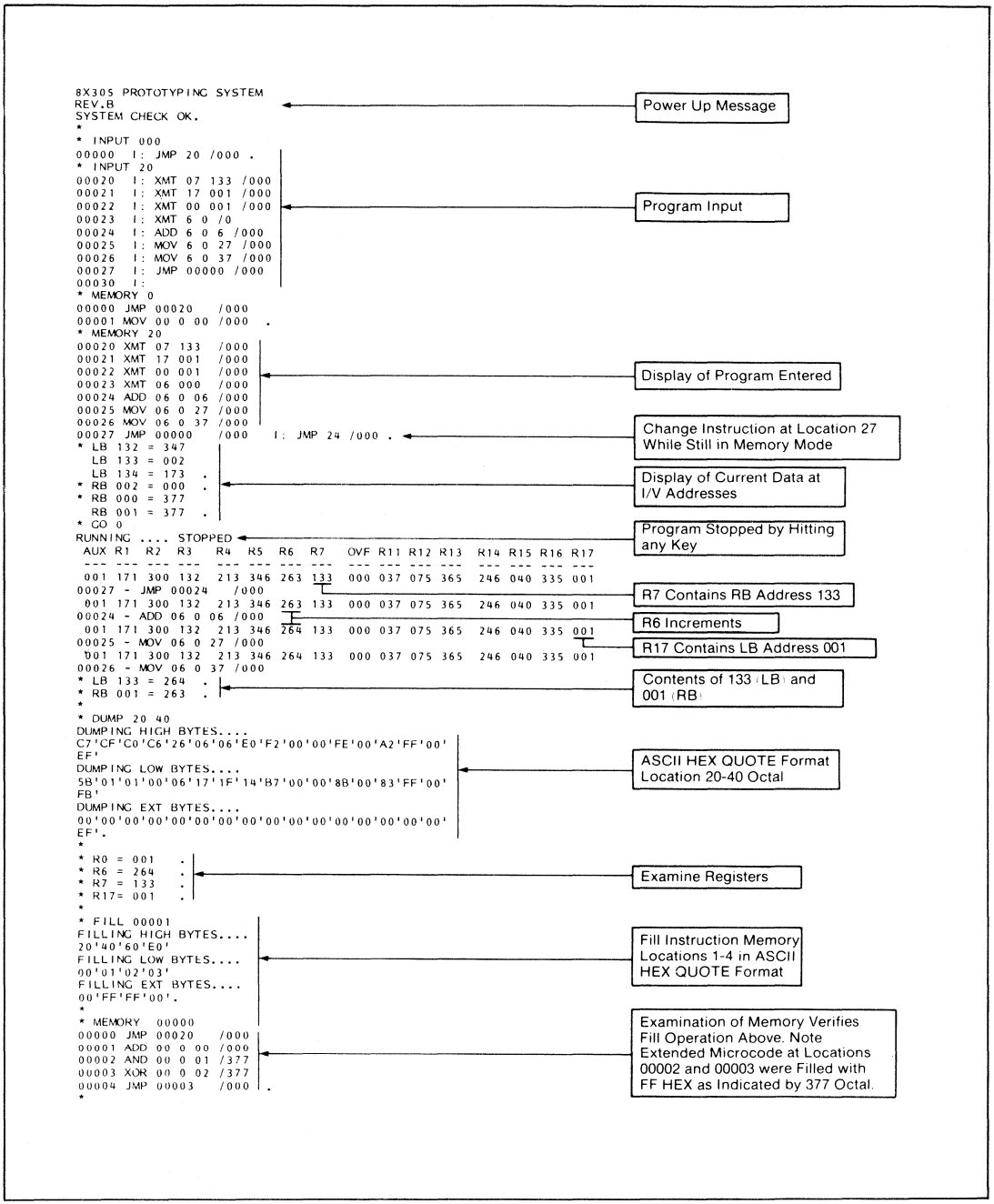


Figure 3-3. Sample Usage

USER CONNECTIONS

4.1 WIRE-WRAP AREA

The wire-wrap area located at the top of the board provides 26 square inches for user prototyping. This space accommodates standard IC widths from 0.3 to 0.9 inches and also provides power and ground connections.

4.2 IV BUS CONNECTIONS

The IV bus is the major communications link between the 8X305 MicroController and its family of peripherals. The bus is logically partitioned into two banks, referred to as the Left-Bank and Right-Bank. In the Prototyping System the 8X350 Working Storage RAM is connected to the Left-Bank, and the 8X320 Register Array, 8X360 Memory Address Director, and 8X372 I/O Ports are connected to the Right-Bank. All IV bus signals are present at connector J2 as shown in Table 4-1.

Table 4-1 IV BUS CONNECTOR J2

Pin No.	Signal	Description
1	$\overline{IV0}$	BUS (MSB)
3	$\overline{IV1}$	BUS
5	$\overline{IV2}$	BUS
7	$\overline{IV3}$	BUS
9	$\overline{IV4}$	BUS
11	$\overline{IV5}$	BUS
13	$\overline{IV6}$	BUS
15	$\overline{IV7}$	BUS(LSB)
17	V_{CC}	+5 V
19	V_{CC}	+5 V
21	$\overline{HALT}$	Halt
23	$\overline{RESET}$	Reset
25	MCLK	Clock
27	$\overline{LB}$	Left-Bank
29	$\overline{RB}$	Right-Bank
31	SC	Select Control
33	WC	Write Control
(All even pins)	GND	Ground

4.3 I/O PORTS

These 8X372 I/O ports have been provided on the Right-Bank of the IV bus for latching of input or output data. The ports are programmed for addresses 000, 001 and 002 and all signals are available at connectors J4, J5 and J6 respectively. Signals on these connectors are described in Table 4-2. Note that I/O port compatibility allows the user to substitute an 8X376 or 8X382 for any one of the 8X372 I/O ports. (One 8X371 non-addressable I/O port may be substituted, but ONLY if all other components on that bank are removed.)

Table 4-2 I/O PORT CONNECTORS J4, J5, AND J6

Pin No.	Signal	Description
2	UD7	User Data (LSB)
4	UD6	User Data
6	UD5	User Data
8	UD4	User Data
10	UD3	User Data
12	UD2	User Data
14	UD1	User Data
16	UD0	User Data (MSB)
18	$\overline{UOC}$	Output Control
20	UIC	Input Control
(All odd pins)	GND	Ground

4.4 EXTENDED MICROCODE CAPABILITY

"Extended Microcode" is a technique commonly used in 8X305 MicroController based designs to optimize performance. It is implemented by designing program memory to be wider than the 16-bit instruction word required for the 8X305. The additional bits are referred to as the extension and can be used for fast I/O selection or other system control and status monitoring purposes.

The Prototyping System uses a 24-bit instruction word, thus providing facilities for 8 bits of extended microcode. These bits are accessible to the user at connector J3 as shown in Table 4-3.

Table 4-3 EXTENDED MICROCODE BITS AT CONNECTOR J3

Pin No.	Signal	Description
1		No Connection
3	ED0	Extended Microcode (MSB)
5	ED1	Extended Microcode
7	ED2	Extended Microcode
9	ED3	Extended Microcode
11	ED4	Extended Microcode
13	ED5	Extended Microcode
15	ED6	Extended Microcode
17	ED7	Extended Microcode (LSB)
19		No Connection
(All even pins)	GND	Ground

Table 4-4 8X320 SIGNAL CONNECTIONS

Signal	Description
B/$\overline{W}$	Byte/Word Control
A0	Primary Port Address (LSB)
A1	Primary Port Address
A2	Primary Port Address
A3	Primary Port Address (MSB)
$\overline{PIOE}$	Programmed I/O Enable
R/$\overline{W}$	Read/Write Control
WS	Write Strobe
$\overline{DMAE}$	Direct Mem. Access Enable
D7A	Primary Data Port (LSB)
D6A	Primary Data Port
D5A	Primary Data Port
D4A	Primary Data Port
D3A	Primary Data Port
D2A	Primary Data Port
D1A	Primary Data Port
D0A	Primary Data Port
D7B	Primary Data Port
D6B	Primary Data Port
D5B	Primary Data Port
D4B	Primary Data Port
D3B	Primary Data Port
D2B	Primary Data Port
D1B	Primary Data Port
D0B	Primary Data Port (MSB)

4.5 8X310 CONNECTIONS

The 8X310 is connected to the 8X305 MicroController and the Writeable Control Store RAM to provide interrupt and subroutine capability. Five additional signals are provided for user interface as described in the 8X310 data sheet. These signals are accessible at tie points just to the right of the 8X310 chip:

STF	Stack Full Status
ID	Interrupt Disable Control
INT0	Interrupt 0 Input
INT1	Interrupt 1 Input
INT2	Interrupt 2 Input

4.6 8X320 CONNECTIONS

An 8X320 Bus Interface Register Array has ben provided on the IV Bus as a Right-Bank I/O device for interfacing to the user's system. The primary data, status and command signals are accessible at tie points located between the 8X320 and the wire-wrap area. A list of these signals is provided in Table 4-4.

4.7 8X360 CONNECTIONS

The 8X360 Memory Address Director has been incorporated into the Prototyping System design to facilitate implementation of a DMA channel. It is connected to the IV Bus as a Right-Bank I/O device. Interconnection to the signals listed in Table 4-5 can be made at the tie points located between the 8X360 and the wire-wrap area.

In applications using extended microcode to enable I/O devices care must be taken to avoid IV Bus contention with 8X300 Family peripheral devices enabled through the more commonly used address — select cycle. Refer to the 8X305 Users Manual for more information on extended microcode operations.

4.8 MEMORY EXPANSION

The Prototyping System is provided with 256 24-bit words of Writeable Control Storage; 16 bits for 8X305 instructions and 8 bits for extended microcode. The depth of Control Storage can be increased by the connection of an expansion module to connector J1, as shown in Table 4-6. A 4096 word Writeable Control Storage Expansion Module is available from Signetics (Part Number 8X300KT2SK). A schematic for this expansion module is provided in Appendix D.

Table 4-5 8X360 SIGNAL CONNECTIONS

Signal	Description
CLK	Clock Input
TC	Terminal Count Status
LABN	Loop Abort Control
TSCL	Tri-state Control
A0	Address Output (LSB)
A1	Address Output
A2	Address Output
A3	Address Output
A4	Address Output
A5	Address Output
A6	Address Output
A7	Address Output
A8	Address Output
A9	Address Output
A10	Address Output
A11	Address Output
A12	Address Output
A13	Address Output
A14	Address Output
A15	Address Output (MSB)
RS0	Register Select (LSB)
RS1	Register Select
RS2	Register Select
RS3	Register Select (MSB)

Table 4-6 MEMORY EXPANSION CONNECTOR J1

Pin No.	Signal	Description
1	RAMEN	Disables on-board RAM
3	GND	Ground
5	GND	Ground
7	OD	Output Disable
9	BKPT	Breakpoint
11	$\overline{W}$	Write
13	—	No Connection
15	—	No Connection
17	—	No Connection
19	—	No Connection
21	CE1	Chip Enable
23	A0	8X305 Address (MSB)
25	A1	8X305 Address
27	A2	8X305 Address
29	A3	8X305 Address
31	A4	8X305 Address
33	A5	8X305 Address
35	A6	8X305 Address
37	A7	8X305 Address
39	A8	8X305 Address
41	A9	8X305 Address
43	A10	8X305 Address
45	A11	8X305 Address
47	A12	8X305 Address (LSB)
49	V_{CC}	+5 V

Pin No.	Signal	Description
2	I15	8X305 Instruction (LSB)
4	I14	8X305 Instruction
6	I13	8X305 Instruction
8	I12	8X305 Instruction
10	I11	8X305 Instruction
12	I10	8X305 Instruction
14	I9	8X305 Instruction
16	I8	8X305 Instruction
18	I7	8X305 Instruction
20	I6	8X305 Instruction
22	I5	8X305 Instruction
24	I4	8X305 Instruction
26	I3	8X305 Instruction
28	I2	8X305 Instruction
30	I1	8X305 Instruction
32	I0	8X305 Instruction (MSB)
34	E7	Extended Microcode (LSB)
36	E6	Extended Microcode
38	E5	Extended Microcode
40	E4	Extended Microcode
42	E3	Extended Microcode
44	E2	Extended Microcode
46	E1	Extended Microcode
48	E0	Extended Microcode (MSB)
50	V_{CC}	+5 V

THEORY OF OPERATION

As can be noted with the aid of the block diagram in Figure 5-1, the 8X305 prototyping system board contains circuits which may be categorized as follows:

1. Monitor Processor
2. 8X305 MicroController and Family
3. Writeable Control Store
4. Run/Step Logic

5.1 FUNCTIONS OF THE MONITOR PROCESSOR

The Monitor Processor, an 8035 microprocessor and peripherals, controls all the commands and operations described in Chapter 3. The Monitor Processor handles all communication with the terminal as well as reading and writing the 8X305's program storage, registers, and I/O port contents.

The 8X305 can execute instructions from Writeable Control Storage or an instruction that is latched into the 8243 by the Monitor Processor. Typically the instruction in the 8243 will store or read an 8X305's register contents, I/O Port contents, or set the address in the 8X305 Program Counter.

The 8243 is also used to read and write the contents of Writeable Control Store. Since the 8X305 does not have an address bus that can be three-stated and because a buffer would increase the memory access time, to read a specific memory location a JMP is "forced" upon the 8X305 by way of the 8243 to set the address lines.

The Monitor Processor reads the register contents of the 8X305 by forcing an XEC Rn, 000, where Rn is the desired register. This causes the register contents to be placed on the lower eight address lines of the 8X305 where it may be read by the Monitor Processor and sent out on the RS-232 interface. To store a value into the 8X305, the Monitor Processor will force a XMT Rn, XXX, where XXX is the desired register contents. (For R12 and R13, this will be accomplished by a XMT followed by a MOV.)

5.2 8X305 FAMILY

With the following two exceptions the 8X305 MicroController and its supporting peripherals connect to the prototyping system in a conventional manner:

1. Rather than a direct tie to the MCLK output of the 8X305, the MCLK input to the 8X310 Interrupt Control Coprocessor is gated. The gating circuits are required to implement correct single-step operation of the system.
2. The $\overline{\text{HALT}}$ and $\overline{\text{RESET}}$ inputs to the 8X305 are gated. Connected in this manner, the $\overline{\text{HALT}}$ and $\overline{\text{RESET}}$ signals will only affect the MicroController in the run mode. User circuits requiring either or both of these inputs should pick up the signals via the IV Bus connector J2.

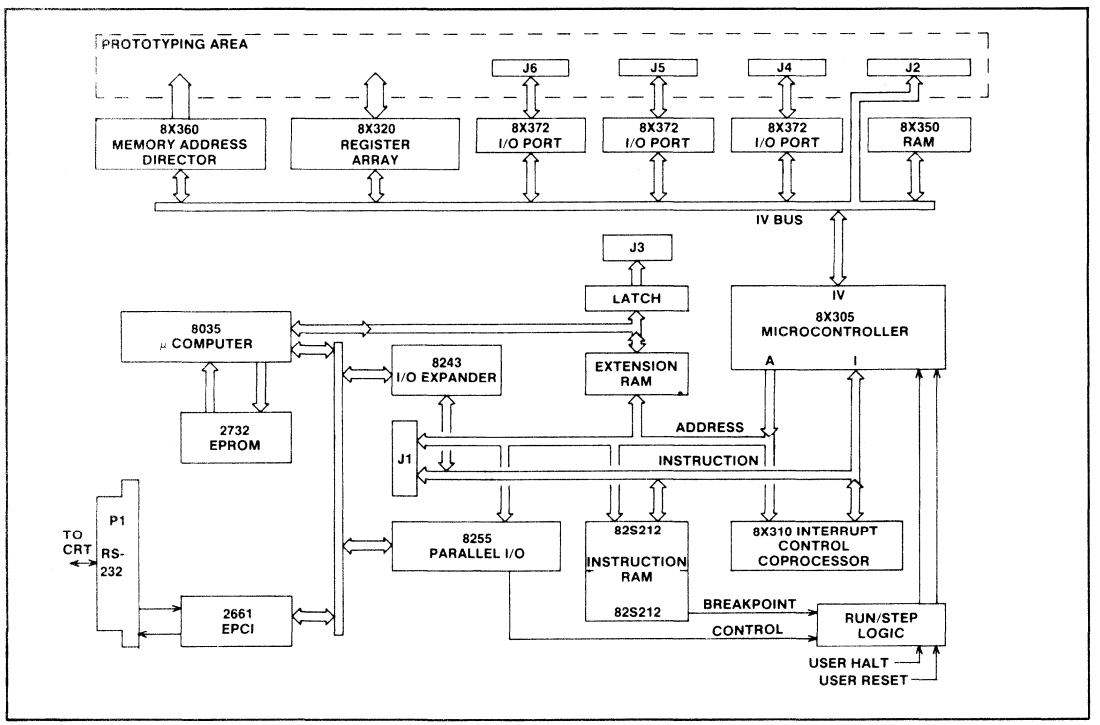


Figure 5-1. Detailed Block Diagram

5.3 WRITEABLE CONTROL STORAGE

Instead of the usual PROM or ROM instruction storage found in a typical 8X305 based system, a Writeable Control Store (WCS) has been implemented with high speed RAM to facilitate programming via the RS-232 terminal. The RAM memory provides 256 x 16 bits for 8X305 instruction storage, 256 x 8 bits for extended microcode, and 256 x 1 bit for breakpoints. If extended microcode is not desired the RAM chip at U21 may be removed and references to the extension will be removed from the display. Any one or all memory address locations may contain a breakpoint.

Note that no page decoding is provided on the board, so the 256 words of instructions will be repeated every 256 addresses throughout the entire 8K memory range of the 8X305.

The memory may be expanded up to the full 8K directly addressable by the 8X305. A 4K word Writeable Control Storage Expansion Module is available from Signetics (Part Number 8X300KT2SK). When the additional memory is installed in J1, the RAM enable (RAMEN) signal is grounded to disable the on-board 256-word memory, and the Monitor Processor is signaled to provide the correct write cycle at J1 for the added RAM. See Figure 5-2 for the differences:

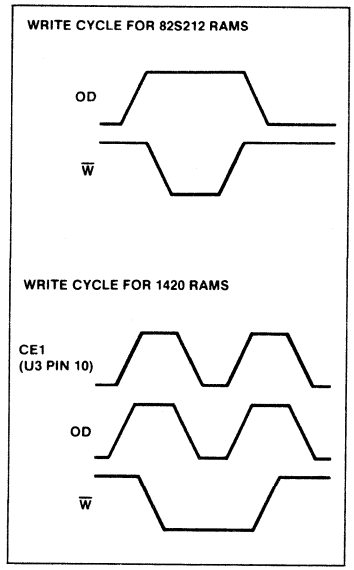


Figure 5-2. Write Cycle Variations

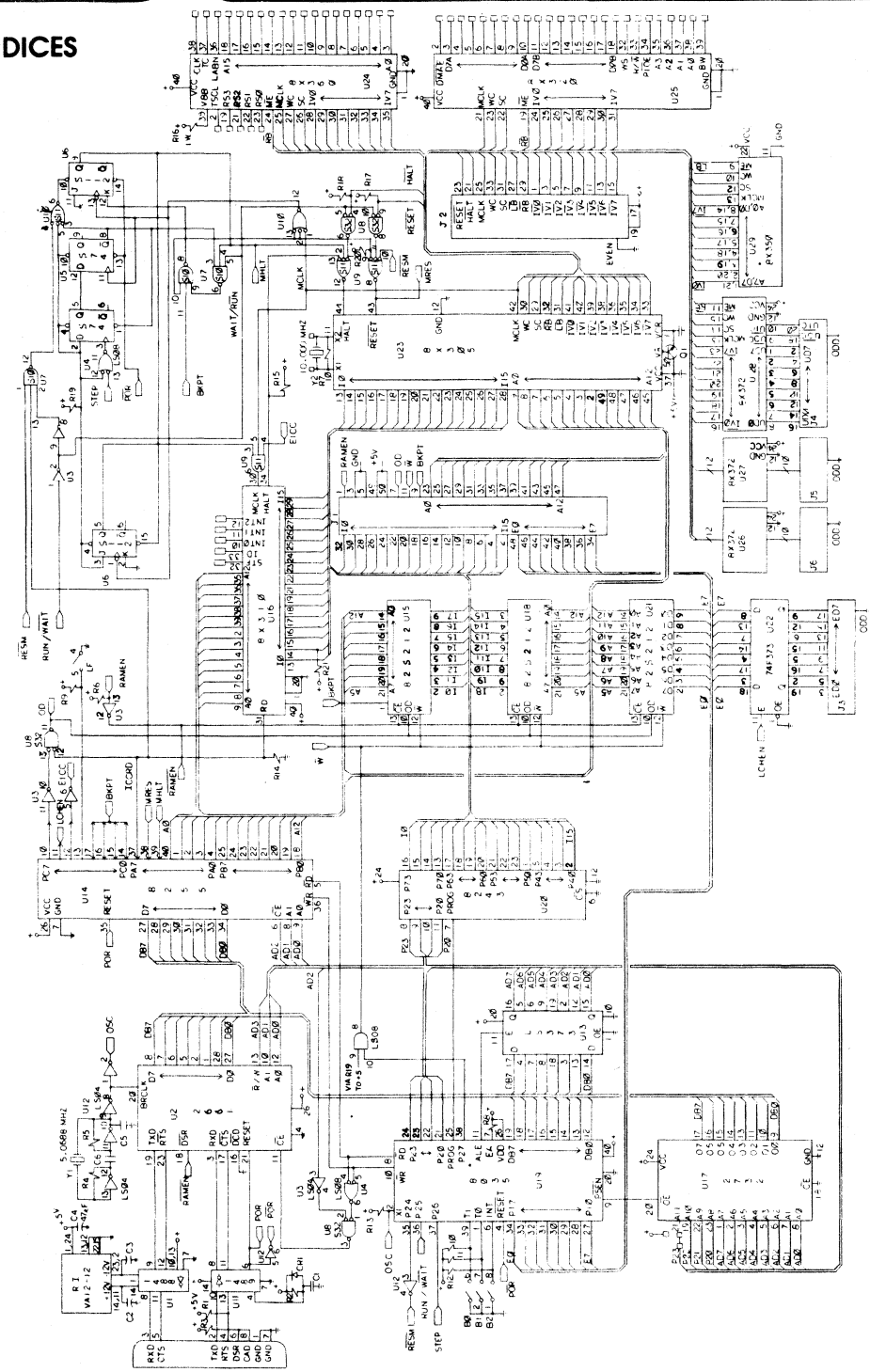
5.4 RUN STEP LOGIC

The Run-Step logic consists of components U5, U6, U7, U9 and U10 on the schematic in Appendix A. These circuits provide the control logic required to allow the 8X305 to execute instructions at full speed or in a single step mode of operation. This is easily accomplished since all instructions are executed within one machine cycle, the time from the falling edge of MCLK to the next falling edge of MCLK. The $\overline{HALT}$ input is sampled by the 8X305 sometime after the falling edge of MCLK. If it is low, the address lines of the MicroController are held stable; the current instruction is executed after the $\overline{HALT}$ input goes high (inactive). During the time that the $\overline{HALT}$ input is low (active), the MCLK output is unaffected. Inputs to the Run-Step logic are labeled RUN/$\overline{WAIT}$, STEP, BKPT and MCLK; the output is labeled $\overline{HALT}$ and connected directly to the $\overline{HALT}$ input of the 8X305 MicroController.

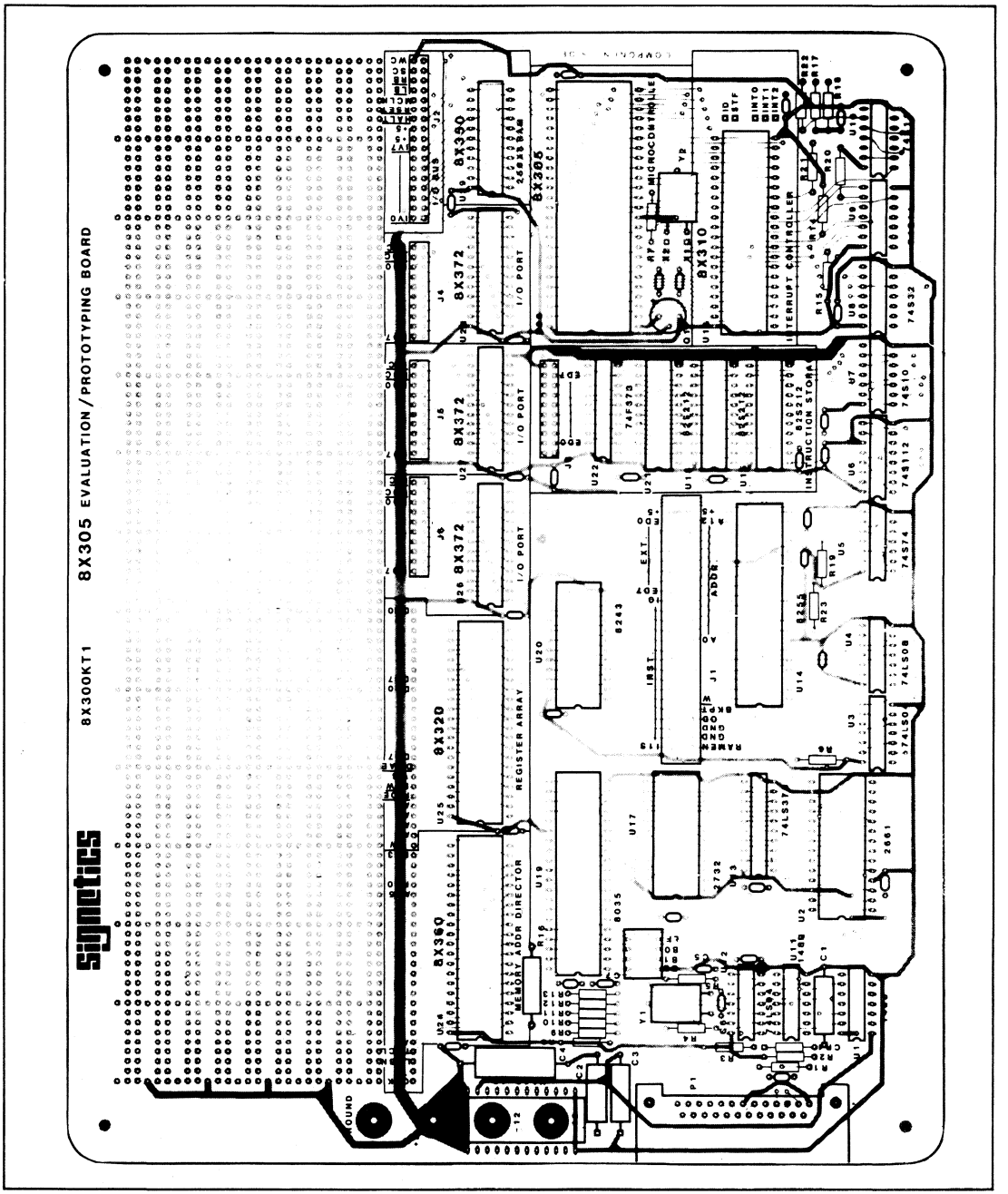
Two of the inputs, RUN/$\overline{WAIT}$ and STEP, are controlled by the Monitor Processor. The BKPT input connects to the extra bit in WCS that is used for breakpoints. The MCLK input comes directly from the MCLK output of the 8X305.

During single stepping the RUN/$\overline{WAIT}$ line is low and a pulse on the STEP line causes the 8X305 to execute only the current instruction. This is because the $\overline{HALT}$ line will go high for just one machine cycle. Entering the run mode the RUN/$\overline{WAIT}$ line is high and a pulse on the STEP line causes the 8X305 to begin executing instructions from the current address at full speed. The $\overline{HALT}$ input will go high and remain so until the RUN/$\overline{WAIT}$ line is brought low or until a breakpoint is encountered.

APPENDICES



Appendix A. 8X300KT1SK Prototyping System Schematic



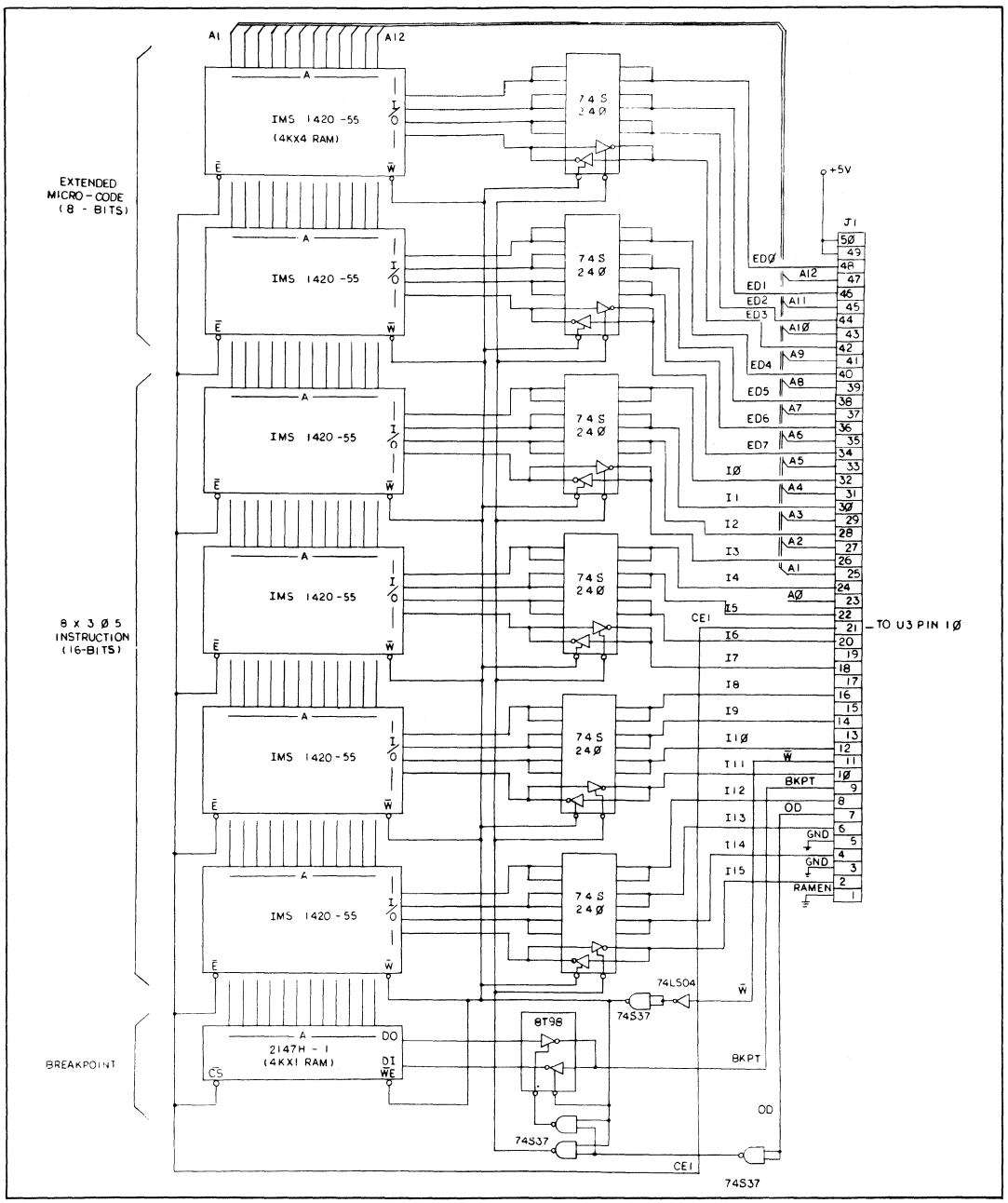
Appendix B. 8X300KT1SK PCB Layout and Parts Placement

Item No.	Manufacturer	Part Number	Description	Designator	Qty.
1	Signetics	MC1488N	Quad Line Driver	U1	1
2	Signetics	SCN2661CC1N28	EPCI	U2	1
3	Signetics	N74LS04N	Hex Inverter	U3, U12	2
4	Signetics	N74LS08N	Quad AND Gate	U4	1
5	Signetics	N74S74N	Dual D-Flip-Flop	U5	1
6	Signetics	N74S112N	Dual J-K Flip-Flop	U6	1
7	Signetics	N74S10N	Triple NAND Gate	U7	1
8	Signetics	N74S32N	Quad OR Gate	U8	1
9	Signetics	N74S11N	Triple AND Gate	U9, U10	2
10	Signetics	MC1489N	Quad Line Receiver	U11	1
11	Signetics	N74LS373N	Octal Latch	U13	1
12	Intel	P8255A	PPI	U14	1
13	Signetics	82S212N	256 x 9 RAM	U15, U18, U21	3
14	Signetics	N8X310N	Interrupt Control Coprocessor	U16	1
15	Intel	2732A-4	4096 x 8 EPROM	U17	1
16	Signetics	SCN8035AC6N40	8-Bit Microcomputer	U19	1
17	Intel	P8243	Input/Output Expander	U20	1
18	Signetics	N74F373N	Octal Latch	U22	1
19	Signetics	8X305I/N	MicroController	U23	1
20	Signetics	N8X360N	Memory Address Director	U24	1
21	Signetics	N8X320N	Register Array	U25	1
22	Signetics	N8X372-000N	I/O Port	U26	1
23	Signetics	N8X372-001N	I/O Port	U27	1
24	Signetics	N8X372-002N	I/O Port	U28	1
25	Signetics	N8X350N	256 x 8 RAM	U29	1
26	ITT CANNON	DBP-25SAA	RS-232 Connector	P1	1
27	TRW CINCH	252-25-30-360	Edge Connector, 50 Pin	J1	
28	Spectra-Strip	800-579	Header, 34 Pin	J2	1
29	Spectra-Strip	800-586	Header, 20 Pin	J3, J4, J5, J6	4

Appendix C. Parts List

Item No.	Manufacturer	Part Number	Description	Designator	Qty.
30	Reliability	VA12-12	DC-DC Converter		
31	Saronix	NMP051L	Crystal, 5.688 MHz	Y1	1
32	Saronix	NMP100L	Crystal, 10.000 MHz	Y2	1
33		2N5320	Transistor	Q1	1
34	ALCO	DSS-4	Switch, Mini-dip	B2, B1, B0, LF	1
35	Smith	230	Binding Post		4
36		1N914	Diode	CR1	1
37			CAP, 0.1 μF		28
38			CAP, 47 μF, 20 V	C2, C3, C4	3
39			CAP, 47 μF, 6 V	C1	1
40			CAP, 47 pF	C5	1
41			CAP, 0.1 μF	C6	1
42			RES., 1K, 1/4 W	R19	1
43			RES., 10K, 1/4 W	R1, R2, R3	3
44			RES., 390, 1/4 W	R4, R5	2
45			RES., 2.2K, 1/4 W	R6-R13, R15, R17, R18, R20, R21	13
46			RES., 18, 1 W	R16	1
47	H.H. Smith	2501	Bolt, Nylon 4"— 40 x 3/8"	P1	2
48	H.H. Smith	2554	HEX Nut, Nylon 4 — 40	P1	2
49	BURNDY	DILBQ50P-101	Socket, 50 Pin	U23	1
50	T.I.	C844002	Socket, 40 Pin	U14, U16, U19, U24, U25	5
51	T.I.	C842802	Socket, 0.6" 28 Pin	U2	1
52	T.I.	C842402	Socket, 0.6" 24 Pin	U17, U20	2
53	EMC	17424-01-445	Socket, 0.4" 24 Pin	U26, U27, U28	3
54	T.I.	C842202	Socket, 0.4" 22 Pin	U15, U18, U21, U29	4
55	T.I.	C842002	Socket, 0.3" 20 Pin	U22	1
56	Signetics	PCB-82001	P.C. Board		1
57	H.H. Smith	2450	Rubber Bumper		5

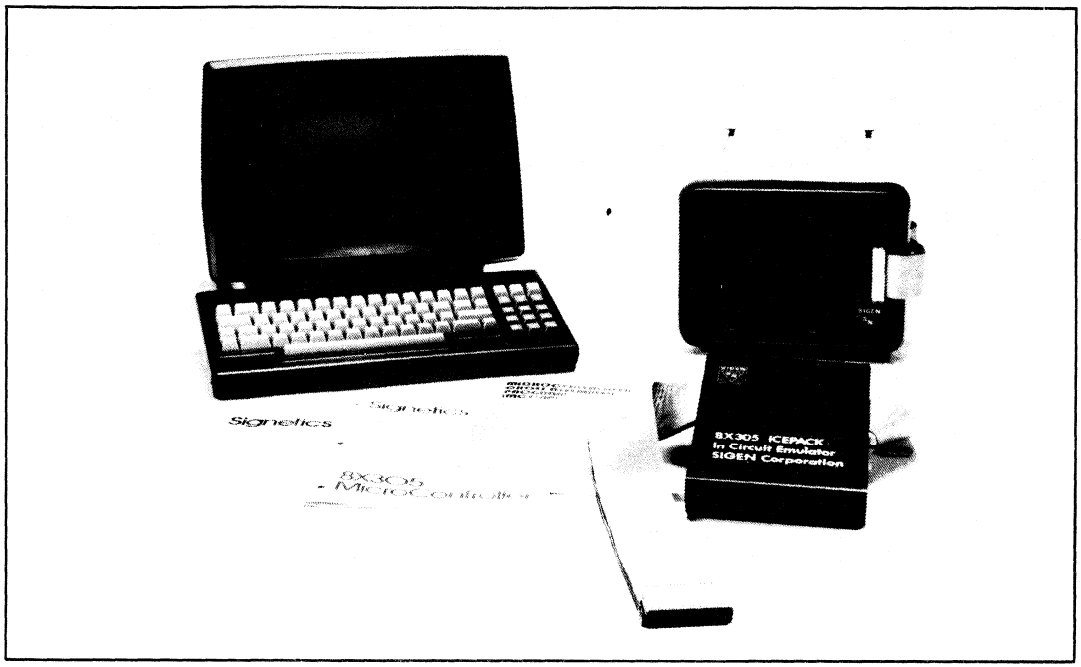
Appendix C. Parts List continued



Appendix D. Memory Expansion Assembly Schematic

ICEPACK

Originally published by Signetics January 1984



FEATURES

- Diagnostic Monitor for controlled program execution, 32 breakpoints, register, memory and I/O port examination/change, download/upload memory
- In-line assembler/disassembler for fast debugging of 8X305 code in symbolic assembly language
- System trace memory for 128 addresses, 12 bites each
- On-board emulation memory of 4096 x 24 bits
- Supports all other 8X305 family devices
- Supports microcoded designs using expanded instruction widths
- Power-on diagnostics
- Emulator and software runs with other CP/M* or ISIS* based systems

COMPLETE HARDWARE/SOFTWARE DEVELOPMENT SYSTEM INCLUDING:

- 8X305 Emulator module
- CP/M 2.2 System — Z80A, 64K RAM, dual minifloppies with 1.6 Mbytes storage
- Screen oriented editor for easy program development
- Cross Assembler supporting Signetics standard format mnemonics
- IBM P.C. version available

FASTER PRODUCT DEVELOPMENT

Demanding control applications based on the low cost, high performance Signetics 8X305 bipolar microcontroller can now be developed and implemented quickly and economically with the SIGEN 8X305 ICEPACK.

ICEPACK is a powerful, high performance development and in-circuit emulation system for use with the 8X305 series microcontroller product family. Designed for CP/M compatibility, ICEPACK provides a cost-effective means of rapid product development without the costly dedicated resources previously required.

---

*CP/M is a trademark of Digital Research Corp.
*ISIS is a trademark of Intel Corporation.

PERFORMANCE YOU NEED

ICEPACK provides both the hardware performance and software tools needed for efficient 8X305 product development throughout a broad range of applications. Designed specifically by SIGEN for the development of 8X305 based products, ICEPACK performance has been proven throughout a variety of products. Its capabilities are especially useful in real time control applications.

The ICEPACK Emulator module simply plugs into the prototype system 8X305 socket through a flat ribbon cable. The ICEPACK's superior noise immunity eliminates typical problems associated with circuit interfacing. The ICEPACK Emulator hardware features high speed electronics supporting clock operation up to 10 MHz. Rugged packaging assures long life and reliable operation.

ICEPACK software includes:

- Powerful screen oriented Editor for easy program devlopment.
- Full featured Cross Assembler supporting standard Signetics mnemonics.
- Powerful diagnostic Monitor enabling controlled program execution including single stepping, breakpoint setting, memory and register examination an modification.
- In-line assembler/disassembler for easy assembly language and debug and patching.

FITS YOUR REQUIREMENTS

ICEPACK is available ready to use as a system, or ready to interface to your floppy based CP/M system. Either way, you get the same powerful ICEPACK System capabilities. The ICEPACK System includes a Z80A based, 64K RAM, dual floppy system, ICEPACK Emulator, interface adapter, CP/M 2.2 and all ICEPACK software and manuals. The ICEPACK Subsystem includes Emulator, parallel interface adapter, interface cables, ICEPACK software, and user manuals.

TECHNICAL SPECIFICATIONS

Power Requirements: 115/230VAC, 100 W

Physical	(CP/M System)	(Emulator)
Height	7.5 inches 190mm	1 inch 25mm
Width	9.5 inches 241mm	6 inches 152mm
Depth	14.5 inches 368mm	8 inches 204mm
Weight	13.0 pounds 5.9 kg	1 pound .45 kg

Environmental

Operating Temperature: 0°C to +40°C
Storage: −40°C to +85°C

Cables

(Target System)

— 100 wire flat cable, 18 inches long
— 20 wire flat cable, 18 inches long

(CP/M System)

— 37 wire flat cable, 5 feet long

Enclosure

(Emulator)

— Anodized brushed aluminum

(CP/M System)

— High impact plastic with internal shielding

DEVELOPMENT DATA I/O PORT PROGRAMMER

Originally published by Signetics January 1984

Hardware Features

- Real time in-circuit emulation to 10 MHz without wait states.
- 8K words of 35ns RAM, 16 or 32-bit words.
- Memory mapping in 1K word increments.
- Compatible with 8X310 Interrupt Control Chip.
- Trace capability includes 128 cycles of address, IV bus, RB, LB, WC, SC, and 3 selectable points in target system. Both input and output phases are traced in each cycle.
- Downloading and Uploading capability provided.
- I/O Port Programmer for 8X372, 8X376, 8X382.

Software Features

- Relocating macroassembler compatible with Signetics MCCAP.
- Linking editor permits linking separately-assembled modules to form one load module.
- Debugger program controls single stepping, stopping on a specified address, printing trace information, disasembling, program patching, changing register contents and memory management. Symbolic debug capability is provided.
- Command file capability provided in the Asembler, Linking Editor, and Debugger program.
- PROM formatting program available. Slices 16 and 32-bit words into 4 or 8-bit groups.

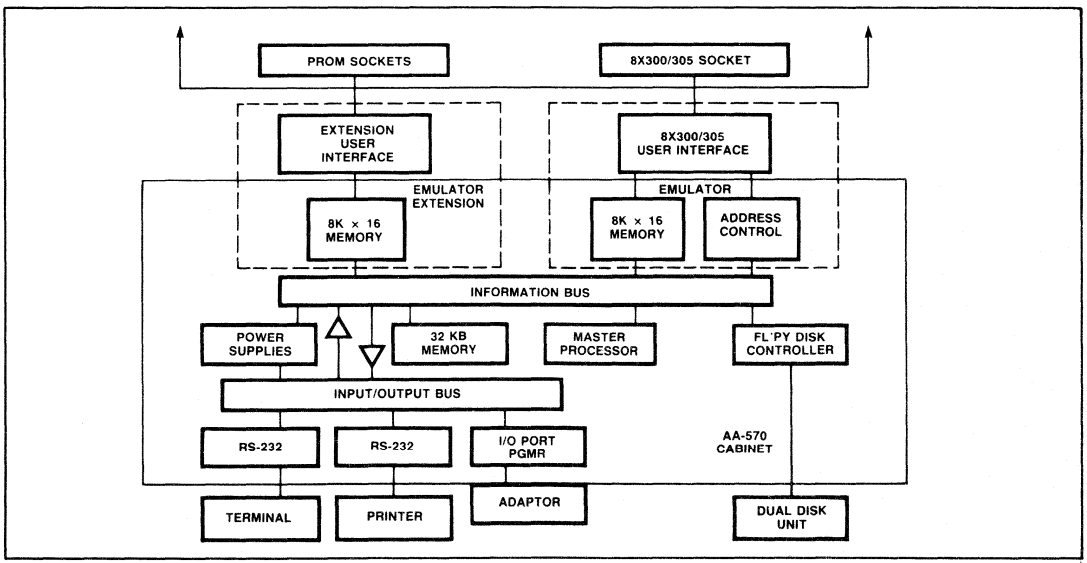
BLOCK DIAGRAM

The block diagram shows how the 8X300/305 support devices are incorporated in an EZ-PRO system. Devices unique to the 8X300/305 includes the AA-572-8X35 In-Circuit Emulator, the AA-572-8X35-M Emulator Extension and the AA-574-8X37 I/O Port Programmer. The Emulator Extension provides an extra sixteen bits of word length over the basic sixteen bits required for the 8X300/305 processor and may or may not be required in a particular development. Note that the emulator consists of three printed circuit board assemblies and the extension, two.

The Address Control assembly incorporates trace memory and logic for memory mapping, stopping, and single stepping as well as circuitry required to communicate with the User Interface and Master Processor. Each Emulator Memory assembly is equipped with 8K 16-bit words of 35 ns memory as well as interface circuitry. This memory is loaded and unloaded under control of the Master Processor and is accessed by addresses generated by the 8X300/305.

The 8X300/305 User Interface has the processor mounted on it along with cable termination networks, cable drivers, some logic and test points. Test points are provided for the three points which may be traced in the target system, connection to the 8X310 Interrupt Control Chip and oscilloscope sync.

TARGET SYSTEM



BLOCK DIAGRAM OF EZ-PRO CONFIGURED TO SUPPORT 8X300/305 DEVELOPMENT WITH 32-BIT WORK LENGTH

The Extension User Interface is equipped with six DIP sockets and cables which permit connection into PROM sockets located in the target system. Four of the sockets are 18 pin (for 4-bit wide PROMs) and two are 24 pin (for 8-bit wide PROMs). Either set of sockets and cables may be used. Pin outs are compatible with Signetics 82S137, 185, 181, and 191 PROM types.

PROM programming is supported in two ways. Both require that PMFORM, the PROM formatting program, be utilized. After PMFORM has created files consisting of either 4-bit or 8-bit wide slices of the object program words, these files may be directed to a DATA I/O (or equivalent) PROM programmer connected to the RS232 printer port. Alternatively, with 8-bit wide slices, the AA-574-27XX EPROM Programmer may be utilized to write the files into 2716 or 2732 EPROMs. With appropriate off-line equipment, information in the EPROMs may be transferred into bipolar PROMs.

The I/O Port Programmer consists of a printed board assembly, an adaptor that fits into the ZIF socket located on the front of the AA-570 Basic Development Unit and a program. After checking to see that the 8X372/376,382 is properly oriented in the ZIF socket and fuses are unblown, the program permits the desired address to be programmed into the I/O port. Complete checking is then accomplished including validation of the address and transfer of information in both directions through the port.

EZ-PRO SYSTEM ELEMENTS FOR 8X300/305 SUPPORT

Model	Description
AA-570-200	Basic Development Unit
AA-59X	Dual Disk Unit
AA-572-8X35	In-Circuit Emulator for 8X300/305
AA-572-8X35-M	Emulator Extension
AA-562	Printer, RS232 Interface
AA-563	Video Terminal with Screen Editor
AA-553	PMFORM, PROM Formatter Program
AA-574-8X37	I/O Port Programmer
AA-574-27XX	EPROM Programmer for 2716 & 2732

Note that all required programs except PMFROM are supplied at no extra cost.

303A-8X PROGRAMMING TEST ADAPTER

The 303A-8X Programming Test Adapter is designed to program address fuses and activate protect fuses for Signetics' I/O Ports 8X372, 8X374, 8X376, and 8X382. Error messages are displayed if the programmed part is defective or if ambiguous addresses occur during the programming procedure. The test adapter operates in conjunction with the Data I/O Logic PAK 303A-V01 and various models of the Data I/O (System 19, 29A, and 100A). The Programming Test Adapter is quick and easy to use and most Signetics' Franchised Distributors provide on-site programming capabilities for customer parts.

ECC APPLICATION NOTE

Originally published by Signetics January 1984

INTRODUCTION

The widespread use of floppy/hard disk technologies in Commercial and Military environments has created the need for high-speed disk controllers that are capable of both error-detection and error-correction capabilities. At the present time, most disk controllers are limited to error-detection only, with one of the more widely used schemes being that of a Cyclic Redundancy generator Checker (CRC) and associated software/hardware support. Although the CRC technique is virtually foolproof from an error-detection point of view, it has no provision for error correction. With bit and track densities constantly on the increase, it is desirable to implement a reliable error-correction feature to improve the performance and reliability of the disk controller.

This application note describes a software approach to error correction for the Signetics 8X305/8X330 based floppy disk controller and 8X305 based hard disk controller. A typical floppy disk controller system is shown in Figure 1. This error correction algorithm can correct up to a 9-bit single burst error and is fully compatible with controllers that use the CRC-CCITT ($X^{16} + X^{12} + X^5 + 1$) polynomial. The Signetics 8X330 floppy disk controller chip provides some unique features, unmatched by other LSI chips, which allow the 8X305 MicroController firmware to implement the error correction procedure.

GENERAL SCHEME FOR ERROR DETECTION AND CORRECTION

Error Detection by CRC Polynomial. The CRC scheme for error detection is implemented with a CRC generator. During transmission, a block of binary serial data passes through a preset CRC generator to generate the necessary and unique check bits for a particular block of data. The check bits are appended to the end of the data block. The complete data and check bits are then transmitted to the receiver. At the receiver, the complete data and check bits pass through the same preset CRC generator. If no errors occurred during data transmission, the CRC generator must have a remainder of all zeros.

Hardware implementation of the CRC generator is simply a feedback shift register with EX-OR gating. Figure 2 shows the equivalent circuit for the CRC-CCITT ($X^{16} + X^{12} + X^5 + 1$) generator. The CRC-CCITT polynomial is the industry standard used in floppy disk controllers; however, if desired, other CRC polynomials can be easily implemented. In any case, there are as many register stages as the highest order of the polynomial and the number of EX-OR gates is one less than the number of X-terms. The detection span for a particular polynomial equals the number of the highest order X-term. For CRC-CCITT, the detection span is 16 bits.

Error Correction by Reverse CRC Polynomial. Several different error correction algorithms are currently used in high-performance disk systems. The error correction procedure described in this application note is based on the reciprocal polynomial algorithm.

The reciprocal or reverse polynomial for a particular forward polynomial can be derived by subtracting the order of every X-term of the forward polynomial from the highest order of the forward polynomial. For example, the CRC-CCITT polynomial is $X^{16} + X^{12} + X^5 + 1$, hence the CRC-CCITT reverse polynomial is $X^{16} + X^{11} + X^4 + 1$ (i.e., $X^{16-16} + X^{16-12} + X^{16-5} + X^{16-0}$). Figure 3 shows the equivalent circuit and the hardware implementation for the CRC-CCITT forward/reverse polynomial.

As described previously, error detection is achieved by examining the remainders of the CRC generation after the complete data and CRC check bits have passed through the CRC generator. A nonzero remainder indicates a read-data error. A nonzero remainder contains information to determine the error pattern and its location. Error correction is achieved by manipulating the nonzero remainder in a reverse CRC generator.

The error correction procedure begins with the loading of the nonzero remainder into the reverse CRC generator in reverse order. The Most Significant and Least Significant Bits

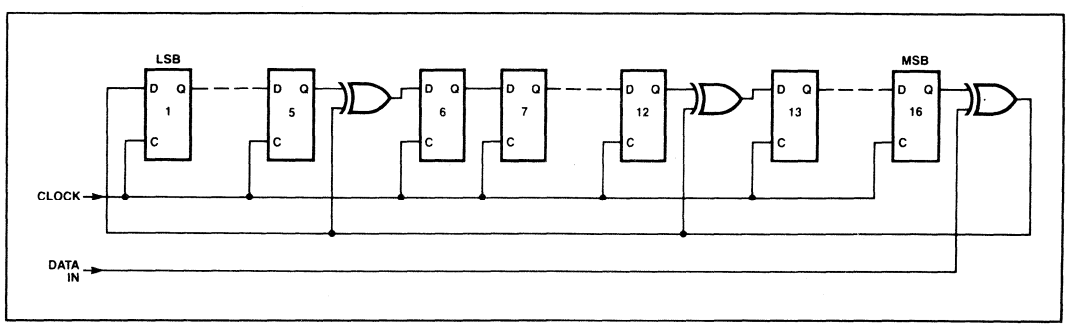
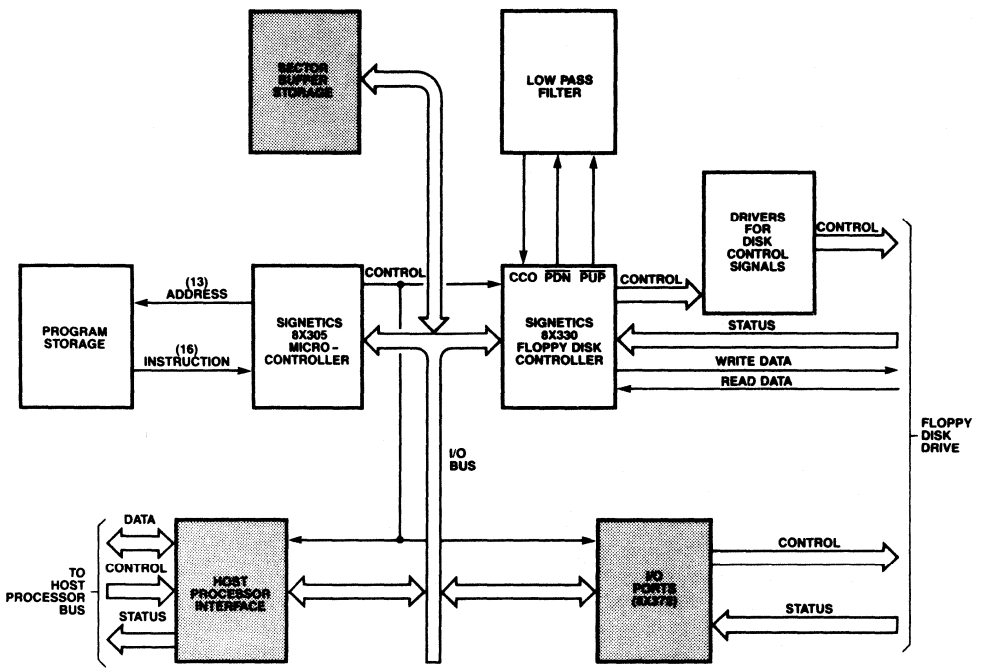


Figure 2. Equivalent Circuit for CRC-CCITT ($x^{16} + x^{12} + x^5 + 1$)



Note:
1. Shaded blocks may change to suit users system requirements.

Figure 1. 8X330 Based Floppy Disk Controller-Block Diagram

of the remainder are loaded into the Least Significant and Most Significant Bits of the reverse CRC generator, respectively. The DATA IN input of the reverse CRC generator is connected to ground to provide zero data input during the complete error correction process. Clocking is then provided for the CLOCK input. The contents of the generator are examined for an error condition after each clock cycle. If the generator has "n" register stages and "m" is the desired single burst correction span, then the error condition is characterized by (n-m) consecutive zeros at specified bit locations. The number of clock cycles needed to find the error pattern provides the error location information.

ERROR DETECTION AND CORRECTION SCHEME FOR 8X305/8X330 BASED CONTROLLER

Features of 8X305/8X330 Based Floppy Disk Controller. The following error correction scheme for 8X305/8X330 based disk controllers requires two forward CRC generators (CRC1F and CRC2F) and two reverse CRC generators (CRC1R and CRC2R). The CRC1F generator has been implemented in the 8X330 hardware. The CRC2F, CRC1R, and CRC2R generators are implemented in the 8X305 firmware. One unique feature of the 8X330 is that the remainder in the internal CRC generator can be read by the controlling processor. This feature provides the required information for the 8X305 MicroController to do error correction in the firmware. The MOS LSI floppy disk controller implementation does not provide this flexibility and only generates a read error status flag for error detection.

The speed of the 8X305 MicroController and its bit-manipulation oriented instruction set, combined with the 8-bit parallel CRC calculation algorithm described below allows the 8X305 to compute CRC for 8 bits of data input in 4 to 5 microseconds. Thus, even with a data transfer rate of 500K bits/sec for a double density 8-inch floppy disk drive, the 8X305 is fast enough to control the 8X330 and compute the CRC for CRC2F in software on-line, as each data byte is transferred between the 8X305 and 8X330 during the read and write cycles.

Two Forward CRC Polynomials. The first 16-bit forward CRC polynomial (CRC1F) is the CRC-CCITT ($X^{16} + X^{12} + X^5 + 1$), which is the standard polynomial used in floppy disk controllers. The second 16-bit forward CRC polynomial (CRC2F) is $X^{16} + X^{13} + X^{12} + X^{10} + X^8 + X^6 + X^5 + X^4 + 1$. This polynomial is chosen by computer search to achieve a single burst correction span of 9 bits. A 16-bit CRC check word is generated for each of the two polynomials. These two 16-bit CRC check words are appended to the end of the transmit data. The CRC-CCITT is chosen for the first polynomial to achieve media compatibility with old (8X300)

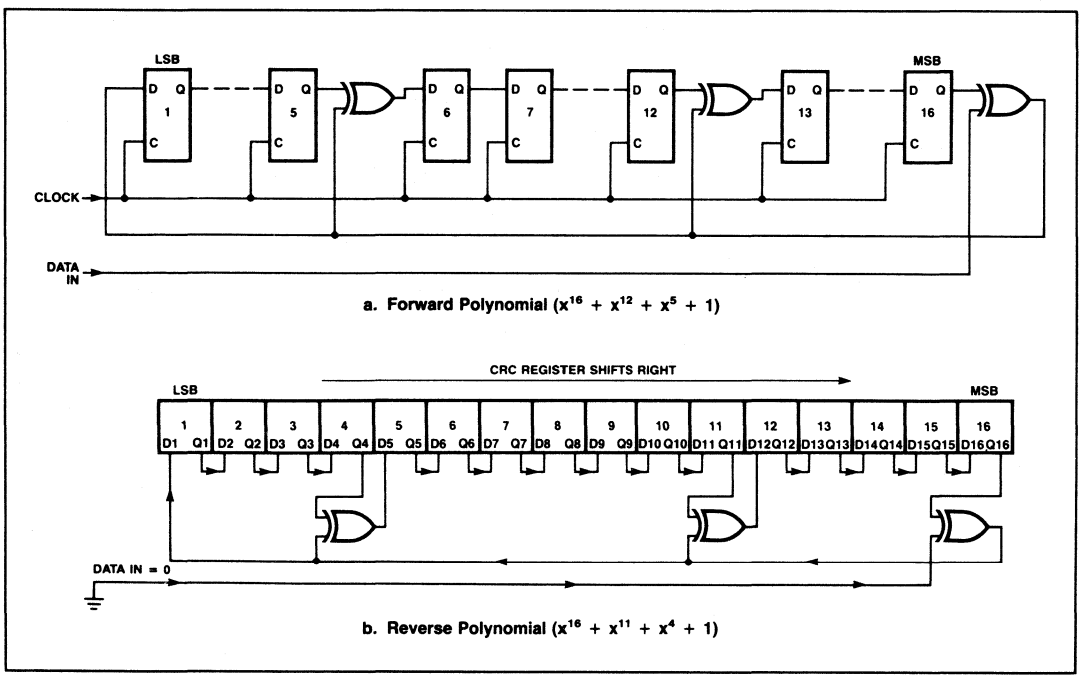


Figure 3. Equivalent Circuit and Hardware Implementation for CRC-CCITT Forward/Reverse Polynomial ($x^{16} + x^{11} + x^4 + 1$)

controllers without error correction capability and the 8X305/8X330 based controller implemented with the proposed error correction scheme. A disk generated by earlier 8X300 MicroControllers can be read by the controller with the proposed error correction scheme and vice versa.

Software Implementation of the CRC2F Generator. The hardware implementation of the CRC2F generator is illustrated in Figure 4. The CRC2F generator shifts left one bit each clock cycle. The most significant bit is located on the left side of the generator. Left shift is used to simplify the software implementation, because the MSB of the data byte transferred between the 8X305 and 8X330 is on the left side. The 8-bit parallel CRC generation mechanism is shown in Figure 5. The initial contents of the generator are represented by symbols "a b c . . . n o p" with "a" and "p" being the MSB and LSB, respectively.

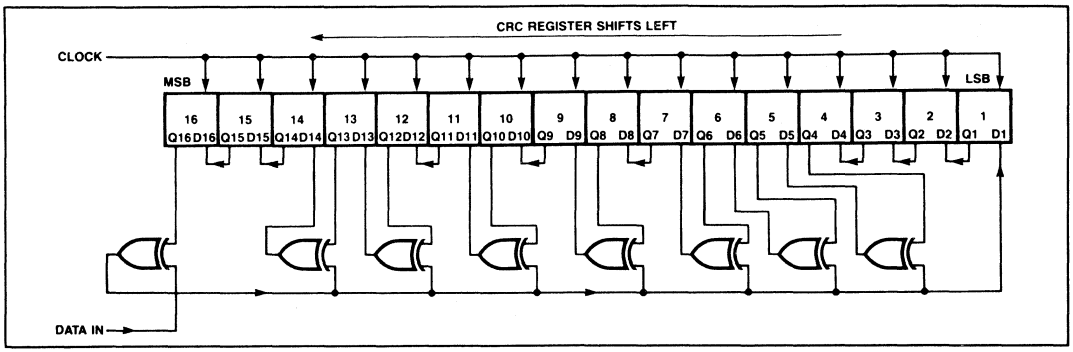


Figure 4. Hardware Implementation for $x^{16} + x^{13} + x^{12} + x^{10} + x^8 + x^6 + x^5 + x^4 + 1$ (CRC2F)

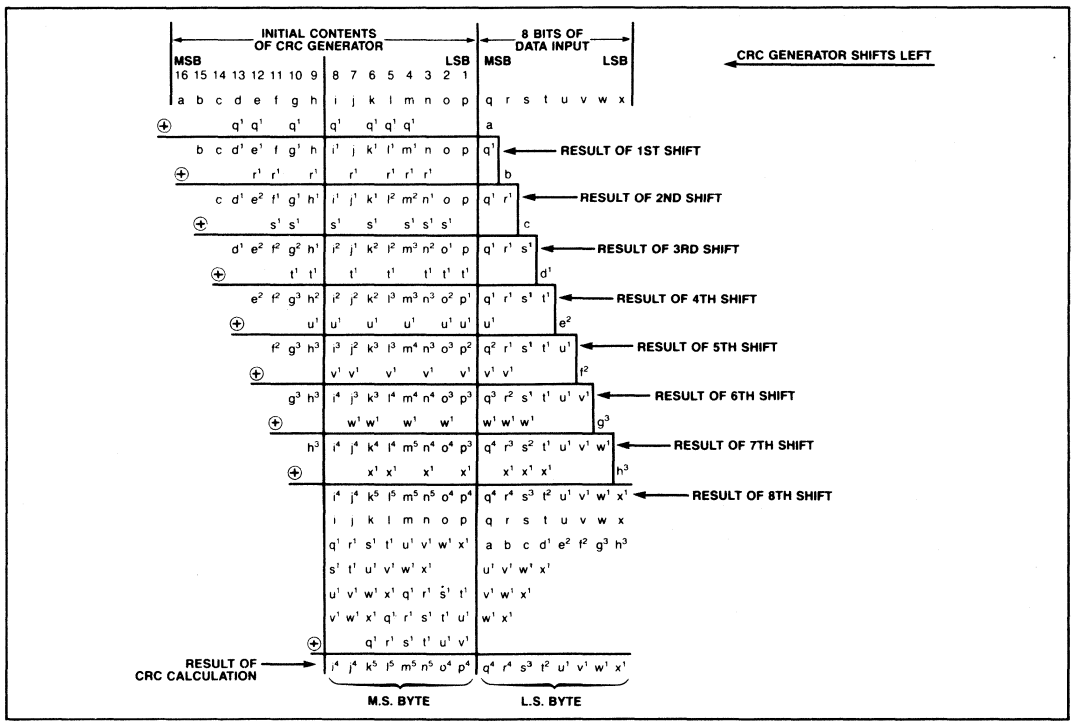


Figure 5. 8-Bit Parallel CRC Generation Mechanism for $x^{16} + x^{13} + x^{12} + x^{10} + x^8 + x^6 + x^5 + x^4 + x^1$

The eight bits of data input are symbolized by "q r s t u v w x", with "q" being the MSB. The top part of Figure 5 is a symbolic representation of the eight left shift operations of the CRC generator shown in Figure 4, for the eight serial data input bits. As the first data bit "q" comes in, it is EX-ORed with the MSB of the CRC generator. The result of this operation, "q′", is then fed back and EX-ORed with the outputs of several register stages as shown in Figures 4 and 5. Then the generator is clocked and shifted left by one bit postion. This sequence of operations is repeated for the eight data inputs shown at the top of Figure 5. The bottom part of Figure 5 summarizes the number of EX-OR operations required to derive the result. The operands of the EX-OR operations are mostly the field, rotated field, or subfield of the byte "q′r′s′t′u′v′w′x". This special characteristic, together with the 8X305 bit manipulation instruction set, permits an 8-bit parallel CRC calculation for a 16-bit polynomial to be accomplished in 4 to 5 microseconds. Figure 6 shows the software steps required to implement the 8-bit parallel CRC calculation. The first step is to derive the byte "q′r′s′t′u′v′w′x′" from the 8-bit data input and the initial contents of the CRC generator by operations A, B, and C. The result is derived by a sequence of EX-OR operations as shown in Figure 6. The software implementation of CRC2F requires a total of 21 instructions, as shown in line numbers 448 to 471 of the program listing.

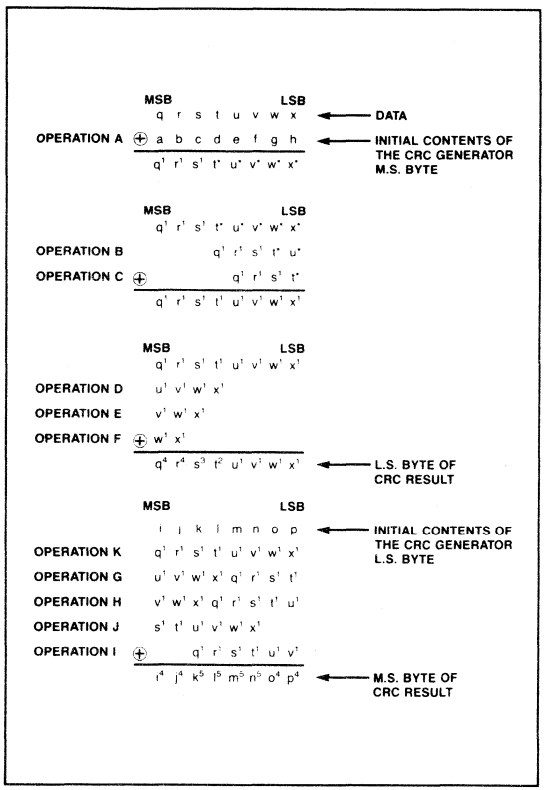


Figure 6. Steps Required for the Software Implementation of CRC2F

Two Reverse CRC Polynomials. Since the error correction scheme described in this application note uses the reciprocal polynomial algorithm, two reverse CRC generators, corresponding to CRC1F and CRC2F, are required during the correction cycle. The two reverse CRC generators are referred to as CRC1R and CRC2R.

$$CRC1R = X^{16} + X^{11} + X^4 + 1$$
$$CRC2R = X^{16} + X^{12} + X^{11} + X^8 + X^6 + X^4 + X^3 + 1$$

Software Implementation of the CRC1R and CRC2R Generators. The implementation of the CRC1R and CRC2R generators in software is similar to that of the CRC2F. There are two exceptions:

- During the correction cycle, the DATA IN inputs of the reverse CRC generators are connected to ground.
- The reverse CRC generators shift right, so that the MSB is on the right-hand side, thus simplifying the software. As errors are detected, the error correction procedure requires that the nonzero remainders of the forward CRC generators be loaded into the reverse CRC generators in reverse order; that is, the MSB from the forward CRC generator is loaded into the LSB of the corresponding reverse CRC generator. This process is achieved by loading the most significant byte of the forward generator into the least significant byte of the reverse generator, because the MSB of the forward generator is on the left side and the MSB of the reverse generator is on the right side.

Figures 7 and 8 show the mechanics and steps required to implement the CRC1R generator of Figure 3 in software. A similar procedure is used to implement the CRC2R generator. The software coding of the CRC1R and CRC2R are shown in the line numbers 487 through 541 of the program listing.

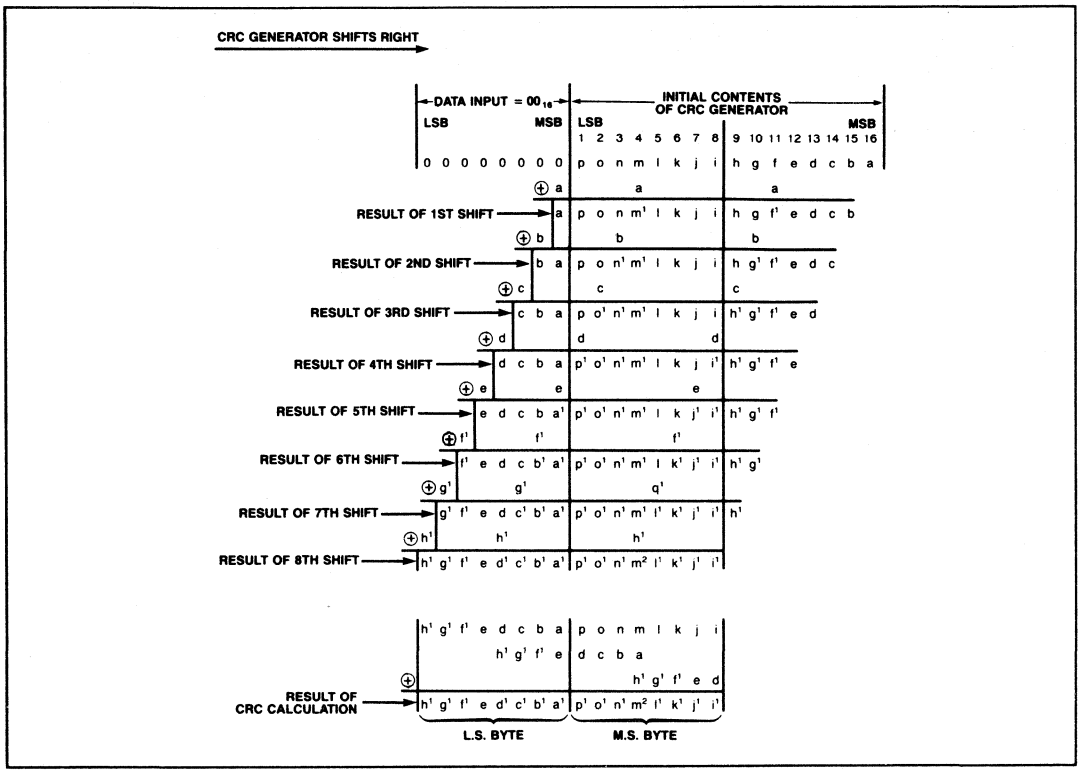


Figure 7. 8-Bit Parallel CRC Generation Mechanism for CRC1R

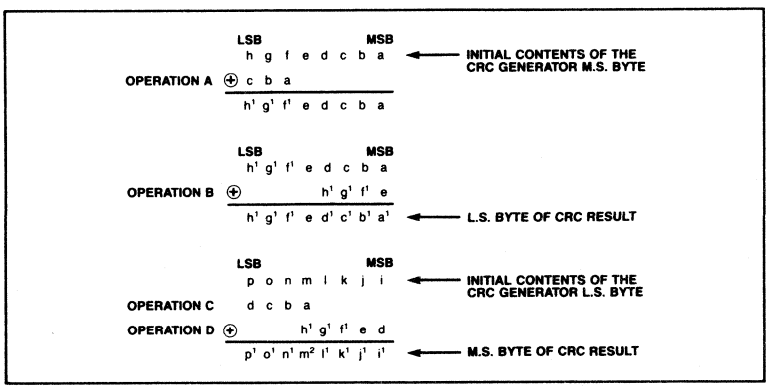


Figure 8. Steps Required for the Software Implementation of CRC1R

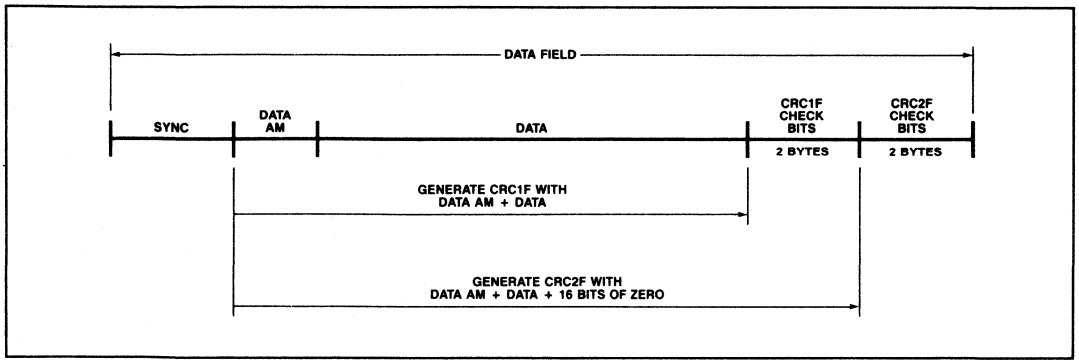


Figure 9. Write Cycle

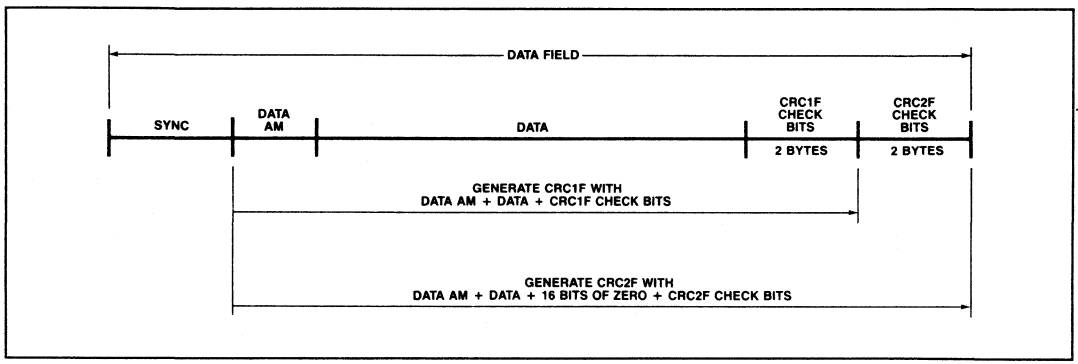


Figure 10. Read Cycle

Error Detection Scheme. As shown in Figure 9, four CRC check bytes are appended to the last bit of data during the write cycle. The first two check bytes are derived by passing the data address mark and data bytes through the preset CRC1F generator. After accepting the data address mark and data bytes, the contents of the CRC1F generator are the two CRC1F check bytes. The two CRC2F check bytes are generated by applying the data address mark, the data, and sixteen bits of zero as inputs to the preset CRC2F generator.

During the read cycle, the preset CRC1F generator calculates from the first bit of data address mark to the last bit of CRC1F check bit as illustrated in Figure 10. During the read cycle the data address mark, the data, sixteen bits of zero, and two CRC2F check bytes are input to the preset CRC2F generator. No error is detected if the remainders of the CRC1F and CRC2F generators are zeros. The types of errors associated with nonzero remainders are as follows:

CRC1F Remainder	CRC2F Remainder	
Zero	Zero	No error
Zero	Nonzero	Correctable error in CRC2F check bytes (or uncorrectable error)
Nonzero	Zero	Correctable error in CRC1F check bytes (or uncorrectable error)
Nonzero	Nonzero	Correctable error in data bytes

Error Correction Scheme. Once an error has been detected, the nonzero remainders of the two forward generators are used to determine the error pattern and location by loading them into the corresponding reverse generators in reverse order. The reverse generators are then generated with zero data input and checked for error conditions, as shown in Figure 11. The method of implementing the error correction scheme is illustrated in Figures 12 through 18. The process of locating an error starts with checking for errors in

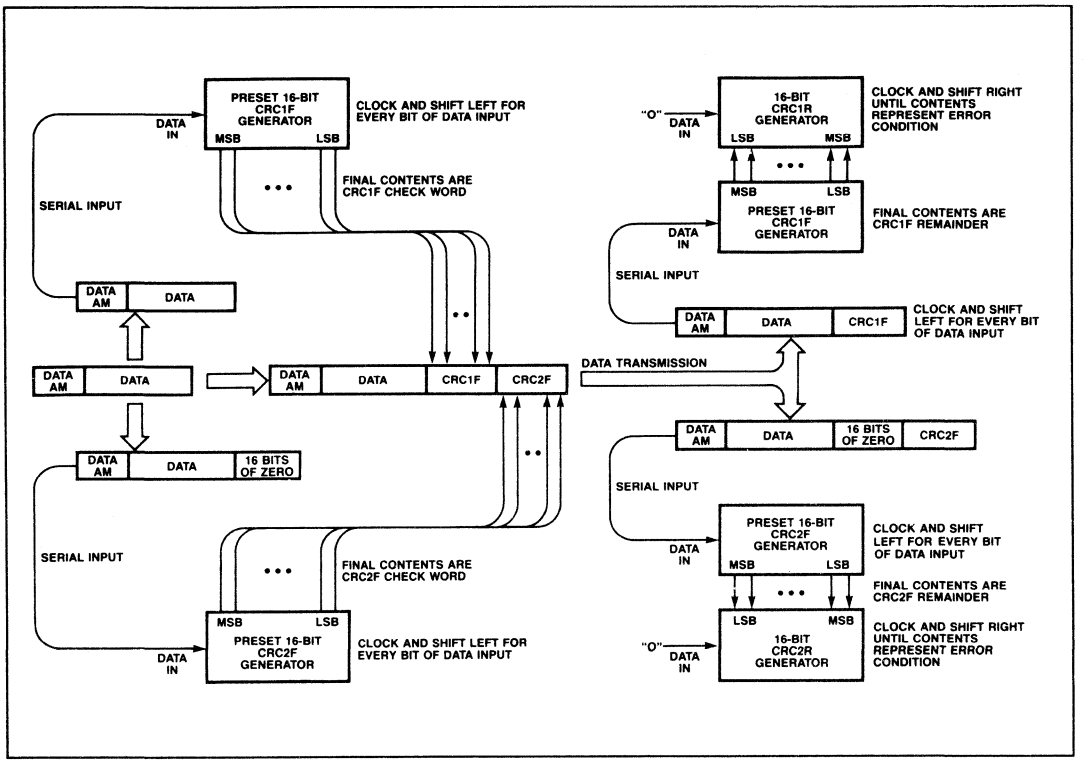


Figure 11. Data Flow

the CRC2F least significant check byte and working backwards towards the first data byte of the sector. If errors are found in the CRC check bytes, the controller will raise a CRC error status flag for the host CPU to take the appropriate action.

During the first two 8-bit parallel reverse generation cycles, only the CRC2R is generated with zero data input, the CRC1R generator remains idle. If errors occur in the CRC2F check bytes, the CRC1R should have 16 bits of zeros and the seven MSBs of CRC2R should be zeros during any of the first sixteen reverse generation clock cycles. Since the CRC2R software reverse generates one byte at a time, it is possible to miss the CRC2R error condition, because the error condition may occur in between the byte boundary. In order to solve this problem, a CHECK subroutine is written to monitor the error condition within the boundary limits. If the error condition is not found, the CHECK subroutine will swap the most significant and least significant bytes of the generator to maintain the contents of the generator before getting into the subroutine.

After the second 8-bit parallel reverse generation cycle, both the CRC1R and CRC2R are reverse generated one byte at a time until the error is found or the error is determined to be uncorrectable. An error is uncorrectable if it cannot be found within 260 8-bit reverse generation cycles. In the third 8-bit reverse generation cycle, the procedure checks for errors in the CRC1F least significant byte or errors that spanned the CRC1F byte and the CRC2F most significant byte. The error conditions are eight bits of zeros for the CRC1R most significant byte and the CRC2R least significant byte at the byte boundary. If these conditions occur, a 16-bit word is formed by concatenating the most significant byte of CRC2R with the least significant byte of CRC1R, and the CHECK subroutine, as described previously, is used to check for seven leading zeros within eight one-bit rotations towards the LSB. This condition is required to assure that errors occur in CRC1F least significant check byte or between CRC1F and CRC2F check bytes. In the fourth 8-bit reverse generation cycle, the procedure checks for errors in the CRC1F check bytes. The error conditions are 16 bits of zeros for CRC2R and seven

leading zeros in CRC1R found by the CHECK subroutine. Errors occurring in the last data byte or errors that span the CRC1F most significant check byte and the last data byte are checked in the fifth 8-bit reverse generation cycle. The error conditions are zeros in the CRC2R most significant byte, a match of the CRC1R and CRC2R least significant byte, and seven leading zeros in CRC1R found by the CHECK subroutine. The sixth to the two hundred sixtieth 8-bit reverse generation cycles check for errors in data byte 255 to the first data byte. The error conditions are a match in CRC1R and CRC2R, and seven leading zeros in CRC1R found by the CHECK subroutine.

The error conditions described in the previous sections determine when to stop the reverse generation cycle. The number of reverse generation cycles required to meet the error conditions are used to calculate the error location by the following equation:

Error location in terms of byte number = Record length in bytes + 4 CRC check bytes − number of reverse generation cycles + 1

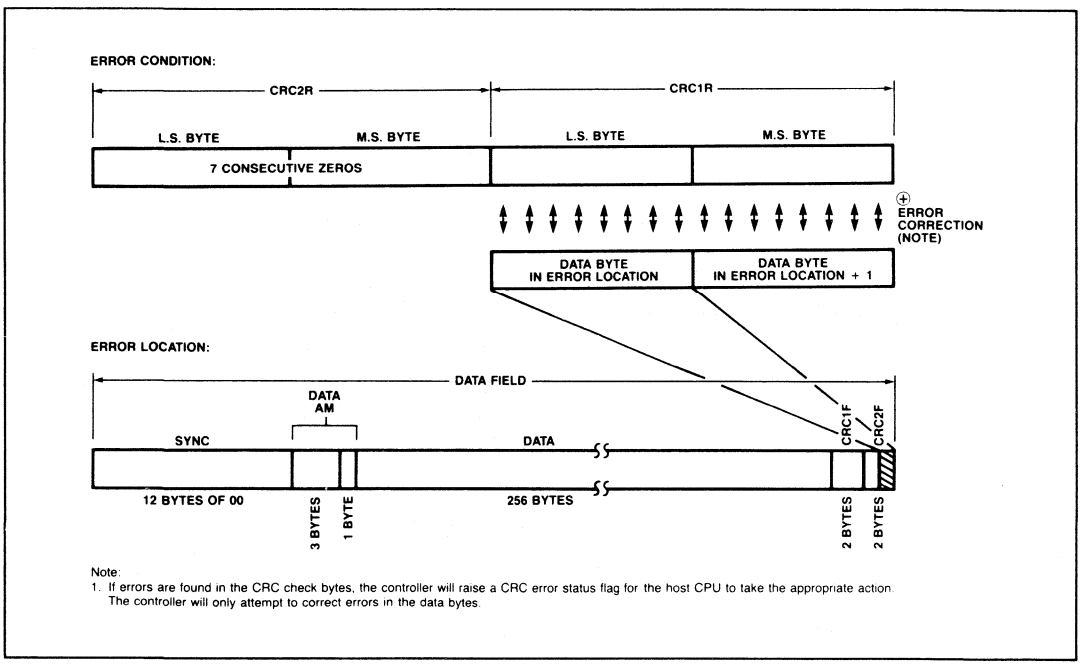
If the error location is one, then errors exist in the first data byte or between the first and second data byte of the sector. For example, if errors exist between the first and second data bytes of a 256-byte sector, then the error conditions are met in the two hundred sixtieth reverse generation cycle.

$$\text{Error location} = 256 + 4 - 260 + 1$$
$$= 1$$

The error pattern is found in the CRC1R generator. Error correction is done by XORing the data bytes in the error location and error location plus one with the CRC1R least significant and most significant byte respectively, as illustrated in Figure 11.

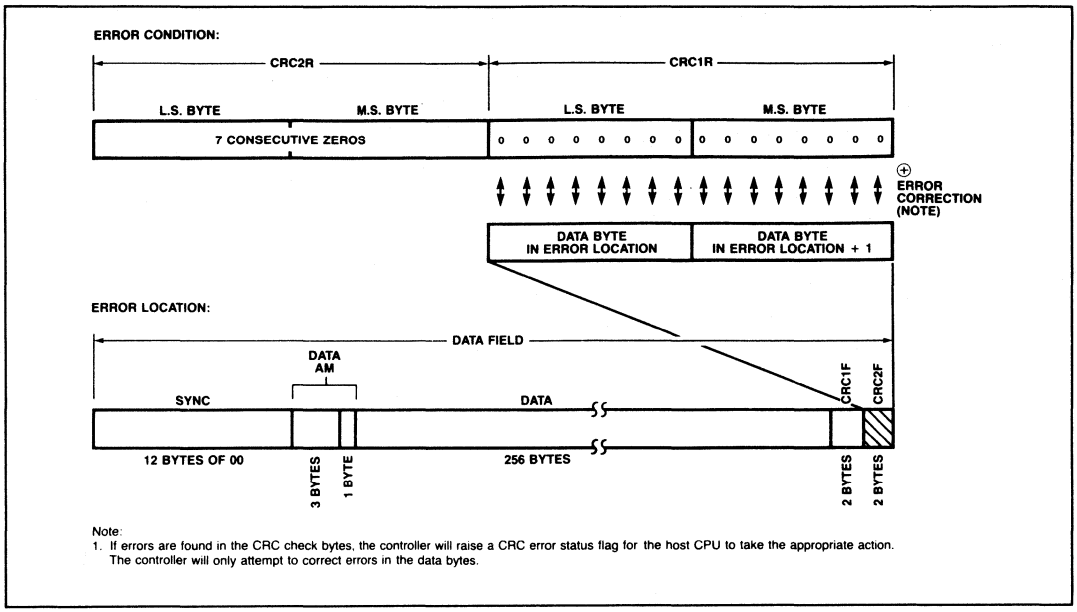
CONCLUSION

This application note describes one unique error correction algorithm for the 8X305 based floppy or hard disk controller. The algorithm can correct up to a 9-bit single burst error. The error correction procedure can be implemented with approximately 300 8X305 instructions. The worst case error correction time for a 256-byte sector is 3.0 msec.



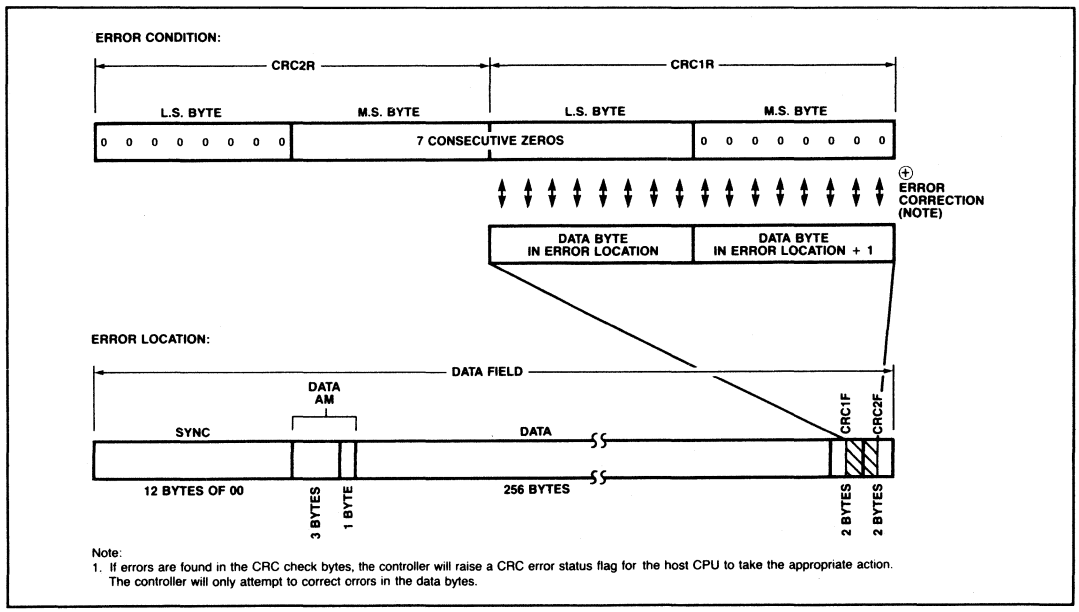
Note:
1. If errors are found in the CRC check bytes, the controller will raise a CRC error status flag for the host CPU to take the appropriate action. The controller will only attempt to correct errors in the data bytes.

Figure 12. Reverse Generation Cycle #1 (CRCIR not reverse generated)



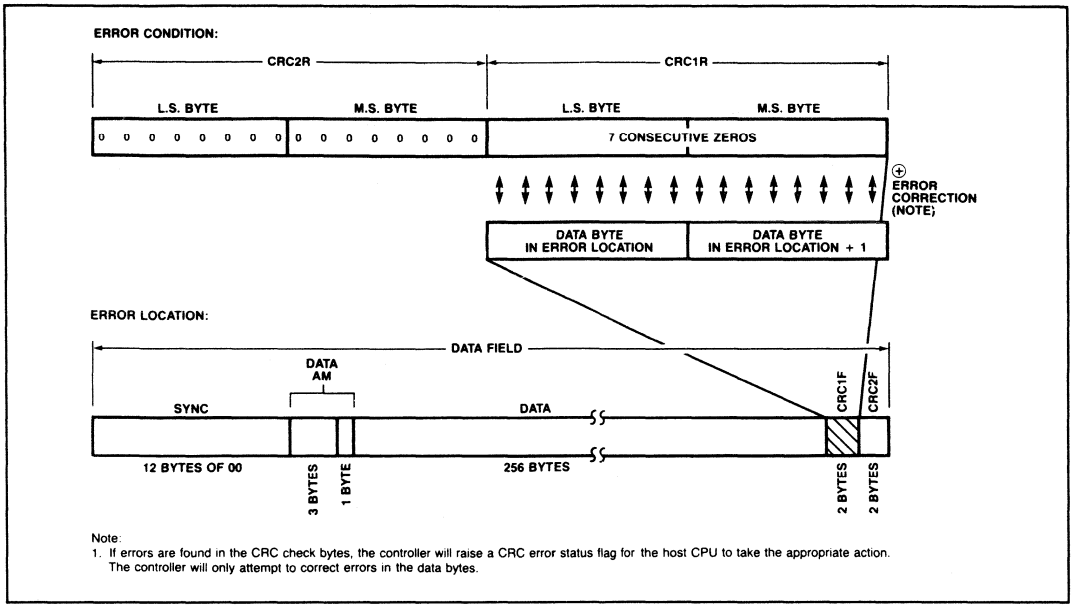
Note:
1. If errors are found in the CRC check bytes, the controller will raise a CRC error status flag for the host CPU to take the appropriate action. The controller will only attempt to correct errors in the data bytes.

Figure 13. Reverse Generation Cycle #2 (CRCIR not reverse generated)



Note:
1. If errors are found in the CRC check bytes, the controller will raise a CRC error status flag for the host CPU to take the appropriate action. The controller will only attempt to correct errors in the data bytes.

Figure 14. Reverse Generation Cycle #3



Note:
1. If errors are found in the CRC check bytes, the controller will raise a CRC error status flag for the host CPU to take the appropriate action. The controller will only attempt to correct errors in the data bytes.

Figure 15. Reverse Generation Cycle #4

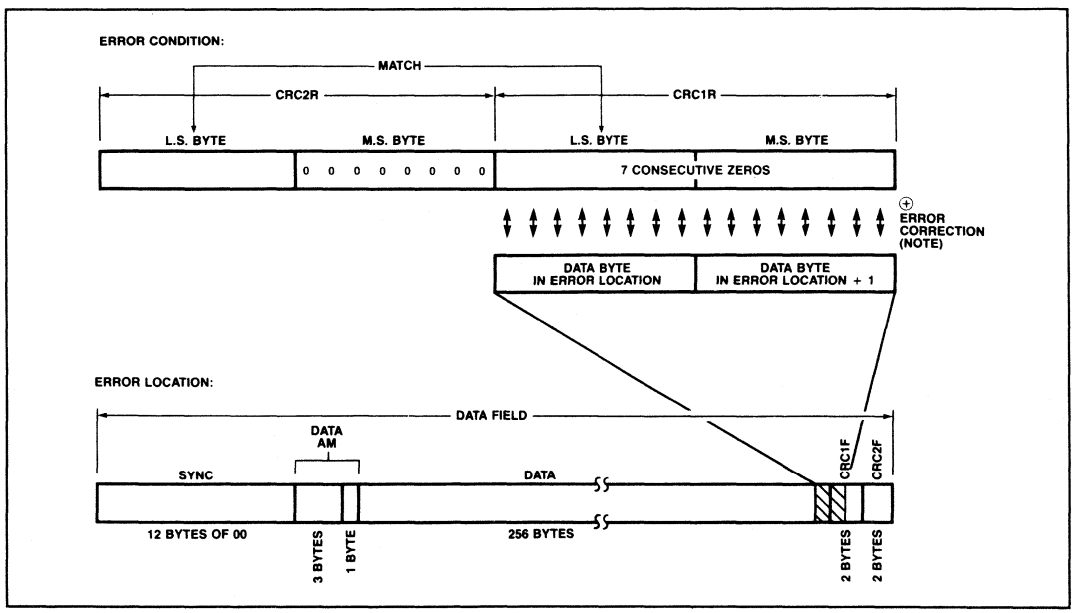


Figure 16. Reverse Generation Cycle #5

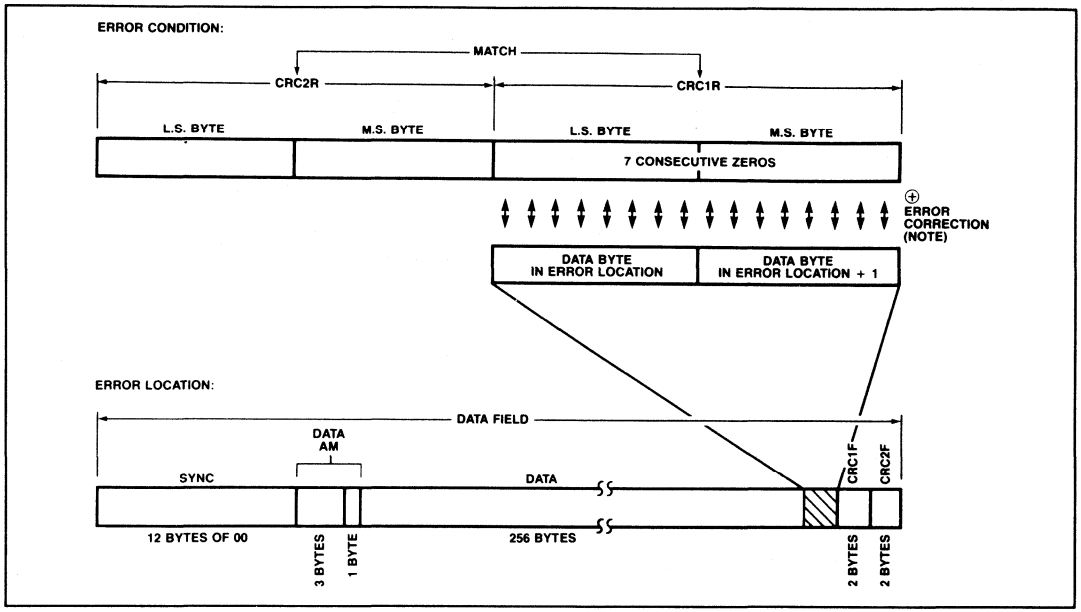


Figure 17. Reverse Generation Cycle #6

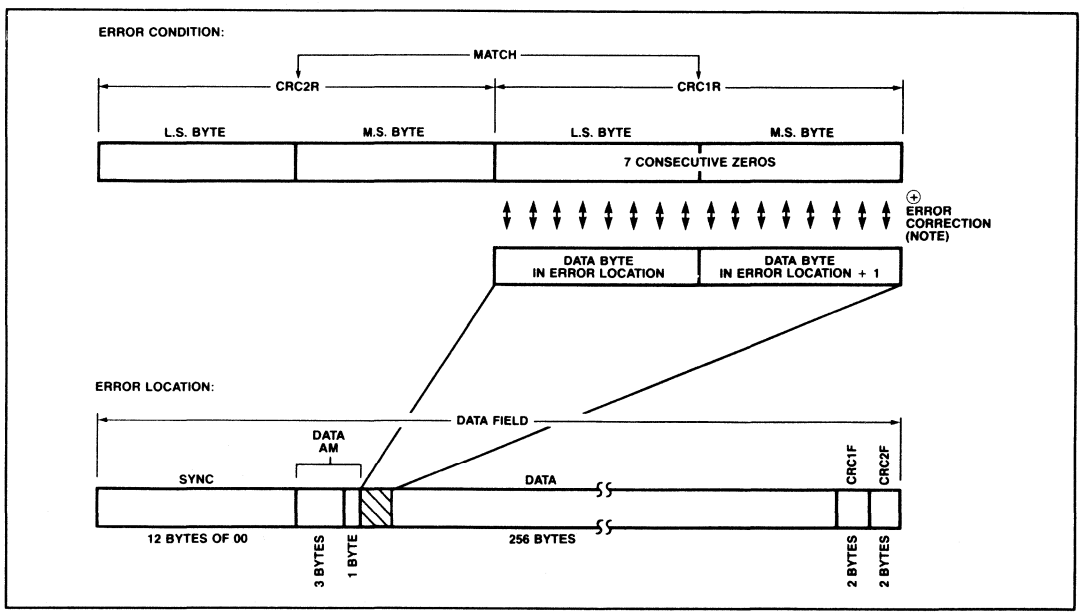
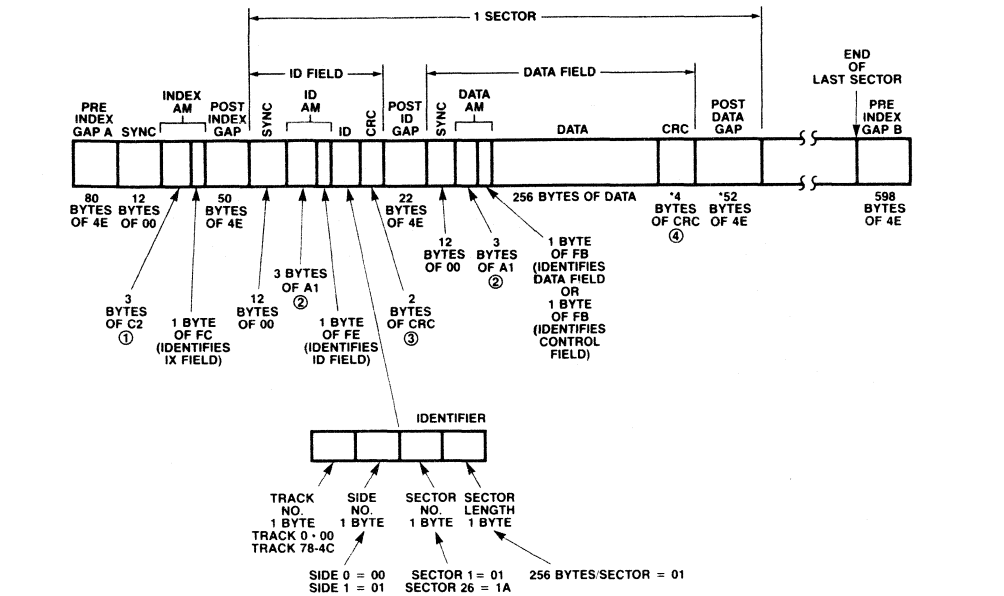


Figure 18. Reverse Generation Cycle #260



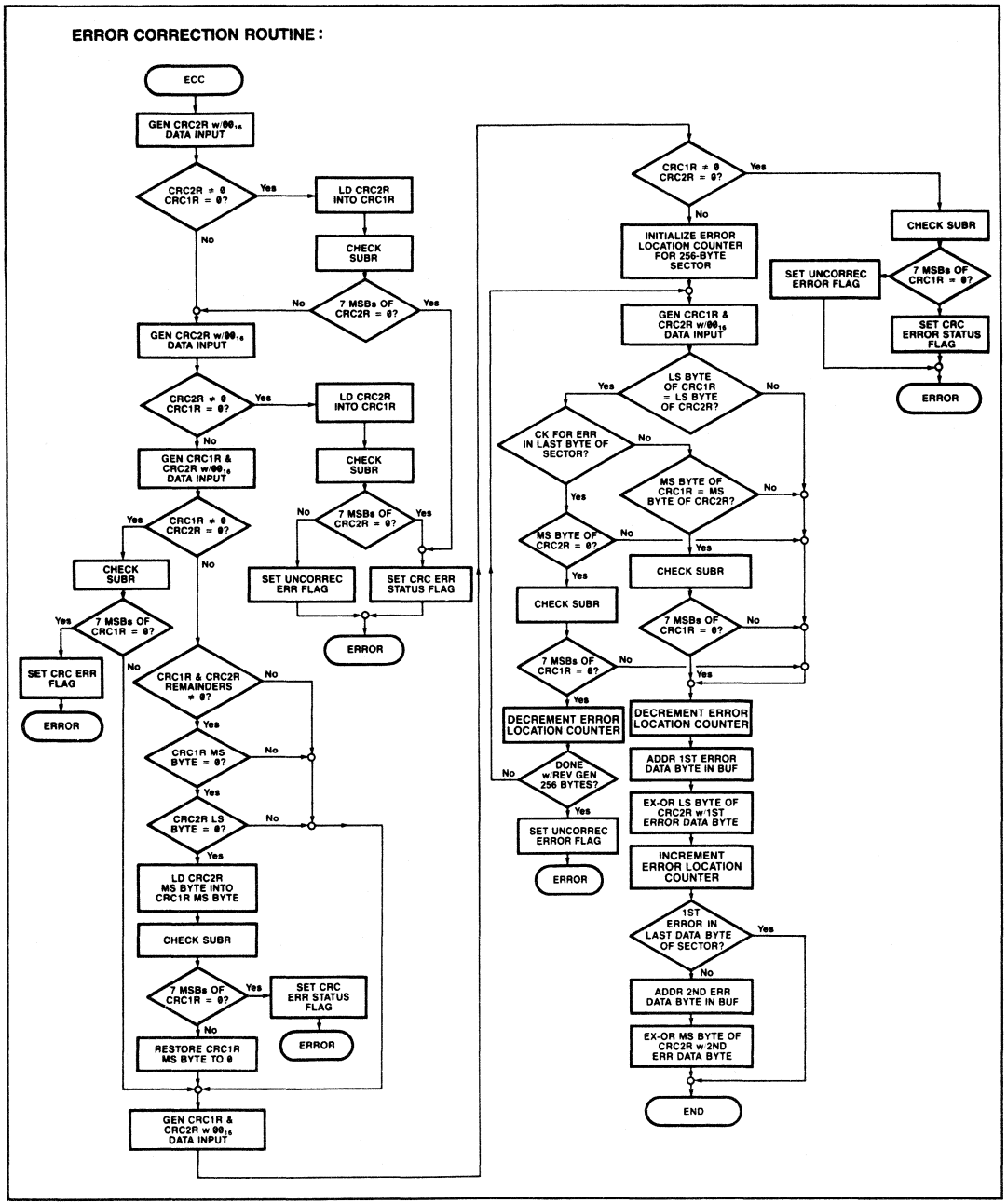
Notes:
*Indicates the modification from the IBM System 34 data track format.
1. These bytes missing clock transitions between data bits 3 and 4. Data flux transition = C2, clock flux transition = 14.
2. These bytes missing clock transitions between data bits 4 and 5. Data flux transitions = A1, clock flux transition = 0A.
3. CRC calculation starts at first byte of ID address mark and ends on the sector length byte using the CRC-CCITT polynomial.
4. The first two bytes are derived from the CRC-CCITT polynomial. The second two bytes are derived from a second 16-bit polynomial ($X^{16} + X^{13} + X^{12} + X^{10} + X^8 + X^6 + X^3 + X^4 + 1$). The CRC calculation of the 1st polynomial starts at first byte of data AM and ends on last byte of data. The CRC calculation of the 2nd polynomial includes data address mark, data bytes and two bytes of zeros.
5. There are 26 sectors of 256 bytes.

Figure 19. Modified IBM System 34 Data Track Format

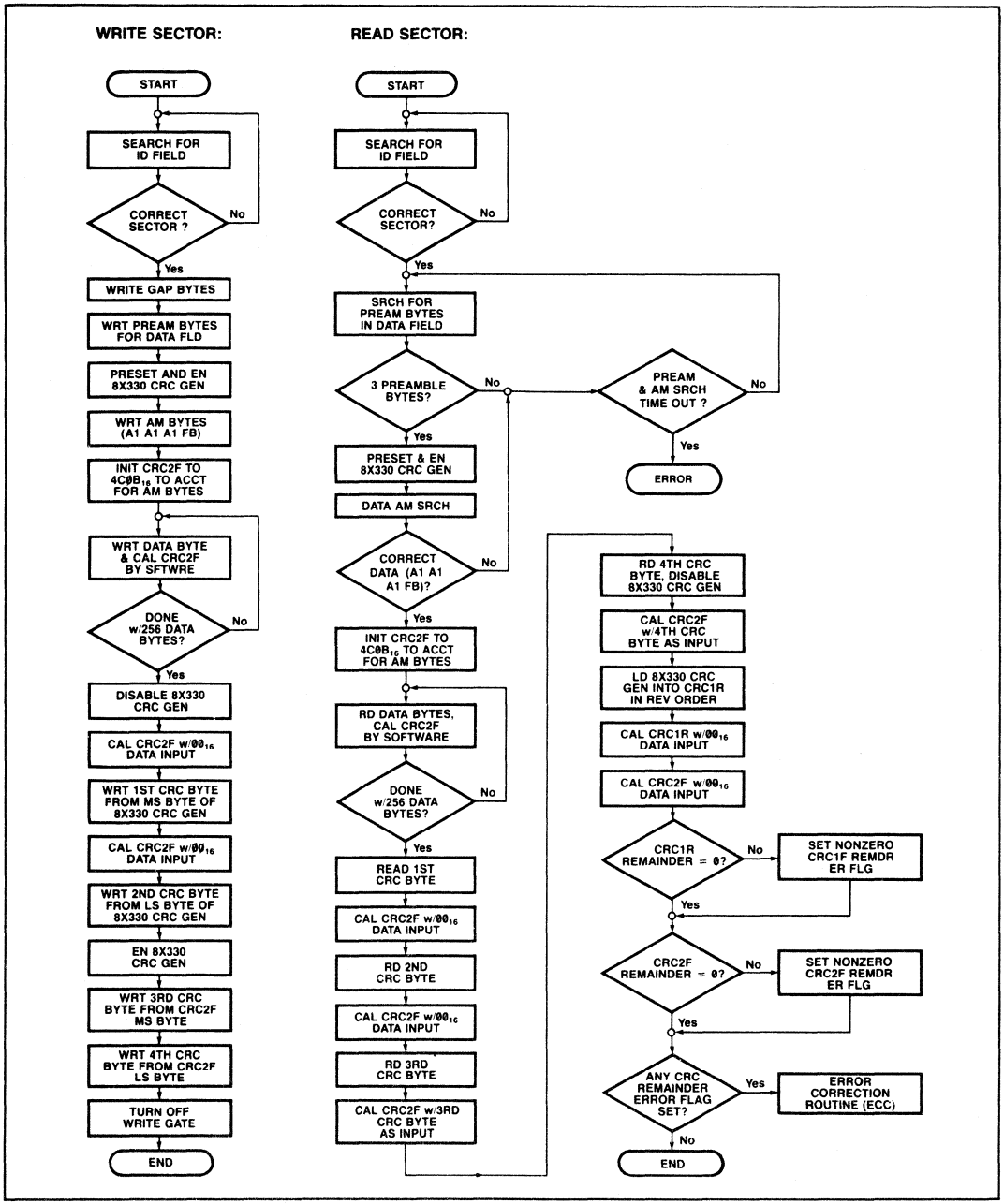
Appendix A
FLOWCHARTS

The CRC generators used to generate four CRC check bytes during the write cycle and to detect errors during the read cycle have been discussed previously and are documented by the WRITE SECTOR and READ SECTOR flowcharts that follow. One point in the READ SECTOR flowchart needs clarification. The error correction algorithm calls for the CRC1F to stop calculation after inputting the CRC1F least significant check byte during the read cycle. There is a slight modification to this procedure with the 8X330 implementation. The 8X330 internal CRC-CCITT generator (i.e., CRC1F) continues to generate up the CRC2F least significant check byte. That means two additional bytes, namely the CRC2F check bytes, are inputted to CRC1F. It is necessary to recover CRC1F remainders right after the input of the CRC1F least significant check byte as required by the error correction algorithm. This information can be recovered by loading the CRC1F remainders, which include the CRC2F check bytes in the generation, onto the CRC1R in reverse order, and then the CRC1R is generated with two bytes of zeros as shown in the READ SECTOR flowchart. At this point, the remainder of the CRC1R has the same information as if the CRC1F stopped generation after the input of the CRC1F least significant check byte.

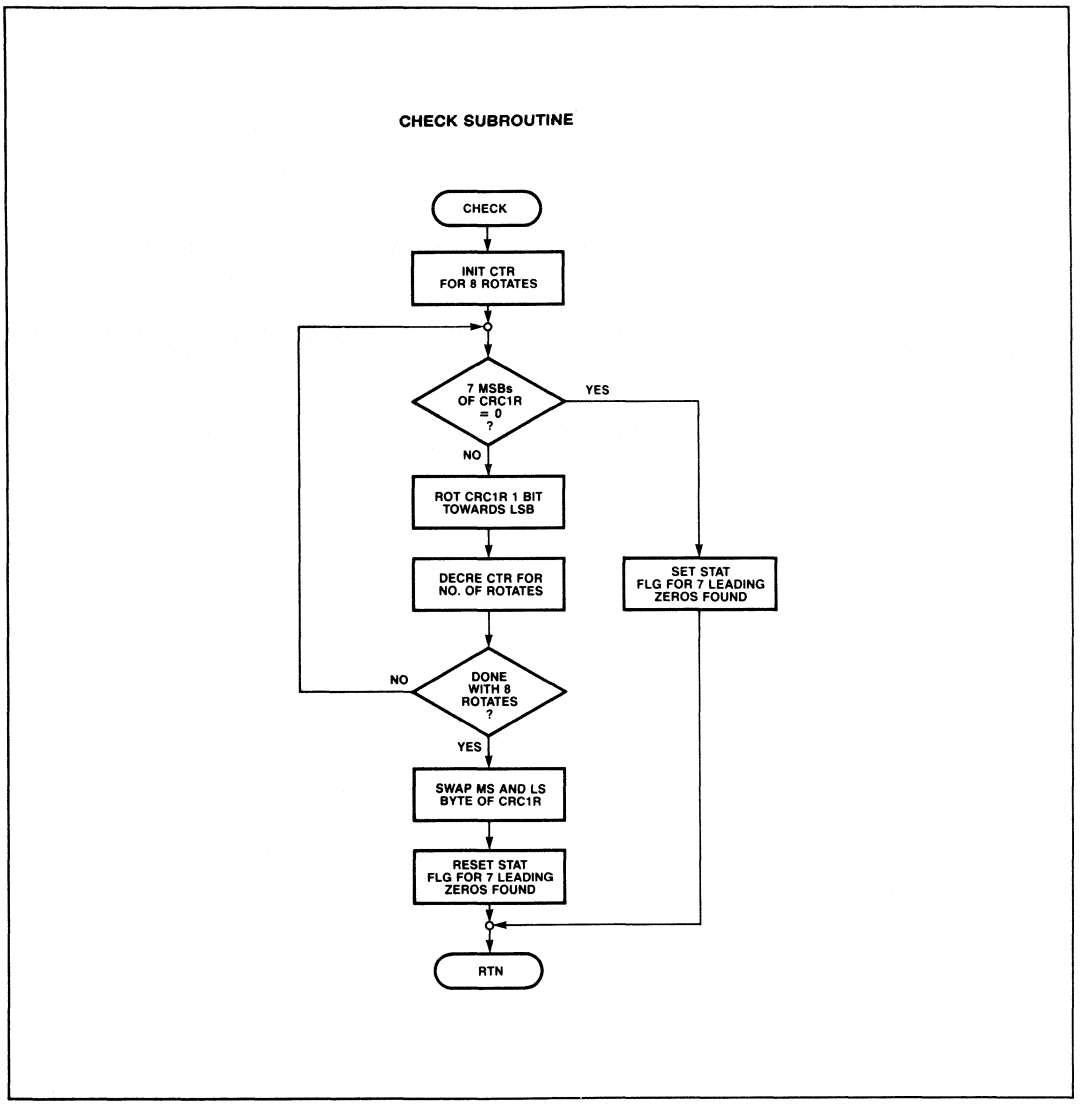
FLOWCHARTS


ERROR CORRECTION ROUTINE:
ECC
GEN CRC2R w/00₁₆ DATA INPUT
CRC2R ≠ 0 CRC1R = 0?
Yes
LD CRC2R INTO CRC1R
CHECK SUBR
7 MSBs OF CRC2R = 0?
No
Yes
GEN CRC2R w/00₁₆ DATA INPUT
CRC2R ≠ 0 CRC1R = 0?
LD CRC2R INTO CRC1R
CHECK SUBR
GEN CRC1R & CRC2R w/00₁₆ DATA INPUT
CRC1R ≠ 0 CRC2R = 0?
7 MSBs OF CRC2R = 0?
SET UNCORREC ERR FLAG
SET CRC ERR STATUS FLAG
ERROR
CHECK SUBR
7 MSBs OF CRC1R = 0?
SET CRC ERR FLAG
ERROR
CRC1R & CRC2R REMAINDERS ≠ 0?
CRC1R MS BYTE = 0?
CRC2R LS BYTE = 0?
LD CRC2R MS BYTE INTO CRC1R MS BYTE
CHECK SUBR
7 MSBs OF CRC1R = 0?
SET CRC ERR STATUS FLAG
ERROR
RESTORE CRC1R MS BYTE TO 0
GEN CRC1R & CRC2R w 00₁₆ DATA INPUT
CRC1R ≠ 0 CRC2R = 0?
INITIALIZE ERROR LOCATION COUNTER FOR 256-BYTE SECTOR
GEN CRC1R & CRC2R w/00₁₆ DATA INPUT
LS BYTE OF CRC1R = LS BYTE OF CRC2R?
CK FOR ERR IN LAST BYTE OF SECTOR?
MS BYTE OF CRC1R = MS BYTE OF CRC2R?
MS BYTE OF CRC2R = 0?
CHECK SUBR
7 MSBs OF CRC1R = 0?
CHECK SUBR
7 MSBs OF CRC1R = 0?
DECREMENT ERROR LOCATION COUNTER
DECREMENT ERROR LOCATION COUNTER
DONE w/REV GEN 256 BYTES?
ADDR 1ST ERROR DATA BYTE IN BUF
EX-OR LS BYTE OF CRC2R w 1ST ERROR DATA BYTE
SET UNCORREC ERROR FLAG
ERROR
INCREMENT ERROR LOCATION COUNTER
1ST ERROR IN LAST DATA BYTE OF SECTOR?
ADDR 2ND ERR DATA BYTE IN BUF
EX-OR MS BYTE OF CRC2R w 2ND ERR DATA BYTE
END
CHECK SUBR
SET UNCORREC ERROR FLAG
7 MSBs OF CRC1R = 0?
SET CRC ERROR STATUS FLAG
ERROR

FLOWCHARTS (Continued)



FLOWCHARTS (Continued)



Appendix B
PROGRAM LISTING

The error correction scheme described in this application note has been implemented in software as shown in the following program listing. This error correction program works with the existing hardware and software of the demonstration floppy disk controller as described in the Signetics 8X330 Floppy Disk Controller Manual. The first part of the error correction program is a software patch to read and write the two CRC2F check bytes for the demonstration controller program. The main error correction routine is located in the second half of the program. The error correction program is designed to work with the IBM system 34 data track format (256 data bytes per sector)—see illustration that follows the program listing. The only modification to the IBM system 34 data tract format is to add two more CRC check bytes to the end of the data field. With slight modifications, this program can also be used for other data track formats. For testing purposes additional coding has been added to the program to print out the contents of both CRC1R and CRC2R after each 8-bit reverse generation cycle. This section of codes can be deleted for the final program.

```
FILE: ECC330   OUTPUT   A   SIGNETICS CCRPORATICN                                   PAGE 001

1          PROG ECC330                 MICRCCONTROLLER CRCSS ASSEMBLER VER 2.0          PAGE    1

                                   PRCG ECC330
                              ****************************************************************
                              *                                                              *
                              *  ERROR CORRECTION PROGRAM.                     AUGUST, 1981. *
                              *                                                              *
                              *  THIS PROGRAM WORKS WITH THE THE HARDWARE AND SOFTWARE OF THE *
                              *  DEMONSTRATION CONTROLLER AS DESCRIBED IN THE SIGNETICS 8X330 FLOPPY *
                              *  DISK CONTROLLER MANUAL.  FOUR CRC CHECK BYTES, DERIVED FROM TWO   *
                              *  SEPARATE 16-BIT POLYNOMIALS, ARE USED FOR THE DATA FIELD.   *
                              *  CRC1F= X(16)+X(12)+X(5)+1                                   *
                              *  CRC2F= X(16)+X(13)+X(12)+X(10)+X(8)+X(6)+X(5)+X(4)+1        *
                              *                                                              *
                              ****************************************************************
                              *
                                   LIST M,S,O
                              *    OBJ  R,B
               124 7 0        CRCM     RIV 124H,7,8               *M.S. BYTE OF CRC2F
               124 7 0        MCRC2F RIV 124H,7,8                 *M.S. BYTE OF CRC2F
               115 7 0        CRCL     RIV 115H,7,8               *L.S. BYTE GF CRC2F
               115 7 0        LCRC2F RIV 115H,7,8                 *L.S. BYTE OF CRC2F
               120 7 0        MCRC1R RIV 120H,7,8                 *M.S. BYTE OF CRC1R
               120 0 1        MOCRC1R RIV 120H,0,1                *BIT 0 CF CRC1R M.S. BYTE
               120 6 1        M6CPC1R RIV 120H,6,1                *BIT 6 OF CRC1R M.S. BYTE
               120 7 1        M7CRC1R RIV 120H,7,1                *BIT 7 CF CRC1R M.S. BYTE
               121 7 0        LCRC1R RIV 121H,7,8                 *L.S. BYTE CF CRC1R
               121 0 1        LOCRC1R RIV 121H,0,1                *BIT 0 OF CRC1R L.S. BYTE
               121 6 1        L6CRC1R RIV 121H,6,1                *BIT 6 OF CRC1R L.S. BYTE
               121 7 1        L7CRC1R RIV 121H,7,1                *BIT 7 OF CRC1R L.S. BYTE
               115 7 0        MCRC2R RIV 115H,7,8                 *M.S. BYTE OF CRC2R
               115 0 1        MOCRC2R RIV 115H,0,1                *BIT 0 CF CRC2R M.S. BYTE
               115 6 1        M6CRC2R RIV 115H,6,1                *BIT 6 CF CRC2R M.S. BYTE
               115 7 1        M7CRC2R RIV 115H,7,1                *BIT 7 OF CRC2R M.S. BYTE
               124 7 0        LCRC2R RIV 124H,7,8                 *L.S. BYTE OF CRC2R
               124 0 1        LOCRC2R RIV 124H,0,1                *BIT 0 OF CRC2R L.S. BYTE
               124 6 1        L6CRC2R RIV 124H,6,1                *BIT 6 OF CRC2R L.S. BYTE
               124 7 1        L7CRC2R RIV 124H,7,1                *BIT 7 CF CRC2R L.S. BYTE
               132 7 0        ST330    RIV 132H,7,8               *8X330 STATUS REGISTER READ
               132 6 1        BYTRA    RIV 132H,6,1               *BYTE TRANSFER ACTIVE = 0
               132 1 1        CRCE     RIV 132H,1,1               *COMPUTE CRC=1
               114 6 2        MODU     RIV 114H,6,2               *10B= SINGLE DENSITY, 00B= DOUBLE DEN
               137 7 0        FDAT     RIV 137H,7,8               *DISK DATA REGISTER
               001 6 1        IDAT     RIV 1H,6,1
               001 5 1        ECRT     RIV 1H,5,1
               135 1 1        RDAT     RIV 135H,1,1
               122 7 0        G1       RIV 122H,7,8
               123 7 0        G2       RIV 123H,7,8
               124 7 0        G3       RIV 124H,7,8
               125 7 0        G4       RIV 125H,7,8
               126 7 0        G5       RIV 126H,7,8
               127 7 0        G6       RIV 127H,7,8
               136 7 0        SLEN     RIV 136H,7,8               *POINTER FOR 2ND LEVEL SUBROUTINE
               112 0 1        PHS10    RIV 112H,0,1               *CRC ERROR BIT IN PHY. REG. 1
                000005        DATA     EQU 5                      *DATA INPUT FOR CRC2F


FILE: ECC330   OUTPUT   A   SIGNETICS CORPORATION                                   PAGE 002

                000006        TEMP     EQU 6                      *TEMP REG.
1          PROG ECC330                 MICROCONTROLLER CRCSS ASSEMBLER VER 2.0          PAGE    2

                000027        LB       EQU 27H                    *LEFT BANK
                000037        RB       EQU 37H                    *RIGHT BANK
                000030        RB0      EQU 30H
                000031        RB1      EQU 31H
                000032        RB2      EQU 32H
                000033        RB3      EQU 33H
                000034        RB4      EQU 34H
                000035        RB5      EQU 35H
                000036        RB6      EQU 36H
                000037        RB7      EQU 37H
                000012        LF       EQU 012H                   *LINE FEED
                000015        CR       EQU 015H                   *CARRIAGE RETURN
                              *
                177777        POINT    SET -1
                              *
                              ****************************************************
                              *                                                  *
                              * PRINT BINARY MACRO.                              *
                              * PRINT THE CONTENT OF THE SELECT PORT IN THE RIGHT *
                              * BANK IN BINARY.                                  *
                              *                                                  *
                              ****************************************************
                              PRTBIN MACRO PRT1
                                       XMIT 1,AUX
                                       XMIT 377H-7,R6
                                       ORG 3,32
                              PRT1     MOVE R5,IVR
                                       XMIT '1',R4
                                       NZT RB,1,*+2
                                       XMIT '0',R4
                                       MOVE RB6,8,RB7
                                       CALL TTYWR
                                       ADD R6,R6
                                       NZT R6,PRT1
                                       ENDM
                              *
                              *
                              ******************************************************
                              *                                                    *
                              *  MACRO FOR SECOND LEVEL SUBROUTINE.                *
                              *  RETURN TO CALL2 BY JMP RTN2                       *
                              *                                                    *
                              ******************************************************
                              CALL2    MACRO SUBR2,LAB
                              *
                              LAB      SEL SLEN
                              POINT    SET POINT+1
                                       XMIT POINT,SLEN        *STORE RETURN JUMP INDEX IN SLEN
                                       JMP SUBR2              *GO TO 2ND LEVEL SUBRCUTINE
                                       ORG RTN2+POINT+3       *INSERT THE NEXT INSTR. IN RETURN TABLE
                                       JMP LAB+3              *2ND LEVEL SUBROUTINE RETURN JUMP
                                       ORG LAB+3
```

FILE: ECC330   OUTPUT   A   SIGNETICS CCRPCRATION   PAGE 003

```
                                   ENDM
                          ************************************************************
1       PRCG ECC330                 MICRCCCNTRCLLER CROSS ASSEMBLER VER 2.0          PAGE   3

                          *                                                          *
                          *  PATCH JMP INSTRUCTIONS IN THE 2K PROGRAM OF THE DEMONSTRATION  *
                          *  CONTROLLER TO THE READ AND WRITE CRC2F CHECK BYTES PROGRAM.    *
                          *                                                          *
                          ************************************************************
                                   ORG 1716H
        01716   7 07000            JMP AA1              *INITIALIZE CRC2F FOR A1,A1,A1,FB
                                   ORG 1746H
        01746   7 07010            JMP AA2              *INITIALIZE CRC2F FOR FB
                                   ORG 1772H
        01772   7 07020            JMP AA3              *CALCULATE CRC2F ON LINE FOR DATA FIELD
                                   CRG 1777H
        01777   7 07027            JMP AA4              *CALCULATE CRC2F FOR 8 BITS OF '0' INPUTS
                                   CRG 2001H
        02001   6 17132            SEL ST330
                                   ORG 2003H
        02003   7 07034            JMP AA5              *CALAULATE CRC2F FOR 8 BITS OF '0' INPUTS
                                   CRG 2010H
        02010   7 07061    P1      JMP PP2              *READ M.S. BYTE OF CRC1 REMAINDER
                                   CRG 2017H
        02017   7 07C64            JMP PP3              *READ 3 MORE BYTES OF CRC
                                   ORG 2255H
        02255   7 07116            JMP PP4              *INITIALIZE CRC2F FOR A1,A1,A1,FB OR FB
                                   CRG 2262H
        02262   7 07132            JMP PP5
                                   CRG 2270H
        02270   7 07140            JMP PP8
                          ******************************************************
                          *                                                    *
                          *  PATCH PRCGRAM TO READ CRC2F CHECK BYTES.          *
                          *                                                    *
                          ******************************************************
                                   ORG 7000H
        C7000   6 05114    AA1     XMIT 01001100B,R5    *INITIAL VALUE FCR CRC2F M.S. BYTE IN D.D
        07001   6 06013            XMIT 00001011B,R6    *INITIAL VALUE FCR CRC2F L.S. BYTE IN D.D
        07002   6 17124            SEL MCRC2F
        07003   0 05037            MOVE R5,MCRC2F       *INITIALIZE M.S. BYTE CF CRC2F IN D.D.
        07004   6 17115            SEL LCRC2F
        07005   0 06037            MOVE R6,LCRC2F       *INITIALIZE L.S. BYTE CF CRC2F IN D.D.
        07006   6 17137            SEL FDAT
        07007   7 01717            JMP 1717H
        07010   6 05177    AA2     XMIT 01111111B,R5    *INITIAL VALUE FOR CRC2F M.S. BYTE IN S.D
        C7011   6 06C33            XMIT C0011011B,R6    *INITIAL VALUE FCR CRC2F L.S. BYTE IN S.D
        07C12   6 17124            SEL MCRC2F
        07013   0 05037            MOVE R5,MCRC2F       *INITIALIZE M.S. BYTE CF CRC2F IN S.D.
        07014   6 17115            SEL LCRC2F
        07015   0 06037            MOVE R6,LCRC2F       *INITIALIZE L.S. BYTE CF CRC2F IN S.D.
        C7016   6 17137            SEL FDAT
        07017   7 01747            JMP 1747H
        C7020   6 00001    AA3     XMIT 1,AUX
        07021   1 01C01            ADC R1,R1
```

FILE: ECC330   OUTPUT   A   SIGNETICS CCRPCRATICN   PAGE 004

```
        07022   1 02CC2            ADD P2,R2
        07023   0 27005            MOVE LB,R5
        07024   6 11000            CALL CRC2F           *CALCULATE CRC2F FOR DATA FIELD
1       PROG ECC330              MICRCCCNTROLLER CRCSS ASSEMBLER VER 2.0            PAGE    4

        07025   7 06342
        07026   7 01774            JMP 1774H
        07027   6 17137    AA4     SEL FDAT
        07030   6 05000            XMIT 0,R5
        07031   6 11001            CALL CRC2F           *CALCULATE CRC2F WITH 8 BITS OF 0 INPUTS
        07032   7 06342                                 *DURING THE FIRST CRC CHECK BYTE TIME
        07033   7 02C00            JMP 2000H
        07034   6 17137    AA5     SEL FDAT
        07035   6 05000            XMIT 0,R5
        C7036   6 11002            CALL CRC2F           *CALCULATE CRC2F WITH 8 BITS OF 0 INPUTS
        07037   7 06342                                 *DURING THE SECOND CRC CHECK BYTE TIME
        07040   6 17132            SEL ST330
        07041   5 36100            NZT BYTRA,*-1        *WAIT FOR 3RD CRC CHECK BYTE
        C7042   6 17137            SEL FDAT
        07043   0 37005            MOVE FDAT,R5         *CRC2F INPUT = 3RD CRC CHECK BYTE
        07044   0 05004            MOVE R5,R4           *STORE 3RD CRC BYTE IN R4
        07045   6 11003            CALL CRC2F           *CALCULATE CRC2F
        07046   7 06342
        07047   6 17132            SEL CRCE
        07050   6 31100            XMIT 0,CRCE          *DISABLE CRC1F CALCULATION
        07051   6 17132            SEL ST330
        07052   5 36111            NZT BYTRA,*-1        *WAIT FOR 4TH CRC CHECK BYTE
        07053   6 17137            SEL FDAT
        07054   0 37005            MOVE FDAT,R5         *CRC2F INPUT = 4TH CRC CHECK BYTE
        07055   0 05C03            MOVE R5,R3           *STORE 4TH CRC BYTE IN R3
        07056   6 11004            CALL CRC2F           *CALCULATE CRC2F
        07057   7 06342
        07060   7 02004            JMP 2004H
        07061   6 17121    PP2     SEL LCRC1R
        C7062   0 00037            MOVE AUX,LCRC1R      *LOAD M.S. BYTE OF CRC1F INTO L.S. BYTE OF
        07063   7 02013            JMP 2013H            *CRC1R
        07064   6 17120    PP3     SEL MCRC1R
        07065   0 00037            MOVE AUX,MCRC1R      *LOAD L.S. BYTE OF CRC1F INTO M.S. BYTE OF
        07066   6 11005            CALL CRC1R           *CALCULATE CRC1R WITH 8 BITS CF 0 INPUT
        07067   7 06370
        07070   6 11006            CALL CRC1R           *BECAUSE 3RD AND 4TH CRC BYTES GO THRU CRC
        07071   7 06370
        07072   0 03000            MCVE R3,AUX          *
        07073   3 37037            XOR MCRC1R,MCRC1R    *
        07074   6 17121            SEL LCRC1R
        07075   0 04000            MOVE R4,AUX          *
        07076   3 37037            XOR LCRC1R,LCRC1R    *
        07077   6 01001            XMIT 1,R1            *R1=1 IF CRC1F REMAINDERS NOT EQ. ZERO
                                   CRG 5,32
        07100   5 37004            NZT LCRC1R,*+4       *M.S. BYTE OF MCRC1F REMAINDER=0? YES, *+4
        07101   6 17120            SEL MCRC1R
        07102   5 37C04            NZT MCRC1R,*+2       *L.S. BYTE OF MCRC1F REMAINDER=0? YES, *+2
        07103   6 01000            XMIT 0,R1            *R1=0 IF CRC1F REMAINCERS EQ. ZERO
        07104   6 00002            XMIT 10B,AUX
        C7105   6 17124            SEL MCRC2F
```

```
FILE: ECC330   CUTPUT   A   SIGNETICS CORPORATICN                                   PAGE 005

      206                          CRG 5,32
      207  07106  5 37012          NZT MCRC2F,LRA12        *M.S. BYTE OF MCRC2F REMAINDER=0? YES,GO
      208  07107  6 17115          SEL LCRC2F
      209  07110  5 37012          NZT LCRC2F,LRA12        *L.S. BYTE OF LCRC2F REMAINDER=0? YES,GO
1          PROG ECC330          MICROCONTROLLER CROSS ASSEMBLER VER 2.0                  PAGE   5

      210  07111  7 07113          JMP *+2                 *NO, GO TO *+2
      211  07112  1 01001   LRA12  ADD R1,R1               *R1=11B, BOTH CRC NCNZERO. R1=10B IF CRC2
      212  07113  5 01115          NZT R1,*+2              *CRC ERROR, GO TO ECC
      213  07114  7 02020          JMP 2020H
      214  07115  7 06C00          JMP ECC
      215                   *
      216                   *
      217                   ************************            ************************************
      218                   *                                                                  *
      219                   *  PATCH PROGRAM         TE CRC2F CHECK BYTES.                      *
      220                   *                                                                  *
      221                   ********************************************************************
      222                   *
      223  07116  6 02000   PP4    XMIT 0,R2
      224  07117  6 17114          SEL MCDU
      225  07120  6 05177          XMIT 01111111B,R5       *INITIAL VALUE OF CRC2F M.S. BYTE IN S.D.
      226  07121  6 06033          XMIT 00011011B,R6       *INITIAL VALUE OF CRC2F L.S. BYTE IN S.D.
      227  07122  5 36225          NZT MCDU,*+3
      228  07123  6 05114          XMIT 01001100B,R5       *INITIAL VALUE CF CRC2F M.S. BYTE IN D.D.
      229  07124  6 06013          XMIT 00001011B,R6       *INITIAL VALUE OF CRC2F L.S. BYTE IN D.D.
      230  07125  6 17124          SEL MCRC2F
      231  07126  0 05037          MOVE R5,MCRC2F          *PRE-INITIALIZE M.S. BYTE CF CRC2F
      232  07127  6 17115          SEL LCRC2F
      233  07130  0 06037          MOVE R6,LCRC2F          *PRE-INITIALIZE L.S. BYTE OF CRC2F
      234  07131  7 02256          JMP 2256H
      235  07132  0 27037   PP5    MOVE LB,RB
      236  07133  0 27005          MOVE LB,R5              *R5= DATA INPUT
      237  07134  6 11007          CALL CRC2F              *CALCULATE CRC2F FOR DATA FIELD
           07135  7 06342
      238  07136  6 00001          XMIT 1,AUX
      239  07137  7 02263          JMP 2263H
      240  07140  6 01376   PP8    XMIT 376H,R1
      241  07141  6 17132          SEL ST330
      242  07142  5 36101          NZT BYTRA,*-1           *WRITE LAST DATA BYTE AND FIRST CRC BYTES
      243  07143  6 31100          XMIT 0,CRCE             *DISABLE CRC
      244  07144  6 17137          SEL FDAT
      245  07145  6 05000          XMIT 0,R5
      246  07146  6 11010          CALL CRC2F              *CALCULATE CRC2F WITH 8 BITS OF 0'S
           C7147  7 06342
      247  07150  6 00001          XMIT 1,AUX              *FCR 1ST AND 2ND CRC BYTE TIME
      248  07151  1 01001          ADC R1,R1
      249  07152  5 01141          NZT R1,PP8+1
      250  07153  6 17132          SEL ST330
      251  07154  5 36113          NZT BYTRA,*-1           *WRIT 2ND CRC BYTF
      252  07155  6 31101          XMIT 1,CRCE             *ENABLE CRC
      253  07156  6 17124          SEL MCRC2F
      254  07157  0 37000          MOVE MCRC2F,AUX
      255  C7160  6 17137          SEL FCAT
      256  07161  0 0C037          MOVE AUX,FCAT
```

```
FILE: ECC330   CUTPUT   A   SIGNETICS CORPORATICN                                   PAGE 006

      257  07162  6 17132   C200   SEL ST330
      258  07163  5 36122          NZT BYTRA,C200          *WRITE 3RD CRC BYTF
      259  C7164  6 17115          SEL LCRC2F
      260  07165  0 37000          MOVE LCRC2F,AUX
      261  07166  6 17137          SEL FDAT
1          PROG ECC330          MICROCCNTROLLER CROSS ASSEMBLER VER 2.0                  PAGE   6

      262  07167  0 00037          MOVE AUX,FDAT
      263  07170  6 01375          XMIT 375H,R1
      264  07171  6 05377          XMIT 377H,R5
      265  07172  6 17132   C202   SEL ST330
      266  07173  5 36132          NZT BYTRA,C202          *WRITE 4TH CRC BYTE + 2 BYTES OF FF
      267  07174  6 17137          SEL FDAT
      268  07175  0 05037          MOVE R5,FDAT
      269  07176  6 00001          XMIT 1,AUX
      270  07177  1 01001          ADD R1,R1
      271  07200  5 01172          NZT R1,C202
      272  07201  6 17132          SEL ST330
      273  07202  7 02300          JMP 2300H
      274                   *
      275                   *****************************************
      276                   *                                       *
      277                   *  2ND LEVEL SUBROUTINE RETURN TABLE    *
      278                   *                                       *
      279                   *****************************************
      280  07203  6 17136   RET2   SEL SLEN
      281          007203   RTN2   SET RET2
      282  07204  0 37000          MOVE SLEN,AUX
      283  07205  4 00206          XEC *+1(AUX)
      284                          CRG RTN2+23
      285                   ***********************************************************
      286                   *                                                         *
      287                   *  ERROR CORRECTION RCUTINE.                              *
      288                   *                                                         *
      289                   ***********************************************************
      290                          CRG 6000H
      291                   ECC    CALL2 PRINT,P100        *PRINT CRC1R AND CRC2R
      291                   +*
      291  06000  6 17136   +P100   SEL SLEN
      291          000000   +PCINT SET PCINT+1
      291  06001  6 37000   +      XMIT PCINT,SLEN          *STCRE RETURN JUMP INDEX IN SLEN
      291  06002  7 06234   +      JMP PRINT                *GO TO 2ND LEVEL SUBRCUTINE
      291                   +      ORG RTN2+POINT+3         *INSERT THE NEXT INSTR. IN RETURN TABLE
      291  07206  7 06003   +      JMP P100+3                *2ND LEVEL SUBROUTINE RETURN JUMP
      291                   +      ORG P100+3
      292  06003  6 11011          CALL CRC2R              *START REVERSE GENERATION
           06004  7 06407
      293                          CALL2 PRINT,P110        *PRINT CRC1R AND CRC2R
      293                   +*
      293  060C5  6 17136   +P110   SEL SLEN
      293          000001   +PCINT SET POINT+1
      293  06006  6 37001   +      XMIT POINT,SLEN          *STCRE RETURN JUMP INDEX IN SLEN
      293  06007  7 06234   +      JMP PRINT                *GO TO 2ND LEVEL SUBRCUTINE
      293                   +      CRG RTN2+POINT+3         *INSERT THE NEXT INSTR. IN RETURN TABLE
      293  07207  7 06010   +      JMP P110+3                *2ND LEVEL SUBROUTINE RETURN JUMP
```

```
FILE: ECC330   OUTPUT   A   SIGNETICS CORPORATION                                PAGE 007

      293                          +       CRG P110+3
      294  06010  6 00002                  XMIT 10B,AUX
      295  06011  3 01000                  XOR R1,AUX
      296  06012  5 00021                  NZT AUX,RG16          *CRC2 N.EQ TO 0 AND CRC1 =0? NO, JMP RG16
      297  06013  6 11012                  CALL TFR2T1           *YES, LOAD CRC2R INTO CRC1R
           06014  7 06466
1          PROG ECC330          MICROCONTROLLER CROSS ASSEMBLER VER 2.0          PAGE    7
      298  06015  6 11013                  CALL CHECK            *CHECK FOR 7 LEADING ZEROS
           06016  7 06434
      299  06017  5 06021                  NZT R6,RG16           *7 LEADING ZEROS FOUND? NO, GO TO RG16
      300  06020  7 06222                  JMP CRCERR            *YES , ERROR IN CRC2
      301  06021  6 11014           RG16   CALL CRC2R
           06022  7 06407
      302                                  CALL2 PRINT,P200        *PRINT CRC1R AND CRC2R
      302                          ++
      302  06023  6 17136          +P200    SEL SLEN
      302       000002             +PCINT   SET PCINT+1
      302  06024  6 37002          +        XMIT PCINT,SLEN        *STORE RETURN JUMP INDEX IN SLEN
      302  06025  7 06234          +        JMP PRINT              *GO TO 2ND LEVEL SUBROUTINE
      302                          +        ORG RTN2+PCINT+3       *INSERT THE NEXT INSTR. IN RETURN TABLE
      302  07210  7 06026          +        JMP P200+3              *2ND LEVEL SUBROUTINE RETURN JUMP
      302                          +        ORG P200+3
      303  06026  6 00002                  XMIT 10B,AUX
      304  06027  3 01000                  XOR R1,AUX
      305  06030  5 00040                  NZT AUX,RG24          *CRC2 N.EQ 0 AND CRC1 = 0? NO, JMP RG24
      306  06031  6 11015                  CALL TFR2T1           *YES, LOAD CRC2 INTO CRC1
           06032  7 06466
      307  06033  6 11016                  CALL CHECK            *LOOK FOR 7 L.S. BITS EQUAL TO ZEROS
           06034  7 06434
      308  06035  5 06037                  NZT R6,*+2            *7 LEADING ZEROS FOUND?
      309  06036  7 06222                  JMP CRCERR            *YES, ERROR IN CRC2
      310  06037  7 06233                  JMP UNCORR            *NO, UNCORRECTABLE ERROR
      311  06040  6 11017           RG24   CALL CRC1R
           06041  7 06370
      312  06042  6 11020                  CALL CRC2R
           06043  7 06407
      313                                  CALL2 PRINT,P300
      313                          ++
      313  06044  6 17136          +P300    SEL SLEN
      313       000003             +PCINT   SET PCINT+1
      313  06045  6 37003          +        XMIT PCINT,SLEN        *STORE RETURN JUMP INDEX IN SLEN
      313  06046  7 06234          +        JMP PRINT              *GO TO 2ND LEVEL SUBROUTINE
      313                          +        ORG RTN2+PCINT+3       *INSERT THE NEXT INSTR. IN RETURN TABLE
      313  07211  7 06047          +        JMP P300+3              *2ND LEVEL SUBROUTINE RETURN JUMP
      313                          +        ORG P300+3
      314  06047  6 00001                  XMIT 01B,AUX
      315  06050  3 01000                  XOR R1,AUX
      316  06051  5 00057                  NZT AUX,RG24P         *CRC2 EQ 0 & CRC1 N.EQ 0?  NO,GO TO RP24P
      317  06052  6 11021                  CALL CHECK            *YES, CHECK FOR 7 LEADING ZEROS
           06053  7 06434
      318  06054  5 06056                  NZT R6,*+2            *7 LEADING ZEROS FOUND?
      319  06055  7 06222                  JMP CRCERR            *YES, ERROR IN CRC1
      320  06056  7 06116                  JMP RG32              *NO
      321  06057  6 00003           RG24P  XMIT 11B,AUX
```

```
FILE: ECC330   OUTPUT   A   SIGNETICS CORPORATION                                PAGE 008

      322  06060  3 01000                  XOR R1,AUX
      323                                  ORG 14,32
      324  06061  5 00116                  NZT AUX,RG32          *BOTH CRC N. EQ TO 0? NO, GO TO RG32.
      325  06062  6 17120                  SEL MCRC1R            *YES, CHECK FOR ERROR SPAN CRC1 AND CRC2
      326  06063  7 06100                  ORG 16,32
      327  06100  5 37016                  NZT MCRC1R,RG32       *CRC1 M.S. BYTE =0? NO, ERROR IN DATA
      328  06101  6 17124                  SEL LCRC2R            *YES
1          PROG ECC330          MICROCONTROLLER CROSS ASSEMBLER VER 2.0          PAGE    8
      329  06102  5 37016                  NZT LCRC2R,RG32       *CRC2 L.S. BYTE =0? NO, ERROR IN DATA
      330  06103  6 17115                  SEL MCRC2R            *YES
      331  06104  0 37000                  MOVE MCRC2R,AUX
      332  06105  6 17120                  SEL MCRC1R
      333  06106  0 00037                  MOVE AUX,MCRC1R       *LOAD CRC2 M.S. BYTE INTO CRC1 M.S. BYTE
      334  06107  6 11022                  CALL CHECK            *CHECK FOR 7 L.S. BITS EQ. ZEROS
           06110  7 06434
      335  06111  5 06113                  NZT R6,*+2            *7 LEADING ZEROS FOUND? NO, ERROR IN DATA
      336  06112  7 06222                  JMP CRCERR            *YES, ERROR SPAN CRC1 AND CRC2.
      337  06113  6 00000                  XMIT C,AUX
      338  06114  6 17120                  SEL MCRC1R
      339  06115  0 00037                  MOVE AUX,MCRC1R       *RESTORE MCRC1R VALUE BEFORE CHECK SUBR
      340  06116  6 11023           RG32   CALL CRC1R
           06117  7 06370
      341  06120  6 11024                  CALL CRC2R            *AT THE BOUNTARY OF DATA AND CRC
           06121  7 06407
      342                                  CALL2 PRINT,P400
      342                          ++
      342  06122  6 17136          +P400    SEL SLEN
      342       000004             +PCINT   SET PCINT+1
      342  06123  6 37004          +        XMIT PCINT,SLEN        *STORE RETURN JUMP INDEX IN SLEN
      342  06124  7 06234          +        JMP PRINT              *GO TO 2ND LEVEL SUBROUTINE
      342                          +        ORG RTN2+PCINT+3       *INSERT THE NEXT INSTR. IN RETURN TABLE
      342  07212  7 06125          +        JMP P400+3              *2ND LEVEL SUBROUTINE RETURN JUMP
      342                          +        ORG P400+3
      343  06125  6 00001                  XMIT 01B,AUX
      344  06126  3 01000                  XOR R1,AUX
      345  06127  5 00135                  NZT AUX,RGDATA        *CRC1 N. EQ TO 0 AND CRC2 = 0? NO, JMP RG
      346  06130  6 11025                  CALL CHECK            *CHECK FOR 7 LEADING ZEROS
           06131  7 06434
      347  06132  5 06134                  NZT R6,*+2            *7 LEADING ZEROS FOUND?
      348  06133  7 06222                  JMP CRCERR            *YES, ERROR IN CRC1
      349  06134  7 06233                  JMP UNCORR            *NO, UNCORRECTIBLE ERROR
      350  06135  6 03000          RGDATA  XMIT 0,R3             *INITIALIZE BYTE COUNTER
      351  06136  6 11026          LOOP    CALL CRC1R            *REVERSE GENERATE CRC1R
           06137  7 06370
      352  06140  6 11027                  CALL CRC2R
           06141  7 06407
      353                                  CALL2 PRINT,P500
      353                          ++
      353  06142  6 17136          +P500    SEL SLEN
      353       000005             +PCINT   SET PCINT+1
      353  06143  6 37005          +        XMIT PCINT,SLEN        *STORE RETURN JUMP INDEX IN SLEN
      353  06144  7 06234          +        JMP PRINT              *GO TO 2ND LEVEL SUBROUTINE
      353                          +        ORG RTN2+PCINT+3       *INSERT THE NEXT INSTR. IN RETURN TABLE
      353  07213  7 06145          +        JMP P500+3              *2ND LEVEL SUBROUTINE RETURN JUMP
```

```
FILE: ECC330    OUTPUT    A    SIGNETICS CORPORATION                                          PAGE 009

                        +       CRG P500+3
 06145  6 17124                 SEL LCRC2R
 06146  0 37000                 MOVE LCRC2R,AUX
 06147  6 17121                 SEL LCRC1R
 06150  3 37000                 XOR LCRC1R,AUX            *COMPARE L.S. BYTE OF CRC1R AND CRC2R
 06151  5 00171                 NZT AUX,NCOMP             *COMPARE? NO,GO TO NCOMP
 C6152  6 17115                 SEL MCRC2R                *YES
 06153  5 03161                 NZT R3,NLDB               *R.G. FOR LAST DATA BYTE. NO, JMP NLDB
1    PROG ECC330                MICROCONTROLLER CROSS ASSEMBLER VER 2.0                 PAGE    9

                                CRG 14,32
 06154  5 37031                 NZT MCRC2R,NCOMP          *YES, M.S. BYTE OF CRC2R E.Q. 0?
 06155  6 11030                 CALL CHECK                *CHECK FOR 7 LEADING ZEROS
 06156  7 06434
 06157  5 06171                 NZT R6,NCOMP              *7 LEADING ZEROS FOUND? NO, JMP NCOMP
 06160  7 06175                 JMP CCRVTR                *YES, GO TO CORRECTION
 06161  0 37000         NLDB    MOVE MCRC2R,AUX           *NOT R.G. FOR LAST DATA BYTE
 06162  6 17120                 SEL MCRC1R
 06163  3 37000                 XOR MCRC1R,AUX            *COMPARE M.S. BYTE OF CRC1R AND CRC2R
 06164  5 00171                 NZT AUX,NCOMP             *COMPARE? NO, GO TO NCOMP
 06165  6 11031                 CALL CHECK                *CHECK FOR 7 LEADING ZEROS
 06166  7 06434
 06167  5 06171                 NZT R6,NCOMP              *7 LEADING ZEROS FOUND? NO, JMP NCOMP
 06170  7 06175                 JMP CCRVTR                *YES, GO TO CORRECTION
 06171  6 00377         NCOMP   XMIT -1,AUX
 06172  1 03003                 ADD R3,R3                 *DECREMENT BYTE COUNT
 06173  5 03136                 NZT R3,LOOP               *RG 256 BYTES? NO, GO TO LOOP
 06174  7 06233                 JMP UNCORR                *UNCORRECTABLE ERROR
 06175  6 04C15         CCRVTR  XMIT CR,R4
 06176  6 11032                 CALL TTYWR                *PRINT CR
 06177  7 06553
 06200  6 04012                 XMIT LF,R4
 06201  6 11033                 CALL TTYWR                *PRINT LF
 06202  7 06553
 06203  6 00377                 XMIT -1,AUX
 06204  1 03003                 ADD R3,R3
 06205  0 03007                 MOVE R3,IVL               *R3 EQ. 1ST LOCATION IN ERROR
 06206  6 17124                 SEL LCRC2R
 06207  0 37000                 MOVE LCRC2R,AUX
 C6210  3 27027                 XOR LB,LB                 *CORRECTION FOR FIRST BYTE
 06211  6 00001                 XMIT 1,AUX
 06212  1 03C03                 ADD R3,R3
 06213  5 03215                 NZT R3,*+2                *1ST ERROR IN LAST DATA BYTE? NO, GO TO *
 06214  7 06221                 JMP DONE                  *YES, GO TO DONE
 06215  0 03007                 MOVE R3,IVL               *ADDRESS BUFFER
 06216  6 17115                 SEL MCRC2R
 06217  0 37000                 MOVE MCRC2R,AUX
 06220  3 27027                 XOR LB,LB                 *CORRECTION IN 2ND BYTE
 06221  7 02C20         DONE    JMP 2020H
 06222  6 17112         CRCERR  SEL PHS10                 *ERROR IN CRC BYTES
 06223  6 30101                 XMIT 1,PHS10              *SET CRC ERROR FLAG
 06224  6 04015                 XMIT CR,R4
 06225  6 11034                 CALL TTYWR                *PRINT CR
 06226  7 06553
 06227  6 04012                 XMIT LF,R4
```

```
FILE: ECC330    OUTPUT    A    SIGNETICS CORPORATION                                          PAGE 010

 C6230  6 11C35                 CALL TTYWR                *PRINT LF
 06231  7 06553
 06232  7 02020                 JMP 2020H
 06233  7 02027         UNCORR  JMP 2027H                 *UNCORRECTABLE ERROR
                        *
                        *
                        ***********************************************************
                        *                                                         *
                        *  2ND LEVEL SUBROUTINE TO PRINT CONTENTS OF CRC1R AND CRC2R.  *
1    PROG ECC330                MICROCONTROLLER CROSS ASSEMBLER VER 2.0                 PAGE   10

                        *                                                         *
                        ***********************************************************
 06234  6 04015         PRINT   XMIT CR,R4
 06235  6 11036                 CALL TTYWR                *PRINT CR
 06236  7 06553
 06237  6 04012                 XMIT LF,R4
 06240  6 11037                 CALL TTYWR                *PRINT LF
 06241  7 06553
 06242  6 05120                 XMIT MCRC1R,R5
                                PRTBIN PRT2               *PRINT M.S. BYTE OF CRC1R. M.S. BIT ON L.
 06243  6 0C001         +       XMIT 1,AUX
 06244  6 06370         +       XMIT 377H-7,R6
                        +       CRG 3,32
 06245  0 05017         +PRT2   MOVE R5,IVR
 06246  6 04061         +       XMIT '1',R4
 06247  5 37111         +       NZT RB,1,*+2
 06250  6 04060         +       XMIT '0',R4
 06251  0 36037         +       MOVE RB6,8,RB7
 06252  6 11040         +       CALL TTYWR
 06253  7 06553
 06254  1 06006         +       ADD R6,R6
 06255  5 06245         +       NZT R6,PRT2
 06256  6 04040                 XMIT ' ',R4
 06257  6 11041                 CALL TTYWR                *PRINT SPACE BETWEEN M.S. AND L.S. BYTE O
 06260  7 06553
 06261  6 05121                 XMIT LCRC1R,R5
                                PRTBIN PRT3               *PRINT L.S. BYTE OF CRC1R. M.S. BIT ON L.
 06262  6 00001         +       XMIT 1,AUX
 06263  6 06370         +       XMIT 377H-7,R6
                        +       CRG 3,32
 06264  0 05017         +PRT3   MOVE R5,IVR
 06265  6 04061         +       XMIT '1',R4
 06266  5 37130         +       NZT RB,1,*+2
 06267  6 04060         +       XMIT '0',R4
 06270  0 36037         +       MOVE RB6,8,RB7
 06271  6 11042         +       CALL TTYWR
 06272  7 06553
 06273  1 06006         +       ADD R6,R6
 06274  5 06264         +       NZT R6,PRT3
 06275  6 04040                 XMIT ' ',R4               *PRINT 2 SPACES BETWEEN CRC1R AND CRC2R
 06276  6 11043                 CALL TTYWR
 06277  7 06553
 06300  6 04040                 XMIT ' ',R4
 06301  6 11044                 CALL TTYWR
```

```
FILE: ECC330    OUTPUT    A    SIGNETICS CORPORATION                                    PAGE 011

          06302   7 06553
     425  06303   6 05115                XMIT MCRC2R,R5
     426                                 PRTBIN PRT4              *PRINT M.S. BYTE OF CRC2R. M.S. BIT ON L.
     426  06304   6 00001          +      XMIT 1,AUX
     426  06305   6 06370          +      XMIT 377H-7,R6
     426                           +      CRG 3,32
     426  06306   0 05017          +PRT4  MOVE R5,IVR
     426  06307   6 04061          +      XMIT '1',R4
     426  06310   5 37112          +      NZT R6,1,*+2
     426  06311   6 04060          +      XMIT '0',R4
1         PROG ECC330                  MICROCONTROLLER CROSS ASSEMBLER VER 2.0                  PAGE   11

     426  06312   0 36037          +      MOVE RB6,8,RB7
     426  06313   6 11045          +      CALL TTYWR
          06314   7 06553
     426  06315   1 06006          +      ADD R6,R6
     426  06316   5 06306          +      NZT R6,PRT4
     427  06317   6 04040                 XMIT ' ',R4
     428  06320   6 11046                 CALL TTYWR               *PRINT SPACE BETWEEN M.S. AND L.S. BYTE O
          06321   7 06553
     429  06322   6 05124                 XMIT LCRC2R,R5
     430                                 PRTBIN PRT5              *PRINT L.S. BYTE OF CRC2R. M.S. BIT ON L.
     430  06323   6 00001          +      XMIT 1,AUX
     430  06324   6 06370          +      XMIT 377H-7,R6
     430                           +      CRG 3,32
     430  06325   0 05017          +PRT5  MOVE R5,IVR
     430  06326   6 04061          +      XMIT '1',R4
     430  06327   5 37131          +      NZT RB+1,*+2
     430  06330   6 04060          +      XMIT '0',R4
     430  06331   0 36037          +      MOVE RB6,8,RB7
     430  06332   6 11047          +      CALL TTYWR
          06333   7 06553
     430  06334   1 06006          +      ADD R6,R6
     430  06335   5 06325          +      NZT R6,PRT5
     431  06336   6 04015                 XMIT CR,R4
     432  06337   6 11050                 CALL TTYWR               *PRINT CR
          06340   7 06553
     433  06341   7 07203                 JMP RTN2                *END OF THE PRINT ROUTINE
     434                            *
     435                            *
     436                            *****************************************************
     437                            *                                                   *
     438                            *                                                   *
     439                            *  CRC2F= X(16)+X(13)+X(12)+X(10)+X(8)+X(6)+X(5)+X(4)+1  *
     440                            *  THE CRC GENERATOR SHIFTS LEFT.                   *
     441                            *  BIT 0 OF CRCM IS THE X(15) TERM.                 *
     442                            *  BIT 7 OF CRCL IS THE X(0) TERM.                  *
     443                            *  DATA= ONE BYTE OF DATA INPUT.                    *
     444                            *  TEMP= TEMPORARY STORAGE.                         *
     445                            *                                                   *
     446                            *****************************************************
     447                            *
     448  06342                     CRCC CRC2F
     449  06343   6 17124           START  SEL  CRCL
     450  06344   0 37000                  MOVE CRCL,AUX

FILE: ECC330    OUTPUT    A    SIGNETICS CORPORATION                                    PAGE 012

     451  06345   3 32000                 XOR  DATA,AUX            *OPERATION A
     452  06346   0 00037                 MOVE AUX,CRCM
     453  06347   3 34400                 XOR  RB4,5,AUX           *OPERATION B
     454  06350   3 33400                 XOR  RB3,4,AUX           *OPERATION C
     455  06351   0 00037                 MOVE AUX,CRCM
     456  06352   3 33433                 XOR  RB3,4,RB3           *OPERATION D
     457  06353   3 32332                 XOR  RB2,3,RB2           *OPERATION E
     458  06354   3 31231                 XOR  RB1,2,RB1           *OPERATION F
     459  06355   0 37006                 MOVE CRCA,TEMP
     460  06356   0 00037                 MOVE AUX,CRCM
     461  06357   6 17115                 SEL  CRCL
1         PROG ECC330                  MICROCONTROLLER CROSS ASSEMBLER VER 2.0                  PAGE   12

     462  06360   3 37000                 XOR  CRCL,AUX            *OPERATION K
     463  06361   0 06037                 MOVE TEMP,CRCL           *L.S. BYTE RESULT IN CRCLS
     464  06362   6 17134                 SEL  CRCM
     465  06363   3 34000                 XOR  RB3,AUX             *OPERATION G
     466  06364   3 34000                 XOR  RB4,AUX             *OPERATION H
     467  06365   3 35600                 XOR  RB5,6,AUX           *OPERATION I
     468  06366   3 31037                 XOR  RB1,RB7             *OPERATION J
     469  06367   0 00037                 MOVE AUX:2,CRCM          *OPERATION J. M.S.  BYTE RESULT IN CRCM.
     470  06370   7 06611                 RTN
     471                                  END  CRC2F
     472                            *
     473                            *
     474                            *****************************************************
     475                            *                                                   *
     476                            *                                                   *
     477                            *  CRC1R= X(16)+X(11)+X(4)+1                        *
     478                            *  REVERSE POLYNOMIAL OF THE CRC-CCITT              *
     479                            *  THIS CRC GENERATOR SHIFTS RIGHT WITH ZERO INPUT  *
     480                            *  NOTE: M.S. BYTE OF CRC1F BECOME L.S. BYTE OF     *
     481                            *        CRC1R.                                     *
     482                            *  BIT 7 OF MCRC1R IS THE X(15) TERM.               *
     483                            *  BIT 0 OF LCRC1R IS THE X(0) TERM.                *
     484                            *                                                   *
     485                            *****************************************************
     486                            *
     487  06370                     CRCC CRC1R
     488  06371   6 17120           START  SEL  MCRC1R
     489  06372   0 37C00                  MOVE MCRC1R,AUX
     490  06373   3 32332                  XOR  RB2,3,RB2           *OPERATION A
     491  06374   0 00037                  MOVE RB2,AUX
     492  06375   3 33433                  XOR  RB3,4,RB3           *OPERATION B
     493  06376   0 37C06                  MOVE MCRC1R,TEMP
     494  06377   0 00037                  MOVE AUX,MCRC1R
     495  06400   6 17100                  SEL  LCRC1R
     496  06401   0 37000                  MOVE LCRC1R,AUX
     497  06402   0 06037                  MOVE TEMP,LCRC1R         *L.S. BYTE RESULT IN CRCLS
     498  06403   6 17120                  SEL  MCRC1R
     499  06404   3 34200                  XOR  RB3:2,AUX           *OPERATION C
     500  06405   3 34000                  XOR  RB4,AUX             *OPERATION D
     501  06406   0 00037                  MOVE AUX:4,MCRC1R        *OPERATION D. M.S.  BYTE RESULT IN CRCM.
     502  06407   7 06611                  RTN
     503                                   END  CRC1R
```

```
FILE: ECC330   OUTPUT   A   SIGNETICS CORPORATION                                  PAGE 013

                            *
                            *
                            *****************************************************
                            *                                                   *
                            *  CRC2R= X(16)+X(12)+X(11)+X(10)+X(8)+X(6)+X(4)+X(3)+1 *
                            *  REVERSE POLYNOMIAL OF THE CRC2R.                 *
                            *  THIS CRC GENERATOR SHIFTS RIGHT WITH ZERO DATA INPUT.*
                            *  NOTE: THIS BYTE CRC (CRC2R) BECOMES 1 BYTE OF CRC2A. *
                            *        IT IS DONE BY THE DECLARATION STATMENT.    *
                            *  BIT 7 OF MCRC2R IS THE X(15) TERM.               *
                            *  BIT 0 OF LCRC2R IS THE X(0) TERM.                *
1       PROG ECC330         *     MICROCONTROLLER CROSS ASSEMBLER VER 2.0          PAGE   13
                            *****************************************************
                            *
                            *
 514  06407                 START   PROC CRC2R
 515  06407  4 17115                SEL  MCRC2R
 516  06410  0 37000                MOVE MCRC2R,AUX
 517  06411  1 32313                XOR  RB3,*,RB3      *OPERATION A
 518  06412  1 32313                XOR  RB3,*,RB3      *OPERATION B
 519  06413  1 31231                XOR  RB1,2,RB1      *OPERATION C
 520  06414  0 37000                MOVE MCRC2R,AUX
 521  06415  1 32700                XOR  RB4,5,AUX      *OPERATION D
 522  06416  1 32700                XOR  RB4,*,AUX      *OPERATION E
 523  06417  1 31700                XOR  RB1,*,AUX      *OPERATION F
 524  06420  0 00006                MOVE AUX,TEMP
 525  06421  0 37000                MOVE MCRC2R,AUX
 526  06422  4 17124                SEL  LCRC2R
 527  06423  1 37000                XOR  LCRC2R,AUX     *OPERATION G
 528  06424  0 00037                MOVE TEMP,LCRC2P    *L.S. BYTE RESULT IN CRCLS
 529  06425  4 17115                SEL  MCRC2R
 530  06426  1 32000                XOR  RB1,AUX        *OPERATION H
 531  06427  1 32400                XOR  RB2,AUX        *OPERATION I
 532  06430  1 33700                XOR  RB5,*,AUX      *OPERATION J
 533  06431  1 31037                XOR  RB1,RB7        *OPERATION K
 534  06432  0 00237                MOVE AUX,2,MC4C2P   *OPERATION K. M.S.  BYTE RESULT IN CRCM.
 535  06433  7 06611                RTN
 536                                 END  CRC2R
 542                         *
 543                         *
 544                         *****************************************************************
 545                         *                                                               *
 546                         *  CHECK SUBROUTINE                                             *
 547                         *   1) CHECK 7 M.S. BITS OF CRC1R FOR ZEROS           *         *
 548                         *   2) ROTATE CRC1R 1 BIT TOWARDS L.S. BIT                      *
 549                         *   3) IF THE NUMBER OF ROTATE IS 8, THEN EXIT, OTHERWISE GO TO 1 *
 550                         *  IF 7 LEADING ZEROS ARE FOUND, SET R6 TO ZERO.                *
 551                         *                                                               *
 552                         *****************************************************************
 553                         *
 554  06434                         PROC CHECK
 555  06434  6 02370         SHIFT  XMIT 377H-7,R2          *YES, LOAD COUNT OF 8 SHIFTS
 556  06435  6 17120         LOOP1  SEL  MCRC1R
```

```
FILE: ECC330   OUTPUT   A   SIGNETICS CORPORATION                                  PAGE 014

 557  06436  7 06440                ORG 4,32
 558  06440  5 37703                NZT MCRC1R,7,**3       *7 M.S.B. OF CRC1R EQ. 0?
 559  06441  6 06C00                XMIT 0,R6               *YES. SET R6 TO 0.
 560  06442  7 06465                JMP ECHECK
 561  06443  0 30100                MOVE MQCRC1,1,AUX       *NO, ROTATE CRC REG. 1 BIT TOWARDS L.S.B.
 562  06444  0 37736                MOVE M7CRC1,7,M6CRC1    *
 563  06445  6 17121                SEL  LCRC1R             *
 564  06446  0 30106                MOVE L0CRC1,1,TEMP      *
 565  06447  0 37736                MOVE L7CRC1,7,L6CRC1    *
 566  06450  0 00137                MOVE AUX,L7CRC1         *
 567  06451  6 17120                SEL  MCRC1R
 568  06452  0 06137                MOVE TEMP,1,M7CRC1
 569  06453  6 00001                XMIT 1,AUX
1       PROG ECC330                MICROCONTROLLER CROSS ASSEMBLER VER 2.0          PAGE   14
 570  06454  1 02002                ADD R2,R2
 571  06455  5 02035                NZT R2,LOOP1            *DONE WITH 8 SHIFTS? NO, GO TO LOOP1
 572  06456  0 37006                MOVE MCRC1R,TEMP        *RESTORE CONTENTS OF CRC1R BEFORE ENTRY
 573  06457  6 17121                SEL  LCRC1R             *OF THIS SUBROUTINE BY SWAPPING THE
 574  06460  0 37000                MOVE LCRC1R,AUX         *MCRC1R AND LCRC1R BYTES.
 575  06461  0 06037                MOVE TEMP,LCRC1R        *
 576  06462  6 17120                SEL  MCRC1R             *
 577  06463  0 00037                MOVE AUX,MCRC1R         *
 578  06464  6 06377                XMIT 377H,R6            *SET R6 TO ALL ONES IF NO 7 LEADING 0
 579  C6465  7 06611         ECHECK RTN
 580                                 END CHECK
 581                         *
 582                         *
 583                         *****************************************************************
 584                         *                                                               *
 585                         *  TFR2T1 SUBROUTINE.                                           *
 586                         *  THIS SUBROUTINE WILL LOAD THE CONTENTS OF CRC2R INT CRC1R.    *
 587                         *                                                               *
 588                         *****************************************************************
 589                         *
 590  06466                         PROC TFR2T1
 591  06466  6 17115                SEL  MCRC2R
 592  06467  0 37000                MOVE MCRC2R,AUX
 593  06470  6 17120                SEL  MCRC1R
 594  06471  0 00037                MOVE AUX,MCRC1R
 595  06472  6 17124                SEL  LCRC2R
 596  06473  0 37000                MOVE LCRC2R,AUX
 597  06474  6 17121                SEL  LCRC1R
 598  06475  0 00037                MOVE AUX,LCRC1R
 599  06476  7 06611                RTN
 600                                 END TFR2T1
 601                         *
 602                         *****************************************
 603                         *                                       *
 604                         * TTYRE SUBRCUTINE                      *
 605                         * READ CRT INPUT AND STORE 7 BITS ASCII IN R4 *
 606                         * G5= NO. OF DATA BITS                  *
 607                         * G1= L.S. TIME DELAY COUNTER           *
 608                         * G2= M.S. TIME DELAY COUNTER           *
 609                         * R4= ASCII CRT INPUT                   *
```

```
FILE: ECC330   OUTPUT   A   SIGNETICS CORPORATION                                  PAGE 015

                                * DELAY IN G1 AND G2= 1 DATA BIT TIME.          *
                                * BIT TIME EQUATION = (((2XG1)+8)XG2+15)X559 NS  *
                                *                                               *
                                *************************************************
                                *
                                * BIT TIME=416.66US FOR 2400 BAUD.  BIT TIME=9.09MS FOR 110 BAUD
                                * BIT TIME=3.33MS FOR 300 BAUD.
                                      PROC TTYRE
 06477                                SEL ECRT
 06477  6 17001                      XMIT 377H,R4            *CLEAR BUFFER REGISTER
 06500  6 04377                      XMIT OH,ECRT            *ENABLE CRT
 06501  6 35100                      ORG 2,32
                                T1    SEL RDAT
 06502  6 17135                      NZT RDAT,T1             *STOP BIT?  YES, GO TO T2
 06503  5 31102
1          PROG ECC330          MICROCONTROLLER CROSS ASSEMBLER VER 2.0                  PAGE   15

 06504  6 17135                T2    SEL RDAT
 06505  5 31107                      NZT RDAT,T3             *START BIT?  YES, GO TO T3
 06506  7 06504                      JMP T2                  *NO, GO TO T2
 06507  6 17126                T3    SEL G5
 06510  6 00370                      XMIT 377H-7,AUX
 06511  0 00037                      MOVE AUX,G5             *LOAD COUNT OF 8.
 06512  6 00331                      XMIT 331H,AUX           **DELAY FOR ONE AND A HALF BIT TIME
 06513  6 17123                T6    SEL G2                  *AUX=331H FOR 300 BAUD
 06514  0 00037                      MOVE AUX,G2             *LOAD M.S. TIME DELAY COUNTER
 06515  7 06534                      JMP T5
 06516  6 17135                T11   SEL RDAT
 06517  0 31100                      MOVE RDAT,AUX
 06520  3 04104                      XCR  R4(1),R4           *STORE DATA BIT IN R4
 06521  6 00001                      XMIT 1,AUX
 06522  6 17126                      SEL G5
 06523  1 37037                      ADD G5,G5               *INCR. COUNT OF DATA BITS
 06524  6 00346                      XMIT 346H,AUX           **DELAY ONE DATA BIT TIME
 06525  7 06513                      JMP T6                  *AUX=346H FOR 300 BAUD
 06526  6 17001                T12   SEL ECRT
 06527  6 35101                      XMIT 1H,ECRT            *DISABLE CRT
 06530  6 00376                      XMIT 376H,AUX
 06531  2 04004                      AND R4,R4               *MASK 7 M.S. BITS
 06532  0 04104                      MOVE R4(1),R4           *ROTATE RIGHT ONE PLACE
 06533  7 06611                      RTN
 06534  6 17122                T5    SEL G1
 06535  6 00004                      XMIT 4H,AUX             **AUX=4 FOR 300 BAUD
 06536  0 00037                      MOVE AUX,G1
 06537  6 00001                      XMIT 1,AUX
                                      CRG 2,32
 06540  1 37037                T4    ADD G1,G1               *START TIME DELAY FOR 1 DATA BIT TIME
 06541  5 37000                      NZT G1,T4
 06542  6 17123                      SEL G2
 06543  1 37037                      ADD G2,G2
                                      CRG 3,32
 06544  5 37006                      NZT G2,T100             *END OF TOME DELAY?  NO, GO BACK TO T5
 06545  7 06547                      JMP *+2                 *YES, END OF 1 DATA BIT TIME DELAY
 06546  7 06534                T100  JMP T5
 06547  6 17126                      SEL G5
 06550  5 37012                      NZT G5,*+2              *DONE WITH ALL 8 DATA BIT?  NO,GO TO T11


FILE: ECC330   OUTPUT   A   SIGNETICS CORPORATION                                  PAGE 016

 06551  7 06526                      JMP T12                 *YES, GO TO T12
 06552  7 06516                      JMP T11
                                      END TTYRE
                                *
                                *
                                *************************************************
                                *                                               *
                                *                                               *
                                * TTYWR SUBROUTINE                              *
                                * OUTPUT ASCII DATA IN R4 TO CRT                *
                                * G2= M.S. TIME DELAY COUNTER                   *
                                * G1= L.S. TIME DELAY COUNTER                   *
                                * G5= COUNT OF BIT TIME                         *
                                * R4= ASCII DATA                                *
                                *                                               *
1          PROG ECC330          MICROCONTROLLER CROSS ASSEMBLER VER 2.0                  PAGE   16

                                *                                               *
                                *************************************************
                                *
                                      PROC TTYWR
 06553                                SEL TDAT
 06553  6 17001                      XMIT 0H,TDAT            *START BIT
 06554  6 36100                      SEL G5
 06555  6 17126                      XMIT 377H-8,AUX
 06556  6 00367                      MOVE AUX,G5             *LOAD COUNT OF 9 BIT TIME
 06557  0 00037                T21   XMIT 346H,AUX           **DELAY ONE BIT TIME.
 06560  6 00346                      SEL G2                  *AUX=346H FOR 300 BAUD
 06561  6 17123                      MOVE AUX,G2             *LOAD M.S. TIME DELAY COUNTER
 06562  0 00037                      ORG 9,32
 06563  6 17122                T25   SEL G1                  *START OF DELAY LOOP
 06564  6 00003                      XMIT 3H,AUX             **AUX=3H FOR 300 BAUD
 06565  0 00037                      MOVE AUX,G1             *LOAD L.S. TIME DELAY COUNTER
 06566  6 00001                      XMIT 1,AUX
 06567  1 37037                T24   ADD G1,G1
 06570  5 37027                      NZT G1,T24
 06571  6 17123                      SEL G2
 06572  1 37037                      ADD G2,G2
 06573  5 37023                      NZT G2,T25
 06574  6 17126                      SEL G5                  *END OF DELAY LOOP
                                      ORG 3,32
 06575  5 37037                      NZT G5,C39
 06576  7 06611                      RTN
 06577  1 37037                C39   ADD G5,G5               *INCR. DATA BIT COUNT
                                      ORG 3,32
 06600  5 37002                      NZT G5,C36
 06601  7 06606                      JMP C37
 06602  6 17001                C36   SEL TDAT
 06603  0 04136                      MOVE R4,TDAT            *SHIFT OUT L.S.B. OF R4 TO TDAT
 06604  0 04104                      MOVE R4(1),R4           *RIGHT ROTATE ONE PLACE
 06605  7 06560                      JMP T21
 06606  6 17001                C37   SEL TDAT
 06607  6 36101                      XMIT 1,TDAT             *XMIT STOP BIT
 06610  7 06560                      JMP T21
                                      END TTYWR
```

```
FILE: ECC330   OUTPUT   A   SIGNETICS CORPORATION                          PAGE 017

      716                              END ECC330

  RETURN TABLE

          06611  4 11212
          06612  7 07026
          06613  7 07033
          06614  7 07040
          06615  7 07047
          06616  7 07060
          06617  7 07070
          06620  7 07072
          06621  7 07136
          06622  7 07150
1         PROG ECC330              MICROCONTROLLER CROSS ASSEMBLER VER 2.0   PAGE   17

          06623  7 06C05
          06624  7 06015
          06625  7 06017
          06626  7 06023
          06627  7 06033
          06630  7 06035
          06631  7 06042
          06632  7 06044
          06633  7 06054
          06634  7 06111
          06635  7 06120
          06636  7 06122
          06637  7 06132
          06640  7 06140
          06641  7 06142
          06642  7 06157
          06643  7 06167
          06644  7 06200
          06645  7 06203
          06646  7 06227
          06647  7 06232
          06650  7 06237
          06651  7 06242
          06652  7 06254
          06653  7 06261
          06654  7 06273
          06655  7 06300
          06656  7 06303
          06657  7 06315
          06660  7 06322
          06661  7 06334
          06662  7 06341

  ASSEMBLER ERRORS =    0
1         PROG ECC330              MICROCONTROLLER CROSS ASSEMBLER VER 2.0   PAGE   18
```

```
FILE: ECC330   OUTPUT   A   SIGNETICS CORPORATION                          PAGE 018

                          SYMBOL TABLE

 *  1

AA1     007000    AA2     007010    AA3     007020    AA4     007027
AA5     007034    AUX     000000    BYTRA   013261    C200    007162
C202    007172    CALL2   000015    CHECK   006434    CORVTR  006175
CR      000015    CRC1R   006370    CRC2F   006342    CRC2R   006407
CRCE    013211    CRCERR  006222    CRCL    011570    CRCM    012470
DATA    000035    DONE    006221    ECC     006000    ECC330  000000
ECRT    000151    FEAT    013770    G1      012270    G2      012370
G3      012470    G4      012570    G5      012670    G6      012770
IVL     000007    IVP     000017    L0CRC1  012101    L0CRC2  012401
L6CRC1  012161    L6CRC2  012461    L7CRC1  012171    L7CRC2  012471
LB      000027    LCRC1R  012170    LCRC2F  011570    LCRC2R  012470
LF      000012    LOOP    006136    LRA12   007112    M0CRC1  012001
M0CRC2  011501    M6CRC1  012061    M6CRC2  011561    M7CRC1  012071
M7CRC2  011571    MCRC1R  012070    MCRC2F  012470    MCRC2R  011570
MODU    011462    NCCMP   006171    NLDB    006161    OVF     000010
P1      002010    PHS10   011201    POINT   000005    PP2     007061
PP3     007064    PP4     007116    PP5     007132    PP8     007140
PRINT   006234    PRTBIN  000001    R0      000000    R1      000001
R11     000011    R2      000002    R3      000003    R4      000004
R5      000005    R6      000006    RA      000037    RB0     000030
RB1     000031    RB2     000032    RB3     000033    RB4     000034
RB5     000035    RB6     000036    RB7     000037    RDAT    013511
RET2    007203    RG16    006021    RG24    006040    RG24P   006057
RG32    006116    RGDATA  006135    RTN2    007203    SLEN    013670
ST330   013270    TCAT    000161    TEMP    000006    TFR2T1  006466
TTYRE   006477    TTYWR   006553    UNCCRR  006233

 *  2

P100    006000

 *  3

P110    006005

 *  4

P200    006023

 *  5

P300    006044

 *  6

P400    006122

 *  7

1
```

```
P50C    006142
* 8
PRT2    006245
* 9
PRT3    006264
* 10
PRT4    006306
* 11
PRT5    006325
* 12
START   006342
* 13
START   006370
* 14
START   006407
* 15
ECHECK  006465      LCCP1   006435      SHIFT   006434
* 16

* 17
T1      006502      T100    006546      T11     006516      T12     006526
T2      006504      T3      006507      T4      006540      T5      006534
T6      006513
* 18
C36     006602      C37     006606      C39     006577      T21     006560
T24     006567      T25     006563
1
```

Software support

MCCAP 8X300/8X305 CROSS ASSEMBLER PROGRAM

Originally published by Signetics January 1984

The MicroController Cross Assembler Program (MCCAP) has been developed to support the Signetics 8X300/8X305 MicroController. MCCAP provides many powerful features including macros, automatic subroutine handling, conditional assembly and extended instructions. These features significantly reduce the time required to compose and assemble MicroController programs. When combined with standard assembler features such as mnemonic op-codes and address labels, these extended features make MCCAP a powerful programming tool.

As input, MCCAP accepts source code written according to the rules presented in this manual. After assembling the source input, MCCAP produces an assembly listing and machine-readable object module.

MCCAP is written in ANSI standard FORTRAN IV and is available on the more popular timesharing services. MCCAP is also available as a fully supported product from Signetics for use on a user's in-house system.

MCCAP 8X300/8X305 CROSS ASSEMBLER PROGRAM

MCCAP, the crossassembler program for the 8X300 and 8X305 Micro-Controllers, is supplied as a 9-track magnetic tape containing FORTRAN IV source code for the crossassembler program. For compatibility with various computer systems, the tape is available in various combinations of density and data encoding. To order, use the following part numbers:

NUMBER	DENSITY	ENCODING
8X300 AS1-3SS	1600 BPI	ASCII
8X300 AS1-4SS	1600 BPI	EBCDIC

8X300/8X305 CROSS ASSEMBLER

INTRODUCTION

The 8X300AS2 runs on an Intel Intellec™ Microcomputer Development System with 64K memory under the contorl of ISIS II operating system.

The 8X300AS2 is composed of a two-pass crossassembler program and a PROM formatter overlay. Both programs are written entirely in Intel 8080 Assembly language, and are assembled on the Intel 8080/8085 Macro Assembler version 4.0, linked and located to execute in overlay.

It assembles both 8X300 and 8X305 programs. Also needed is at least one single or double density disk drive with the PROM formatter overlay always residing in drive zero.

The 8X300AS2 software is contained on three diskettes. Disks 1 and 2 contain the Single Density version and disk 3 contains the Double Density version. Both versions will be shipped when ordered under this part number.

Special purpose circuits

Signetics

8X01A/9401
CRC Generator/Checker

Product Specification

Logic Products

FEATURES
- **TTL inputs/outputs**
- **12MHz (Max) data rate**
- **Separate preset/reset controls**
- **SDLC specified pattern match (8X01A only)**
- **Automatic right justification**
- **Pin-for-pin compatibility and functionally identical with 8X01 (8X01A only)**
- **V_{CC} = 5V**
- **14-Pin DIP**

APPLICATIONS
- **Floppy and other disk systems**
- **Digital cassette and cartridge systems**
- **Data communication systems**

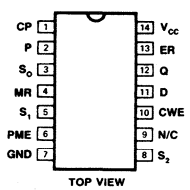
DESCRIPTION
The CRC Generator/Checker (8X01A or 9401) provides error-correction capabilities for digital systems that handle serial data. The two parts differ in that the 8X01A provides Synchronous Data Link Control (SDLC).

The serial data stream is divided by a selected polynomial; the remainder resulting from this algebraic process is transmitted at the end of the data stream as a Cyclic Redundancy Check Character (CRCC). At the receiving end, the same calculation is performed on the data. If the received message is error-free, the calculated remainder should satisfy a predetermined pattern. In most cases, the remainder is zero; however, where SDLC protocols (8X01A only) are used, the correct remainder is 1111000010111000 ($X^0 - X^{15}$).

Eight polynomials are provided and any of these can be selected via a 3-bit control bus. Popular polynomials, such as CRC-16 and CCITT are implemented and the one selected can be programmed to start with all zeroes or all ones. Right justification for polynomials of degree less than 16 is automatic.

8X01A & 9401 PACKAGE/PIN DESIGNATOR

N, F PACKAGE



TOP VIEW

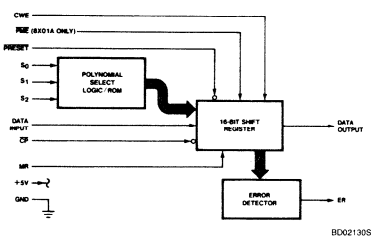
S_0, S_1, S_2	—	Polynomial Select Inputs
D	—	Data Inputs
$\overline{CP}$	—	Clock Input (Active On High-To-Low Transition)
CWE	—	Check Word Enable
$\overline{P}$	—	Preset Input (Active Low)
MR	—	Master Reset Input (Active High)
Q	—	Data Output
ER	—	Error Output (Active High)
$\overline{PME}$	—	Pattern Match Enable (Active Low) (8X01A ONLY)
N/C	—	No Connection

ORDER NUMBERS

N8X01A, N9401N

* 8X01A only; for 9401, pin 6 is not connected (N/C)

BLOCK DIAGRAM OF 8X01A & 9401



NOTE:
Refer to Truth Table on next page for selection of polynomials.

FUNCTIONAL OPERATION
8X01A and 9401
The CRC Generator/Checker circuit provides a means of detecting errors in a serial data communications environment. A binary message can be interpreted as a binary polynomial H(x). This polynomial can be divided by a generator polynomial P(x) such that H(x) = P(x) Q(x) + R(x) whereby Q(x) is the quotient and R(x) is the remainder. During transmission, the remainder is appended to the end of the message as check bits. For a given message, a unique remainder is generated. Hardware implementation of division is simply a feedback shift register with Exclusive-OR gating. Subtraction and addition in modulo 2 is implemented by the Exclusive-OR function. The number of shift register stages is equal to the degree of the divisor polynomial.

The accompanying truth table defines the polynomials implemented in the CRC circuit. Each polynomial can be selected via control inputs S_0, S_1 and S_2. To generate the check bits, the data stream is entered via the Data (D) input, using the high to low transition of the Clock ($\overline{CP}$) input. This data is gated with the most significant output (Q) of the shift register which, in turn, controls the exclusive OR gates. The Check Word Enable (CWE) must be held high while the data is being entered. After the last data bit is entered, the CWE is brought low and the check bits are shifted out of the register and appended to the data bits using external gating — see Check Word Generation diagram.

CRC Generator/Checker | 8X01A/9401

To check an incoming message for errors, both the data and check bits are entered through the "D" input with the CWE input held high. The 8X01A while not in the data path, monitors the message. After the last check bit is entered, in the 8X01A, the ERror output is made valid by a high-to-low transition of $\overline{CP}$. If no error is detected during the data transmission, all bits of the internal register are low and the ERror output is also low; if an error is detected, it is reflected by the bit pattern and the ERror output is high. The ERror output status remains valid until the next high-to-low transition of $\overline{CP}$ or until initialized by the preset ($\overline{P}$) or reset (MR) functions. The $\overline{PME}$ line must be high if the ERror output is used to indicate an all-zero result.

A high level applied to the Master Reset (MR) input asynchronously clears the shift register. A low level applied to the Preset (P) input asynchronously sets all bits to the appropriate state if the control-code inputs (S_0, S_1, and S_2) specify a 16-bit polynomial. In the case of check polynomials that are 8-or-12 bits in length, only the most significant 8-or-12 bits of the shift register are set; all remaining bits are cleared.

8X01A ONLY

For data communications using the Synchronous Data Link Control (SDLC) protocol, the 8X01A is preset to an all-ones configuration before any accumulation is done; this applies to both transmitting and receiving modes of operation. Using SDLC, the check sum shifted out of the 8X01A must be inverted.

During the receiving mode, a special pattern of 1111000010111000 ($X^0 - X^{15}$) is used in place of all-zeroes to check for a valid message. The Pattern Match Enable pin allows the user to select this option. If $\overline{PME}$ is low during the last bit time of the message, the ERror output is low providing the result matches the special pattern; if an error occurs, ER is high.

TRUTH TABLE

SELECT CODE			POLYNOMIAL	REMARKS
S_2	S_1	S_0		
L	L	L	$X^{16} + X^{15} + X_2 + 1$	CRC-16
L	L	H	$X^{16} + X^{14} + X + 1$	CRC-16 REVERSE
L	H	L	$X^{16} + X^{15} + X^{13} + X^7 + X^4 + X^2 + X^1 + 1$	
L	H	H	$X^{12} + X^{11} + X^3 + X^2 + X + 1$	CRC-12
H	L	L	$X^8 + X^7 + X^5 + X^4 + X + 1$	
H	L	H	$X^8 + 1$	LRC-8
H	H	L	$X^{16} + X^{12} + X^5 + 1$	CRC-CCITT
H	H	H	$X^{16} + X^{11} + X^4 + 1$	CRC-CCITT REVERSE

RECOMMENDED OPERATING CONDITIONS

PARAMETER		LIMITS			UNIT
		Min	Typ	Max	
V_{CC}	Supply voltage	4.75	5.0	5.25	V
$\overline{CP}$	Clock input	0		12	MHz

DC ELECTRICAL CHARACTERISTICS FOR 8X01A

PARAMETER	DESCRIPTION	TEST CONDITIONS[1]	LIMITS			UNIT
			Min	Typ	Max	
V_{IH}	Input high voltage		2.0			V
V_{IL}	Input low voltage				0.8	V
V_{IC}	Input clamp diode voltage	V_{CC} = Min, I_{IN} = −18mA		−0.9	−1.5	V
V_{OH}	Output high voltage	V_{CC} = Min, I_{OH} = −400μA	2.7	3.4		V
V_{OL}	Output low voltage	V_{CC} = Min, I_{OL} = 4.0mA		0.35	0.4	V
		V_{CC} = Min, I_{OL} = 8.0mA		0.45	0.5	V
I_{IL}	Input low current	V_{CC} = Max, V_{IN} = 0.4V		−0.22	−0.36	mA
I_{IH}	Input high current	V_{CC} = Max, V_{IN} = 2.7V			20	μA
I_{IH}	Max input current	V_{CC} = Max, V_{IN} = 7V			0.1	mA
I_{OS}	Output short circuit current	V_{CC} = Max, V_{OUT} = 0V[2]	−10		−42	mA
I_{CC}	Supply current	V_{CC} = Max, inputs open		60	110	mA

DC ELECTRICAL CHARACTERISTICS FOR 9401

PARAMETER	DESCRIPTION	TEST CONDITIONS[1]	LIMITS			UNIT
			Min	Typ	Max	
V_{IH}	Input high voltage	Guar. input high voltage	2.0			V
V_{IL}	Input low voltage	Guar. input low voltage			0.8	V
V_{IC}	Input clamp diode voltage	V_{CC} = Min, I_{IN} = −18mA		−0.9	−1.5	V
V_{OH}	Output high voltage	V_{CC} = Min, I_{OH} = −400μA	2.4	3.4		V
V_{OL}	Output low voltage	V_{CC} = Min, I_{OL} = 4.0mA		0.35	0.4	V
		V_{CC} = Min, I_{OL} = 8.0mA		0.45	0.5	V
I_{IL}	Input low current	V_{CC} = Max, V_{IN} = 0.4V		−0.22	−0.36	mA
I_{IH}	Input high current	V_{CC} = Max, V_{IN} = 2.7V		1.0	40	μA
		V_{CC} = Max, V_{IN} = 5.5V			1.0	mA
I_{OS}	Output short circuit current[2]	V_{CC} = Max, V_{OUT} = 0V	−15		−100	mA
I_{CC}	Supply current	V_{CC} = Max, inputs open		70	110	mA

NOTES:
1. Commercial — V_{CC}(MIN) = 4.75V; V_{CC}(MAX) = 5.25V.
2. No more than one output should be shorted at a time.

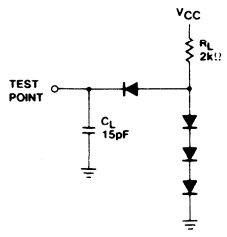
AC ELECTRICAL CHARACTERISTICS FOR 8X01A V_{CC} = 5V, T_A = +25°C

PARAMETER	DESCRIPTION	FROM	TO	TEST CONDITIONS	LIMITS			UNIT
					Min	Typ	Max	
f_{max}	Max clock freq				12			MHz
Pulse widths:								
t_w – $\overline{CP}$(L)	Clock low			See Figure 2	35			ns
t_w – $\overline{P}$(L)	Preset low			See Figure 3	35			ns
t_w – MR(H)	Master reset high			See Figure 4	35			ns
Set-up/hold times:								
t_s – D	Set-up time	Data	Clock		55			ns
t_s – $\overline{CWE}$(L)	Set-up time	CWE	Clock	See Figure 5	55			ns
t_h – D & CWE	Hold time	Data & CWE	Clock		0			ns
Propagation delay:								
$t_{PLH,PHL}$	Low-to-High and High-to-Low	$\overline{PRESET}$	Data output	See Figures 1, 2, & 3			55	ns
$t_{PLH,PHL}$	Low-to-High and High-to-Low	Master reset	Data output	See Figure 4			55	ns
$t_{PLH,PHL}$	Low-to-High and High-to-Low	$\overline{PRESET}$	Error output	See Figure 3			55	ns
$t_{PLH,PHL}$	Low-to-High and High-to-Low	Master reset	Error output	See Figure 4			55	ns
$t_{PLH,PHL}$	Low-to-High and High-to-Low	$\overline{CP}$	Data output	See Figure 2			55	ns
$t_{PLH,PHL}$	Low-to-High and High-to-Low	$\overline{CP}$	Error output	See Figure 2			55	ns
t_{REC}	Recovery time	Preset, MR	Clock	See Fig. 3 & 4	35			ns

AC ELECTRICAL CHARACTERISTICS FOR 9401 V_{CC} = 5V, T_A = +25°C

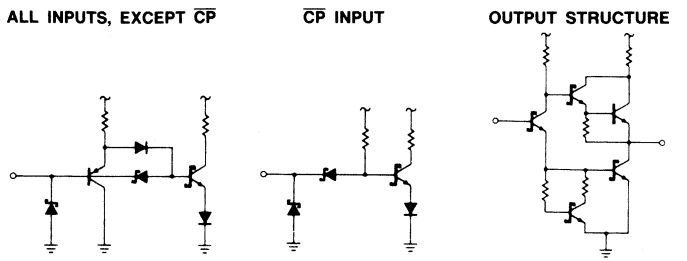
PARAMETER	DESCRIPTION	FROM	TO	TEST CONDITIONS	LIMITS			UNIT
					Min	Typ	Max	
f_{max}	Max clock freq				12	20		MHz
Pulse widths:								
t_w – $\overline{CP}$(L)	Clock low			See Figure 2	35			ns
t_w – $\overline{P}$(L)	Preset low			See Figure 3	40	30		ns
t_w – MR(H)	Master reset high			See Figure 4	35	25		ns
Set-up/hold times:								
t_s – D	Set-up time	Data	Clock		55	35		ns
t_s – CWE	Set-up time	CWE	Clock	See Figure 5	55	35		ns
t_h – D & CWE	Hold time	Data & CWE	Clock		0	–8		ns
Propagation delay:								
$t_{PLH,PHL}$	Low-to-High and High-to-Low	$\overline{PRESET}$	Data output	See Figures 1, 2, & 3		40	60	ns
$t_{PLH,PHL}$	Low-to-High and High-to-Low	Master reset	Data output	See Figure 4		30	55	ns
$t_{PLH,PHL}$	Low-to-High and High-to-Low	$\overline{PRESET}$	Error output	See Figure 3		40	60	ns
$t_{PLH,PHL}$	Low-to-High and High-to-Low	Master reset	Error output	See Figure 4		40	60	ns
$t_{PLH,PHL}$	Low-to-High and High-to-Low	$\overline{CP}$	Data output	See Figure 2		30	55	ns
$t_{PLH,PHL}$	Low-to-High and High-to-Low	$\overline{CP}$	Error output	See Figure 2		40	60	ns
t_{REC}	Recovery time	Preset, MR	Clock	See Fig. 3 & 4	35	25		ns

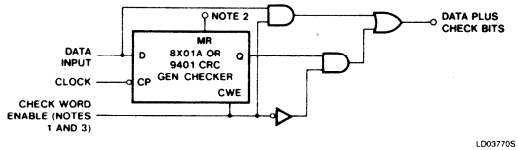
TEST CIRCUIT



NOTES:
1. Diodes are 1N3064 or equivalent.
2. C_L includes jig and probe capacitance.

INPUT/OUTPUT STRUCTURES

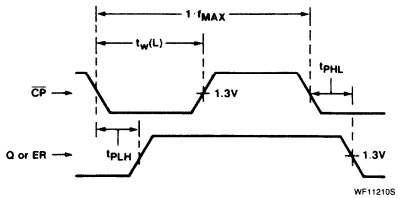




NOTES:
1. Check Word Enable is HIGH while data is being clocked: it is LOW during transmission of check bits.
2. The 8X01A (or 9401) must be RESET or PRESET before computation.
3. CRC check bits are generated and appended to data bits.

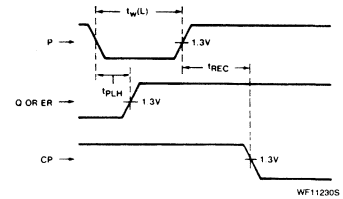
Figure 1. Check Word Generation

TEST CIRCUITS AND WAVEFORMS



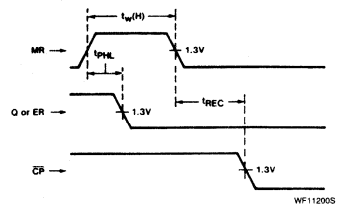
V_M = 1.3V for 74S; V_M = 1.5V for all other TTL familes.

Figure 2. Propagation Delay — $\overline{CP}$ to Q and $\overline{CP}$ ER



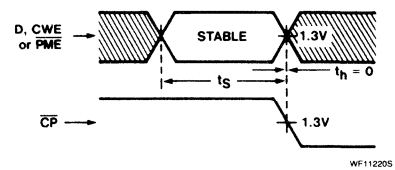
V_M = 1.3V for 74S; V_M = 1.5V for all other TTL familes.

Figure 3. Propagation Delay — $\overline{P}$ to Q and ER; Recovery Time — $\overline{P}$ to $\overline{CP}$



V_M = 1.3V for 74S; V_M = 1.5V for all other TTL familes.

Figure 4. Propagation Delay — MR to Q and ER; Recovery Time — MR to $\overline{CP}$



V_M = 1.3V for 74S; V_M = 1.5V for all other TTL familes.
The shaded areas indicate when the input is permitted to change for predictable output performance.

Figure 5. Set-up and Hold Times — D to $\overline{CP}$, CWE to $\overline{CP}$, and $\overline{PME}$ to $\overline{CP}$

Signetics

8X02A
Control Store Sequencer

Product Specification

Logic Products

FEATURES

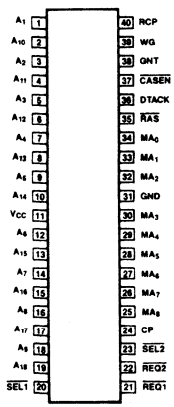
- **10-Bit Address Generator (1024 Microinstruction Addressability)**
- **Operating Frequency Exceeding 12 MHz**
- **Direct Branching Over Full Address Range**
- **Conditional Branching**
- **Subroutine Branching Capability**
- **4-Level Stack Register File**
- **Loop Control Facility Using Stack**
- **Three-State Address Outputs**

PRODUCT DESCRIPTION

The Signetics 8X02A Control Store Sequencer generates addresses to access instructions from a microprogram memory (control store). This high-speed device provides an efficient means of controlling the flow through a microprogram with a powerful set of sequencing functions. The 8X02A can directly address up to 1024 microinstructions; however, the total address space can be expanded by adding conventional paging techniques. Combined with memory, the 8X02A forms a powerful control section for CPU's, controllers, test equipment, and other microprogram-controlled systems.

8X02A PACKAGE AND PIN DESIGNATIONS

N PACKAGE



ORDER NUMBER

N8X02AN

PIN NO.	IDENTIFIER	FUNCTION
1, 28, 27	$AC_2 - AC_0$	Inputs used to select any one of eight Address Control Functions – see Table 1.
2	$\overline{EN}$	Enable three-state address outputs ($A_0 - A_9$); active-low input.
3 – 6, 8 – 13	$A_0 - A_9$	Three-state address outputs used to specify microprogram address; (A_0 = LSB, A_9 = MSB).
7	GND	Ground
14 – 21, 23, 24	$B_0 - B_9$	Branch address inputs: (B_0 = LSB, B_9 = MSB).
22	V_{CC}	Supply voltage.
25	CLK	Clock input (positive edge used for all triggering).
26	TEST	Active-high condition input used to determine conditional skips, branches, subroutine calls, and loop termination.

Control Store Sequencer 8X02A

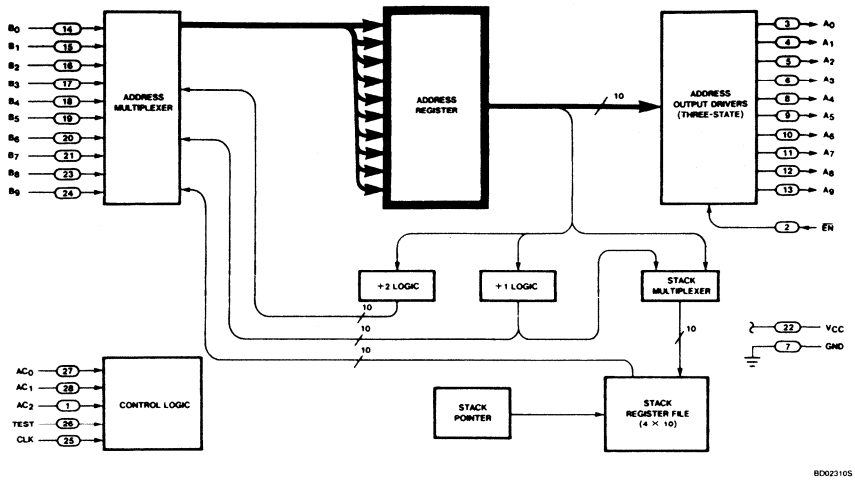


Figure 1. 8X02A Control Store Sequencer — Functional Block Diagram

FUNCTIONAL OPERATION

As shown in Figure 1, the data appearing on the address output pins ($A_0 - A_9$) is the contents of the 10-bit Address Register. On the rising edge of the clock input pulse (CLK), a new address is latched into the Address Register. This new address is supplied via the Address Multiplexer which selects one of five sources:

- Branch Address Input ($B_0 - B_9$)
- Current Address + 1
- Current Address + 2 (for the SKIP function)
- Stack Register File (most recent entry)
- All Zeroes (RESET)

The selection of the next address is determined by the "Address Control Function" specified by inputs $AC_0 - AC_2$ and the TEST input. Table 1 defines the eight Address Control Functions.

The "Reset" (RST) Address Control Function unconditionally forces all Address Register bits to zero on the rising edge of CLK. Sequential microprogram flow is provided by the "Increment" (INC) function which unconditionally increments the Address Register by one for each clock cycle. The Address Register automatically wraps around from the highest address (all "1s") to the lowest address (all "0s").

As shown in Table 1, the TEST input is used to conditionally execute four of the eight Address Control Functions. If the TEST input is **low** (false), the Address Register is simply incremented by one — (for the BLT function, the Stack Pointer is also decremented). If the TEST input is **high** (true), the sequencer executes one of the following:

- Skip (TSK) — the Address Register is incremented by two.
- Branch (BRT) — the Address Register is loaded from the Branch Address Inputs.
- Branch-to-Subroutine (BSR).
- Branch-to-Loop (BLT).

The Stack Register File holds up to four 10-bit addresses and operates in the Last-In/First-Out (LIFO) mode. A Stack Pointer keeps track of the next register of the Stack File to be written into; the pointer is incremented after each "push" and decremented after each "pop" — see Table 1. When branching to a subroutine (BSR), the return address (current address + 1) is "pushed" onto the stack and the branch address input is loaded into the Address Register. To return from a subroutine, the "POP" function pops the return address off the stack and loads it into the Address Register.

The "Push-for-Looping" (PLP) function may be specified in the first instruction of a loop to "push" the current address onto the stack: the Address Register is incremented. A "Branch-to-Loop" (BLT) function placed at the end of the loop "pops" the stack and conditionally branches to the top-of-loop address, depending on the TEST input. If the test for repeating the loop is satisfied (TEST input **high**), the sequencer causes a branch back to the first instruction of the loop in which the top-of-loop address is "pushed" back onto the stack. If the test fails (TEST input **low**), the top-of-loop address is discarded, the stack pointer is decremented and the Address Register is incremented. A combination of subroutines and loops may be nested up to four levels deep.

In abnormal circumstances, the Stack Pointer will wraparound from the fourth to the first register of the Stack File and vice-versa. If the stack is full (four addresses currently stored), an additional "push" causes the first (oldest) entry to be overwritten — (the four most recent entries are always maintained). If the stack is empty, a "pop" will access the fourth register of the Stack File; however, the contents of this register may be unpredictable.

The three-state address outputs ($A_0 - A_9$) are controlled by a common enable input ($\overline{EN}$). When the enable input is **high**, the output drivers are placed in the high-impedance state allowing alternative access to the microprogram memory. Other circuit functions are unaffected by $\overline{EN}$.

NOTE

To implement a RESET externally it is necessary to force all Address Control Inputs ($AC_0 - AC_2$) to the **high** state until at least one rising edge of CLK has occurred. If the AC inputs are supplied directly from the microprogram memory, a RESET may be accomplished by disabling the memory outputs. Pullup resistors should be provided to achieve the required high voltage level.

Table 1. Address Control Functions

MNEMONIC AND DESCRIPTION	CONTROL LINES				NEXT ADDRESS	STACK OPERATION	STACK POINTER
	AC_2	AC_1	AC_0	Test			
TSK – Test and skip	0 0	0 0	0 0	0 1	Current address + 1 Current address + 2	No change No change	No change No change
INC – Increment	0	0	1	X	Current address + 1	No change	No change
BLT – Branch to loop if test condition is true	0 0	1 1	0 0	0 1	Current address + 1 From stack register file	POP (ignore data) POP (read)	Decrement by 1 Decrement by 1
POP – Pop stack (return from subroutine)	0	1	1	X	From stack register file	POP (read)	Decrement by 1
BSR – Branch to subroutine if test condition is true	1 1	0 0	0 0	0 1	Current address + 1 Branch address inputs $B_0 - B_9$	No change PUSH (write current address + 1)	No change Increment by 1
PLP – Push for looping	1	0	1	X	Current address + 1	PUSH (write current address)	Increment by 1
BRT – Branch if test condition is true	1 1	1 1	0 0	0 1	Current address + 1 Branch address inputs $B_0 - B_9$	No change No change	No change No change
RST – Reset address to zero	1	1	1	X	All zeroes	No change	No change

X = Don't Care

ABSOLUTE MAXIMUM RATINGS

PARAMETER		RATING	UNIT
V_{CC}	Power supply voltage	+7	Vdc
V_{IN}	Input voltage	+5.5	Vdc
V_O	Off-state output voltage	+5.5	Vdc
T_{STG}	Storage temperature range	−65 to +150	°C

DC ELECTRICAL CHARACTERISTICS

Conditions: Commercial – V_{CC} = 5.0V (±5%), 0°C ≤ T_A ≤ 70°C

PARAMETER	DESCRIPTION	TEST CONDITIONS	LIMITS			UNIT
			Min	Typ[1]	Max	
V_{IH}	High level input voltage	V_{CC} = Min	2			V
V_{IL}	Low level input voltage	V_{CC} = Min			0.8	
V_{IC}	Input clamp voltage	V_{CC} = Min; I_I = −18mA			−1.5	
V_{OH}	High level output voltage	V_{CC} = Min; I_{OH} = −2.6mA	2.4	3.4		
V_{OL}	Low level output voltage	V_{CC} = Min; I_{OL} = 8mA		0.42	0.5	
I_I	Input current at maximum input voltage	V_{CC} = Max; V_I = 5.5V		1	100	μA
I_{IH}	High level input current: $AC_2 - AC_0$, TEST, CLK	V_{CC} = Max; V_{IH} = 2.7V		< 0.1	40	μA
	$B_9 - B_0$, $\overline{EN}$			< 0.1	20	
I_{IL}	Low level input current: $AC_2 - AC_0$, TEST, CLK	V_{CC} = Max; V_{IL} = 0.4V		−24	−800	μA
	$B_9 - B_0$, $\overline{EN}$			−12	−400	
I_{OS}	Short circuit output current[2]	V_{CC} = Max	−15	−60	−100	mA
I_{OZH}	High-Z state output current – high level	V_{CC} = Max; V_{OH} = 2.7V			20	μA
I_{OZL}	High-Z state output current – low level	V_{CC} = Max; V_{OL} = 0.4V			−20	μA
I_{CC}	Supply current	V_{CC} = Max		170	250	mA

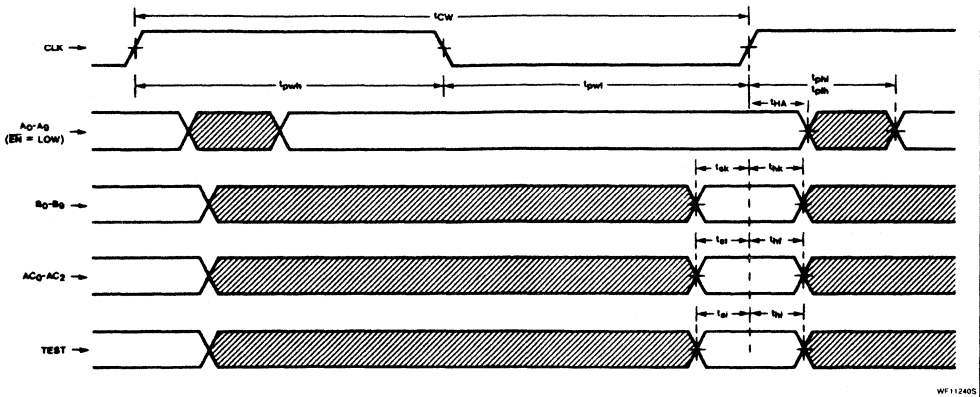
NOTES:
1. Typical limits are: V_{CC} = 5.0V and T_A = 25°C.
2. For purposes of testing, not more than one output should be shorted at a time.

AC ELECTRICAL CHARACTERISTICS **Conditions:** Commercial – V_{CC} = 5.0V (±5%), 0°C ≤ T_A ≤ 70°C
Loading – See Test Loading Circuit

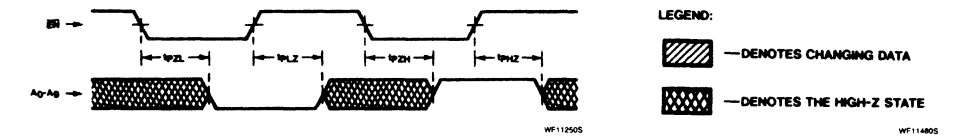
PARAMETERS[1]		REFERENCES		LIMITS[4]			UNIT
		From	To	Min	Typ[2]	Max	
Pulse width:							
t_{CW}	Clock cycle time	↑ CLK	↑ CLK	80			ns
t_{PWH}	Clock high	↑ CLK	↓ CLK	35	24		ns
t_{PWL}	Clock low	↓ CLK	↑ CLK	15	9		ns
Propagation delay:							
t_{PLZ}	Low to high-Z	↑ $\overline{EN}$	$A_0 - A_9$		14	20	ns
t_{PHZ}	High to high-Z	↑ $\overline{EN}$	$A_0 - A_9$		35	42	ns
t_{PZL}	High-Z to low	↓ $\overline{EN}$	$A_0 - A_9$		10	20	ns
t_{PZH}	High-Z to high	↓ $\overline{EN}$	$A_0 - A_9$		20	30	ns
t_{PHL}	High to low	↑ CLK	↓ $A_0 - A_9$		25	45	ns
t_{PLH}	Low to high	↑ CLK	↑ $A_0 - A_9$		25	45	ns
t_{HA}	Address output hold time[3]	↑ CLK	$A_0 - A_9$	13			ns
Set-up/hold times:							
t_{SF}	Function set-up time	$AC_0 - AC_2$	↑ CLK	20	18		ns
t_{SK}	Branch set-up time	$B_0 - B_9$	↑ CLK	15	7		ns
t_{SI}	Test set-up time	TEST	↑ CLK	20	15		ns
t_{HF}	Function hold time	↑ CLK	$AC_0 - AC_2$	20	2		ns
t_{HK}	Branch hold time	↑ CLK	$B_0 - B_9$	15	9		ns
t_{HI}	Test hold time	↑ CLK	TEST	12	−2		ns

NOTES:
1. Parameter definitions are illustrated in the Timing Diagrams – See Figure 2.
2. Typical limits are: V_{CC} = 5.0V and T_A = 25°C.
3. t_{HA} is the minimum time the current address outputs remain stable before changing. This delay may be used to provide some of the hold times required for the AC, B, and TEST inputs, if these inputs are determined by the microprogram memory addressed by the 8X02A.
4. This data supercedes the November, 1980 edition of this data sheet.

TIMING DIAGRAM



a. Clock Synchronization

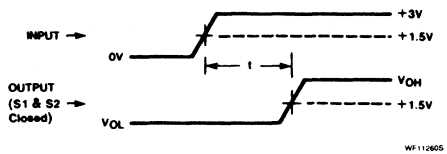


b. Output Enable

Figure 2. Timing Diagrams

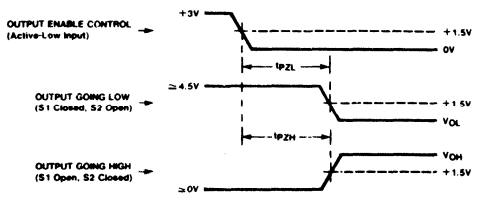
AC VOLTAGE WAVEFORMS AND TEST LOADING

PROPAGATION DELAY (Typical Example):

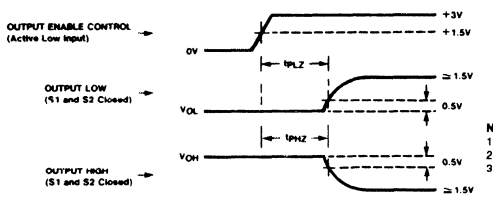


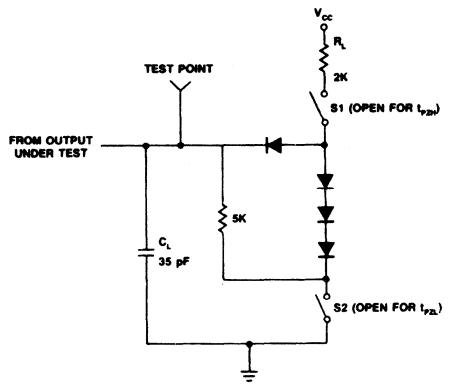
NOTE:
Pulse widths and setup/hold times are measured using the same reference points shown in the above waveforms.

OUTPUT ENABLE TIMES (Three-state outputs):



OUTPUT DISABLE TIMES (Three-state outputs):





NOTES:
1. C_L includes probe and jig capacitance.
2. All diodes are 1N3064 or equivalent.
3. Switches S1 and S2 are both closed for all measurements except Output Enable times — See AC VOLTAGE WAVEFORMS.

APPLICATION

FUNCTIONAL DESCRIPTION

Figure 3 shows a typical configuration of an 8X02A-based control section in a CPU application. Microinstructions read from the memory are used to produce control signals for the CPU and to determine the next microinstruction via the 8X02A Address Control inputs ($AC_0 - AC_2$). In the case of a conditional branch or skip, the status condition applied to the 8X02A TEST input is selected according to the microinstruction. In a branch-type microinstruction, a branch field typically supplies the 8X02A Branch Address inputs ($B_0 - B_9$). (In non-branching instructions, this field may contain other CPU control information.) When a macroinstruction is presented to the CPU, the starting address of the microprogram routine which executes the macroinstruction is presented to the Branch Address inputs. Similar configurations may be used for other applications in which the Branch Address inputs are typically supplied directly from the microprogram memory.

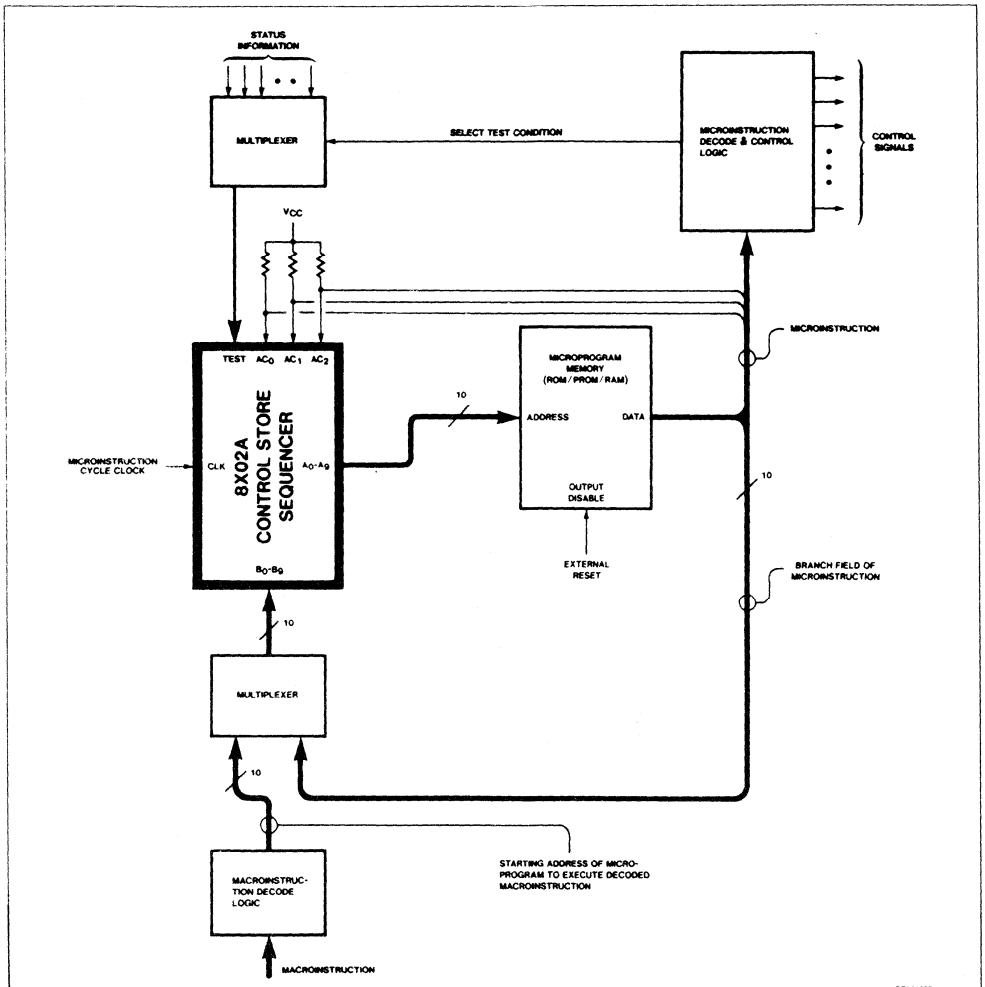


Figure 3. Control Section of a Microprogrammed CPU

Signetics

8X41
Autodirectional Bus Transceiver

Product Specification

Logic Products

DESIGN FEATURE

- **Intelligent bidirectional bus repeater with self-generating or external control**
- **Eight independent channels**
- **Open-collector outputs (meets DEC UNIBUS* specifications)**
- **TTL compatible**
- **High speed (30-nanoseconds max)**
- **Expandable to any number of bits**
- **High input impedance for every operating value of V_{CC}**
- **Low input current (less than 100-microamperes); high output current (up to 70-milliamperes)**
- **0.6 in. 24-pin DIP**
- **+5V supply**

USE AND APPLICATION

- **Minicomputers**
- **Microcomputers MOS/Bipolar**
- **Communications**
- **Signal buffer**
- **Bus fan-out extensions**
- **Distributed processing**
- **Bidirectional bus connector/ isolator**

PRODUCT DESCRIPTION

The Signetics 8X41 Autodirectional Bus Transceiver is a general purpose asynchronous device ideal for system bus expansion applications. The 8X41 consists of eight data channels, each with one pair of terminals (A_i and B_i); each data channel can be operated independently.

The device requires no external controls since all intelligence is internally generated; thus, operation of the device is completely autonomous. The first logic low signal that occurs on one channel terminal (A_i and B_i) will be repeated on the corresponding terminal (B_i or A_i) of the same channel.

The 8X41 is designed for use in open-collector bus systems where high speed and low-current inputs/high-current outputs are required. In system configurations, the discrete capabilities of the bus transceiver can be expanded by parallel connection to service any number of bits. To provide reliable operation and integrity of data transfers, all channels are disabled by an on-chip power monitor whenever V_{CC} falls below approximately 4V.

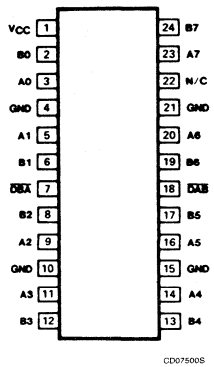
FUNCTIONAL OPERATION

The 8X41 (Figure 1) consists of eight functionally independent yet logically identical channels. Each channel consists of two bus terminals (A_i and B_i); each terminal is internally connected to an open-collector driver and a high-impedance receiver. The monitoring state of each channel is defined when both terminals (A_i and B_i) are "high"; in this state, the internal logic of the 8X41 continually examines the A and B bus signals to determine signal direction — A_i to B_i or B_i to A_i. A low signal occurring at either of the two terminals causes the open-collector driver on the opposite terminal to follow suit; hence, the signal is repeated by the 8X41. For each channel, latches L1 and L2 determine signal direction. As shown in the truth table for these latches, there is no transmission of data when both signals are low; however, this condition should never occur during normal system operation.

The internal automatic direction control can be overridden by either or both of the common disable inputs — $\overline{DBA}$ and $\overline{DAB}$. When $\overline{DBA}$ is driven low ($\overline{DAB}$ = high), the B_i to A_i path is interrupted and the device becomes a unidirectional repeater in the A_i to B_i direction only. With these conditions reversed ($\overline{DAB}$ = low and $\overline{DBA}$ = high), the A_i to B_i path is interrupted and the chip functions as a unidirectional repeater in the B_i to A_i direction. When both control signals are low, data passage is inhibited in both directions. Refer to the I/O truth table for all possible input/output conditions.

8X41 PACKAGE/PIN DESIGNATIONS

N PACKAGE



CD07500S

ORDER NUMBER
N8X41N

PIN NO.	IDENTIFIER	FUNCTION
1	V_{CC}	+5V power
3, 5, 9, 11, 14, 16, 20 and 23	$A_0 - A_7$	A-bus input/ output
2, 6, 8, 12, 13, 17, 19 and 24	$B_0 - B_7$	B-bus input/ output
4, 10, 15, and 21	GND	Circuit ground
7 & 18	$\overline{DBA}$ & $\overline{DAB}$	Input/output control
22	N/C	No connection

*Trademark of the Digital Equipment Corporation

Autodirectional Bus Transceiver — 8X41

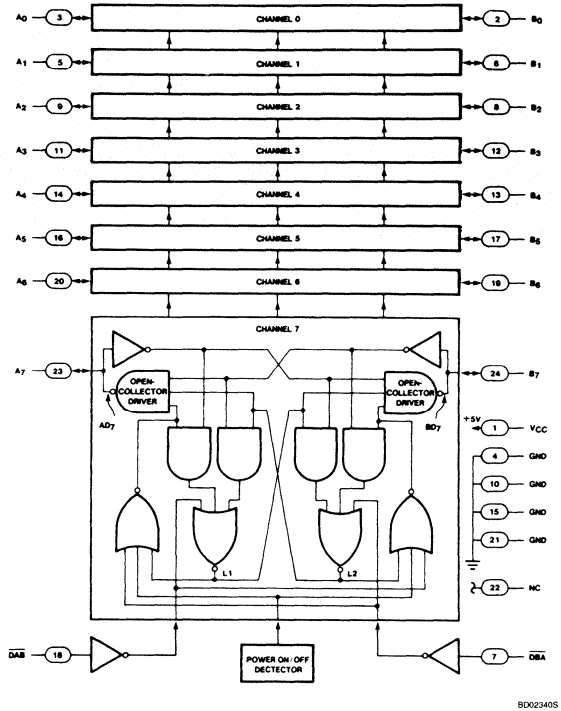


Figure 1. Logic Diagram of 8X41

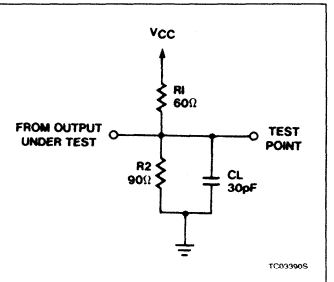
$\overline{DBA}$	$\overline{DAB}$	FUNCTION
0	0	Data transmission inhibited
0	1	$A_i \rightarrow B_i$
1	0	$A_i \leftarrow B_i$
1	1	$A_i \rightarrow B_i$ $A_i \leftarrow B_i$

i = Channel 0, 1, 2, 3, 4, 5, 6, or 7
$A_i \rightarrow B_i$ = Data transmission from A_i to B_i
$A_i \leftarrow B_i$ = Data transmission from B_i to A_i

TRUTH TABLE

LATCHES		DIRECTION OF DATA
L_1	L_2	
1	1	Monitoring state
1	0	A_i to B_i
0	1	B_i to A_i
0	0	No transmission

LOAD CIRCUIT FOR OUTPUTS



INPUT/OUTPUT TRUTH TABLE

EXTERNAL CONTROLS		INPUT SIGNALS		OUTPUT DRIVER SIGNALS	
$\overline{DAB}$	$\overline{DBA}$	A_i	B_i	AD_i	BD_i
H	H	L	L	H	H
H	H	L	H	H	L
H	H	H	L	L	H
H	H	H	H	H	H
H	L	L	L	H	L
H	L	L	H	H	L
H	L	H	L	H	H
H	L	H	H	H	H
L	H	L	L	L	H
L	H	L	H	H	H
L	H	H	L	L	H
L	H	H	H	H	H
L	L	X	X	H	H

NOTES:
A_i = External signal
AD_i = Output A driver
B_i = External signal
BD_i = Output B driver
X = Don't care

DC CHARACTERISTICS V_{CC} = 5V (±5%); T_A = 0°C to 70°C

PARAMETER		TEST CONDITIONS	LIMITS			UNIT
			Min	Typ	Max	
V_{OL}	Bus output low voltage (driver ON)	I_{OL} = 70mA; V_{CC} = MIN			0.5	V
*V_B	Bus input threshold voltage (driver OFF)		1.3		1.7	V
V_{IH} ($\overline{DBA}$, $\overline{DAB}$ only)	High level input voltage		2.0			V
V_{IL} ($\overline{DBA}$, $\overline{DAB}$ only)	Low level input voltage				0.8	V
V_{IC}	Input clamp voltage	V_{CC} = MIN; I_{IL} = −18mA			−1.5	V
V_{PD}	Power ON/OFF detector threshold voltage		3.7		4.35	V
I_{IH} ($\overline{DBA}$, $\overline{DAB}$ only)	High level input current	V_{CC} = Max; V_{IN} = 2.7V			20	µA
I_{IL} ($\overline{DBA}$, $\overline{DAB}$ only)	Low level input current	V_{CC} = Max; V_{IN} = 0.4V			−0.4	mA
I_I	Bus input current (driver OFF)	V_{CC} = Max; V_B = 2.5V*			100	µA
		V_{CC} = Max; V_B = 0V*			−20	
I_{OFF}	Bus leakage current (power OFF)	V_{CC} = 0V; V_B = 2.5V*			100	µA
I_{CC}	Supply current	V_{CC} = Max; $A_0 - A_7$ = Low or $B_0 - B_7$ = Low and $\overline{DBA}$ = $\overline{DAB}$ = High		145	180	mA

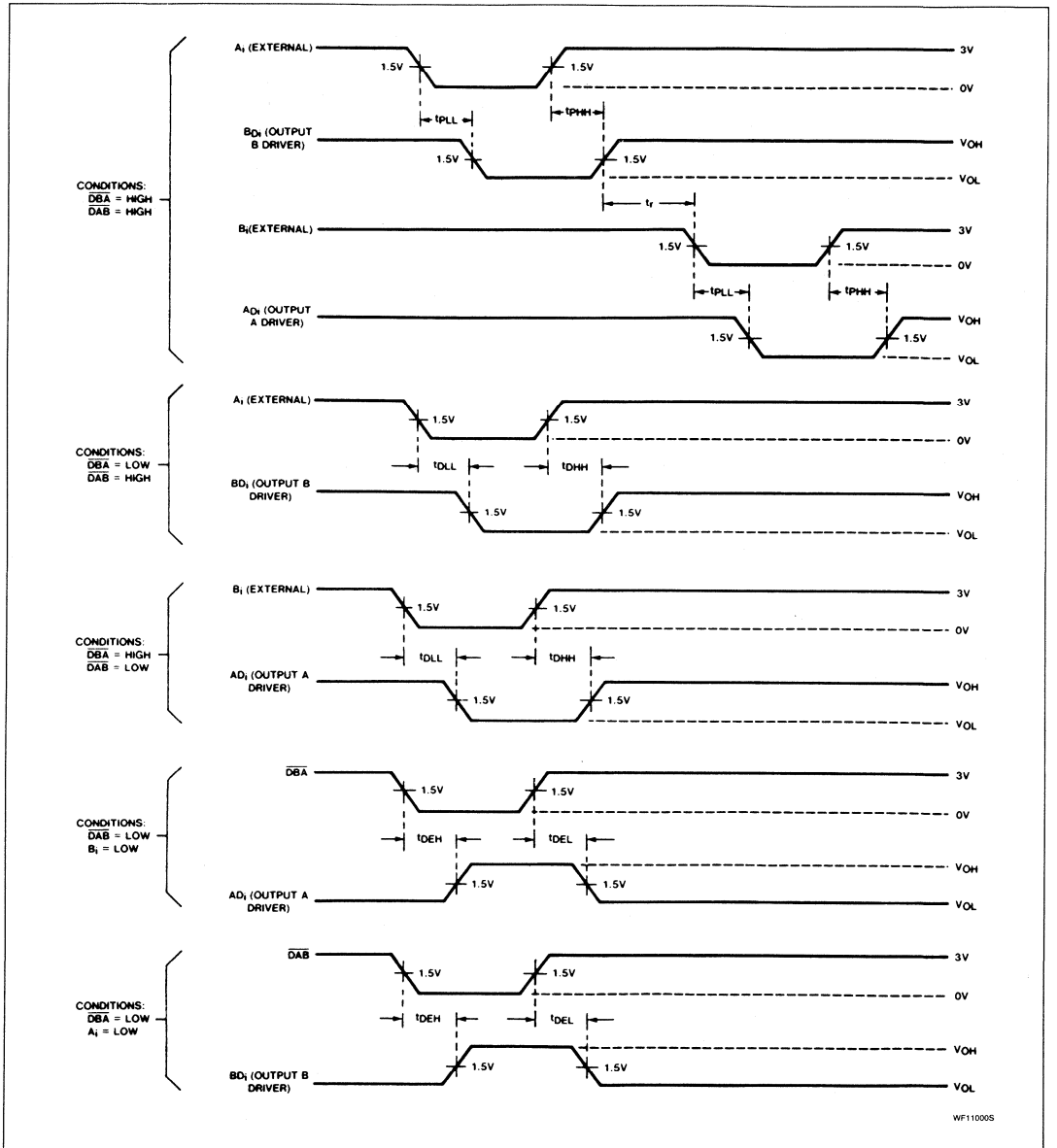
NOTE:
*$V_B = V_{BUS}$

AC CHARACTERISTICS V_{CC} = 5V (±5%); T_A = 0°C to 70°C

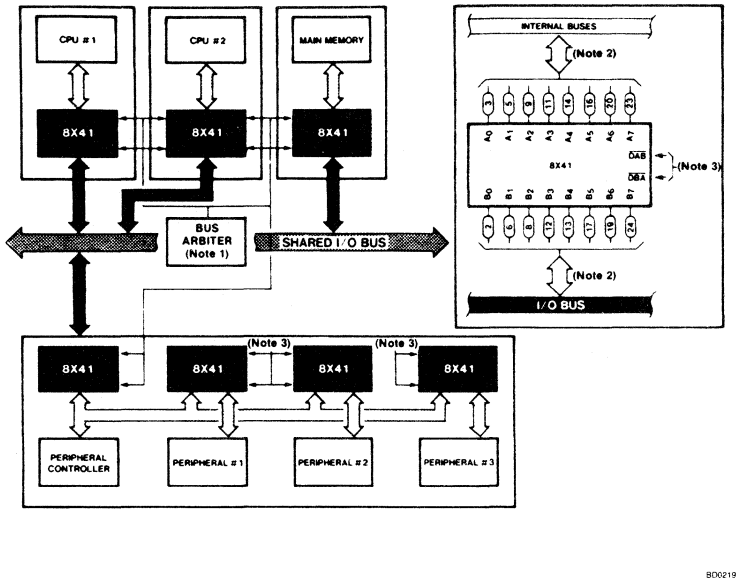
PARAMETER	DESCRIPTION	FROM	TO	TEST CONDITIONS	LIMITS			UNIT
					Min	Typ	Max	
t_{PLL}	Propagation delay	Low A_i Low B_i	Low BD_i Low AD_i	$\overline{DBA}$ = $\overline{DAB}$ = High			30	ns
t_{PHH}	Propagation delay	High A_i High B_i	High BD_i High AD_i	$\overline{DBA}$ = $\overline{DAB}$ = High			30	ns
t_{DHH}	Propagation delay	High A_i	High BD_i	$\overline{DBA}$ = Low; $\overline{DAB}$ = High			25	ns
		High B_i	High AD_i	$\overline{DAB}$ = Low; $\overline{DBA}$ = High			25	ns
t_{DLL}	Propagation delay	Low A_i	Low BD_i	$\overline{DBA}$ = Low; $\overline{DAB}$ = High			25	ns
		Low B_i	Low AD_i	$\overline{DAB}$ = Low; $\overline{DBA}$ = High			25	ns
t_{DEH}	Propagation delay	Low $\overline{DAB}$	High AD_i	$\overline{DAB}$ = Low; A_i = Low			30	ns
t_{DEL}	Propagation delay	High $\overline{DBA}$	Low AD_i	$\overline{DAB}$ = Low; B_i = Low			30	ns
t_{DEH}	Propagation delay	Low $\overline{DBA}$	High BD_i	$\overline{DBA}$ = Low; B_i = Low			30	ns
t_{DEL}	Propagation delay	High $\overline{DAB}$	Low BD_i	$\overline{DBA}$ = Low; A_i = Low			30	ns
t_r	Recovery time (see timing diagram)	—	—	$\overline{DBA}$ = $\overline{DAB}$ = High		20		ns

NOTES:
A_i = External signal
AD_i = Output A driver
B_i = External signal
BD_i = Output B driver

8X41 TIMING DIAGRAM



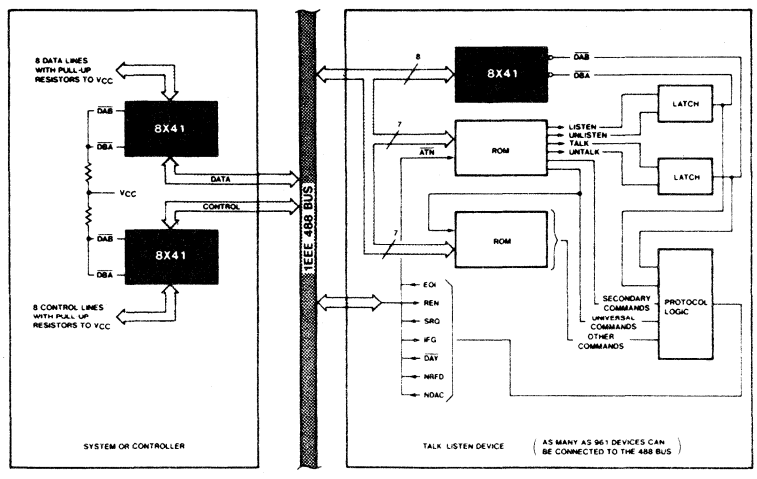
USING THE 8X41 IN A BUS-SHARED CONFIGURATION



BD02190S

NOTES:
1. The bus-control logic can be handled as shown here, integrated in each system module, or externally situated.
2. The A-port and B-port of the 8X41 are transparent to external connections, that is $A_1 - A_7$ can be connected to the I/O bus and $B_0 - B_7$ can be connected to the internal bus, and vice-versa.
3. Refer to Truth Tables adjacent to Figure 1 for data control capabilities.

INTERFACING 8X41 TO IEEE 488 BUS



BD02200S

Signetics

8X60
FIFO RAM Controller (FRC)

Product Specification

Logic Products

FEATURES

- **12-Bit FIFO Address Generator**
- **Data Rate Exceeding 8MHz**
- **Asynchronous Read/Write Operations**
- **Three-State Address Outputs**
- **User-Defined Word Width**
- **Specifically Designed for Use with High-Speed Bipolar RAMs (Adaptable for Use with MOS RAMs)**
- **TTL Input and Output**
- **16mA Address-Drive Capability**

USE AND APPLICATION

- **Interface Between Independently-Clocked Systems**
- **Buffer Memories for Disk and/or Tape**
- **Data Communication Concentrators**
- **CPU/Terminal Buffering**
- **DMA Applications**
- **CRT Terminals**

PRODUCT DESCRIPTION

The Signetics 8 x 60 FIFO RAM Controller (FRC) is an address and status generator designed to implement a high-speed/high-capacity First-In/First-Out (FIFO) stack utilizing standard off-the-shelf RAMs — see **Applications** on the last page of this data sheet. The FRC can control up to 4096 words of buffer memory; intermediate buffer sizes can be selected — refer to the memory length table on the next page. Built-in arbitration logic handles read/write operations on a first-come/first-served basis.

As shown in Figure 1, the FRC consists of:

- A 12-Bit Write Address Generation Counter (Counter #1) and a 12-Bit Read Address Generation Counter (Counter #2).
- A 12-Bit Up/Down Status Counter (Counter #3).
- Twelve Three-State Address Drivers.
- Control Logic.

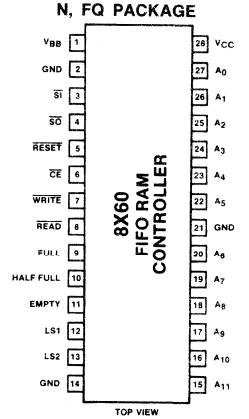
The two address counters, #1 and #2, respectively, are used to generate write and read addresses; the outputs of these counters are multiplexed to the three-state address drivers. Counter #3 generates **full, empty,** and **half full** status.

FUNCTIONAL OPERATION

The FRC operates in either of two basic modes — **write** into the FIFO buffer memory or **read** from the FIFO buffer memory. These two operations are described in subsequent paragraphs and the complete sequence is summarized in Table 1. Typical Write/Read timing relationships, arbitration logic, and chip-enable control are shown in the Timing Diagrams.

PACKAGE AND PIN DESIGNATOR

N, FQ PACKAGE



TOP VIEW

ORDER NUMBERS

N8X60N, RB8X60F

PIN NO.	IDENTIFIER	FUNCTION
1	V_{BB}	Supply voltage for internal circuits.
2, 14, 21	GND	Circuit ground.
3	$\overline{SI}$	Shift-In request for write cycle; active-low input.
4	$\overline{SO}$	Shift-Out request for read cycle; active-low input.
5	$\overline{RESET}$	Active-low master reset input.
6	$\overline{CE}$	Active-low chip enable input.
7	$\overline{WRITE}$	Write cycle address valid; active-low output.
8	$\overline{READ}$	Read cycle address valid; active-low output.
9	FULL	Memory full status output; also, override input capability. Active when **high.**
10	HALF FULL	Memory half-full status output; active-high.
11	EMPTY	Memory empty status output; also, override input capability. Active when **high.**
12	LS1	Least significant bit (LSB) of the memory length select input.
13	LS2	Most significant bit (MSB) of the memory length select input.
15 – 20 22 – 27	$A_{11} - A_0$	Three-state address outputs: A_0 = LSB.
28	V_{CC}	Supply voltage.

FIFO RAM Controller (FRC) — 8X60

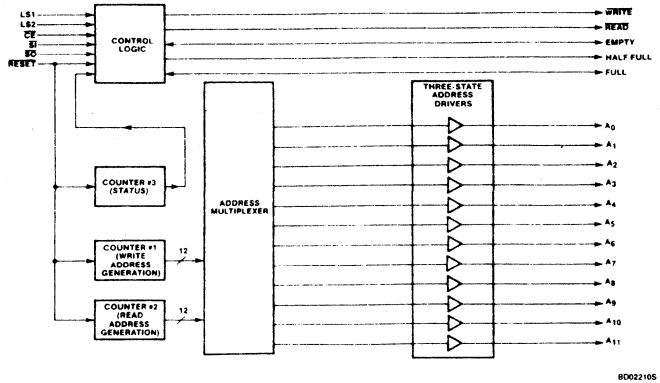


Figure 1. Functional Block Diagram of FIFO RAM Controller

FIFO BUFFER MEMORY — WRITE CYCLE

To perform a write operation, $\overline{SO}$ must be **high** and SI must be **low.** When these conditions exist and other control parameters (Table 1) are satisfied, the write address in Counter #1 (Figure 1) is output to the address bus via the multiplexer and $\overline{WRITE}$ output goes **low.** (**Note.** Normally, the $\overline{WRITE}$ output goes **low** after the address output becomes state — refer to **$\overline{WRITE}$ Cycle Timing Diagram.** The $\overline{WRITE}$ output may then act as a **write** or **chip** enable for the RAMs that are used to implement the memory.

When the **write** cycle is ended ($\overline{SI}$ is forced high), the $\overline{WRITE}$ output goes **high**, the address output buffers return to a high-impedance state. Counter #1 (Write Address Generation) and Counter #3 (Status) are both incremented, and Counter #2 (Read Address Generation) remains unchanged.

FIFO BUFFER MEMORY — READ CYCLE

To perform a read operation, $\overline{SI}$ must be **high** and $\overline{SO}$ must be **low.** When these conditions exist and other control parameters (Table 1) are satisfied, the read address contained in Counter #2 (Figure 1) is output to the address bus and the $\overline{READ}$ output goes **low.**

When the **read** cycle is ended ($\overline{SO}$ is forced **high**) the $\overline{READ}$ output goes **high**, the output buffers return to a high-impedance state. Counter #2 (Read Address Generation) is incremented. Counter #3 (Status) is decremented, and Counter #1 (Write Address Generation) remains unchanged.

MEMORY LENGTH

LS1	LS2	HALF LENGTH	FULL LENGTH
L	L	2048	4096
H	L	32	64
L	H	512	1024
H	H	128	256

CONTROL LOGIC

To prevent the possibility of operational conflicts, $\overline{SI}$ and $\overline{SO}$ are treated on a first-come/first-served basis; these two input signals are controlled by internal arbitration logic — refer to the applicable **Timing Diagrams** and **AC Characteristics** for functional and timing relationships. If one cycle is requested while the other cycle is in progress, the requested cycle will commence as soon as the current-cycle is complete (provided other control parameters are satisfied).

As shown in the accompanying diagram, the buffer length of the FIFO memory can be hardware-selected via the Length Select (LS1, LS2) Inputs. When less than the maximum length is selected, the unused high-order bits of the address outputs are held in the high-impedance state.

Generation of the status output signals (HALF FULL, FULL and EMPTY) is a function of the Length Select (LS1, LS2) inputs and the current state of Status Counter #3. In general, the status outputs reflect the conditions that follow:

- **HALF FULL** — this status output signals goes **high** on the positive-going edge of $\overline{SI}$ if the MSB of the selected length of Counter #3 becomes a "1". The HALF FULL signal will go from **high-to-low** on the positive-going edge of $\overline{SO}$ when, after the **read** cycle, the selected length of Counter #3 changes from "100 ... 00" to "011 ... 11". For example, if the selected memory length is 256 words (FULL = 256), then HALF FULL = 128 words; hence, on the positive-going edge of $\overline{SO}$ when Counter #3 reaches a count of 127, the HALF FULL output will go from **high-to-low.**
- **FULL** — this signal serves both as a status output and as an override input. The FULL signal goes **high** on the negative-going edge of $\overline{SI}$ if all bits of Counter #3 for selected length are equal to "1". The FULL output goes from **high-to-low** on the negative-going edge of $\overline{SO}$.
- **EMPTY** — this signal also serves as a status output and as an override input. On the negative-going edge of $\overline{SO}$, the EMPTY output is driven **high** if Status Counter #3 contains a value of "1"; on the positive-going edge of $\overline{SO}$, the counter is decremented to "0". The EMPTY output goes from **high-to-low** on the negative-going edge of $\overline{SI}$.

Once the FULL signal is **high**, further Write Cycle Requests ($\overline{SI}$ = low) are ignored; similarly, once the EMPTY signal is **high**, further Read Cycle requests ($\overline{SO}$ = low) are ignored. However, to accommodate diversified applications, the FULL and EMPTY outputs are

open-collector with on-chip 4.7K passive pull-up resistors. If either the FULL or EMPTY pins are forced **low** via external control, the corresponding **write** or **read** cycle may resume (provided the external FULL or EMPTY input is held **low** until the corresponding $\overline{\text{WRITE}}$ or $\overline{\text{READ}}$ output goes **low**) and the address/status counters will continue normal operation* — refer to Table 1.

The user must force the $\overline{\text{RESET}}$ input **low** to initialize the chip. (**Note.** If the $\overline{\text{RESET}}$ signal is driven **low** during a **write** or **read** cycle, the address output may have a short period of uncertainty before assuming a high-impedance state.) The following actions occur when $\overline{\text{RESET}}$ is active:

- All internal counters are set to "0".
- All address output lines are forced to the high-impedance state.
- HALF FULL and FULL outputs are forced **low**.
- $\overline{\text{WRITE}}$, $\overline{\text{READ}}$, and EMPTY outputs are forced high.

When $\overline{\text{CE}}$ is **high**, the address output lines are forced to the high-impedance state, further **write** or **read** cycle requests are ignored, and all counters remain unchanged. If $\overline{\text{CE}}$ switches from **low**-to-**high** during a **write** or **read** cycle, the cycle in progress is always completed before the disabled state is entered. For details of these operations, refer to the timing information shown later in this data sheet.

*Refer to **Note** on inside back cover

Table 1. Summary of Operation

INPUTS				INITIAL CONDITIONS	RESULTING OUTPUTS			COMMENTS
$\overline{\text{RESET}}$	$\overline{\text{CE}}$	$\overline{\text{SI}}$	$\overline{\text{SO}}$		$\overline{\text{WRITE}}$	$\overline{\text{READ}}$	Address Bus	
L	X	X	X		H	H	Hi-Z	Reset all counters to 0.
H	X	H	H		H	H	Hi-Z	No action
H	L	L	H	FULL = L	L	H	Write address from Ctr #1	Shift into FIFO stack (Write Cycle)
H	L	L	H	FULL = H	H	H	Hi-Z	Stack full (write inhibited)
H	L	H	L	EMPTY = L	H	L	Read address from Ctr #2	Shift out of FIFO stack (Read Cycle)
H	L	H	L	EMPTY = H	H	H	Hi-Z	Stack empty (read inhibited)
H	L	L	↓	Write cycle in progress	L	H	Write address from Ctr#1	Continue write cycle (until $\overline{\text{SI}}$ goes high)
H	L	↓	L	Read cycle in progress	H	L	Read Address from Ctr #2	Continue read cycle (until $\overline{\text{SO}}$ goes high)
H	L	L	L	EMPTY = H	L	H	Write address from Ctr #1	Shift in (read inhibited)
H	L	L	L	FULL = H	H	L	Read address from Ctr #2	Shift out (write inhibited)
H	L	↑	H	Write cycle in progress	↑	H	Goes to Hi-Z	Increment write address counter #1 and status counter #3
H	L	H	↑	Read cycle in progress	H	↑	Goes to Hi-Z	Increment read address counter #2; decrement status counter #3
H	L	↑	L	Write cycle in progress (note 1)	↑	↓	Changes to read address from Ctr #2	Increment write address counter #1 and status counter #3
H	L	L	↑	Read cycle in progress (note 2)	↓	↑	Changes to write address from Ctr #1	Increment read address counter #2; decrement status counter #3
H	H	↓	H		H	H	Hi-Z	Chip disabled
H	H	H	↓		H	H	Hi-Z	Chip disabled
H	↑	L	X	FULL = L; write cycle begun (note 1)	L	H	Write address from Ctr #1	Continue write cycle (until $\overline{\text{SI}}$ goes high)
H	↑	X	L	EMPTY = L: read cycle begun (note 2)	H	L	Read address from Ctr #2	Continue read cycle (until $\overline{\text{SO}}$ goes high)
H	↓	L	L	FULL = L; EMPTY = L	–	–	–	This set of conditions should be avoided

NOTES:
1. Write cycle will occur if either $\overline{\text{SI}}$ goes **low** before $\overline{\text{SO}}$ goes **low** or EMPTY = H when $\overline{\text{SO}}$ goes **low.**
2. Read cycle will occur if either $\overline{\text{SO}}$ goes **low** before $\overline{\text{SI}}$ goes **low** or FULL = H when $\overline{\text{SI}}$ goes **low.**

ABSOLUTE MAXIMUM RATINGS

PARAMETER		RATING	UNIT
V_{CC}	Power supply voltage	+7	Vdc
V_{BB}	Supply voltage for internal circuits	+4	Vdc
V_{IN}	Input voltage	+5.5	Vdc
V_O	Off-state output voltage	+5.5	Vdc
T_{STG}	Storage temperature range	−65 to +150	°C

DC ELECTRICAL CHARACTERISTICS

Conditions: Commercial — V_{CC} = 5.0 V (±5%), V_{BB} = 1.5 V (±5%)[1], $0°C \leq T_A \leq 70°C$

PARAMETER		TEST CONDITIONS	LIMITS			UNITS
			Min	Typ[2]	Max	
V_{IH}	High level input voltage	Note 3	2.0			V
V_{IL}	Low level input voltage				0.8	V
V_{OH}	High level output voltage: All outputs except FULL and EMPTY	V_{CC} = MIN; I_{OH} = −2.6mA	2.7	3.5		V
V_{OL}	Low level output voltage: Address Bus, $\overline{WRITE}$, $\overline{READ}$	V_{CC} = MIN; I_{OL} = 16mA		0.38	0.5	V
	HALF FULL, FULL, and EMPTY	V_{CC} = MIN; I_{OL} = 8mA		0.35	0.5	V
V_{CD}	Diode clamp voltage: All inputs except FULL and EMPTY	V_{CC} = MIN; I_{CD} = −18mA	−1.5	−0.8		V
I_{IH}	High level input current: All inputs except FULL and EMPTY	V_{CC} = MAX; V_{IH} = 2.7V		0.1	20	μA
	FULL and EMPTY	V_{CC} = MAX; V_{IH} = 2.7V; stack FULL or stack EMPTY (Note 3)		−470	−750	μA
I_{IL}	Low level input current: All inputs except FULL and EMPTY	V_{CC} = MAX; V_{IL} = 0.4V		−0.17	−0.4	mA
	FULL and EMPTY	V_{CC} = MAX; V_{IL} = 0.4V; Stack FULL or Stack EMPTY		−1.12	−1.8	mA
I_{OH}	High level output current: FULL, EMPTY	V_{CC} = MIN; V_{OH} = V_{CC} (min)		15	100	μA
I_{OZH}	High-Z output current (HIGH); address bus (Three-state)	V_{CC} = MAX; V_{OUT} = 2.4V		0.9	20	μA
I_{OZL}	High-Z output current (LOW); address bus (Three-state)	V_{CC} = MAX; V_{OUT} = 0.5V		−0.6	−20	μA
I_I	Input leakage current: All inputs except FULL and EMPTY	V_{CC} = MAX; V_{IN} = 5.5V		0.03	0.1	mA
I_{OS}	Short-circuit output current: address bus and HALF FULL	V_{CC} = MAX; V_{OH} = 0V	−15	−68	−100	mA
	$\overline{WRITE}$, $\overline{READ}$	V_{CC} = MAX; V_{OH} = 0V	−40	−73	−100	mA
I_{CC}	Supply current from V_{CC}	V_{CC} = MAX; Address Bus = High-Z; 0°C →		81	140	mA
		70°C →		81	110	
I_{BB}	Supply current from V_{BB}	V_{BB} = Max; 0°C →		63	95	mA
		70°C →		63	85	

NOTES:

1. V_{BB} can be obtained from a regulated 1.5V supply; alternately, proper supply current (I_{BB}) can be obtained by connecting a 56-ohm (± 5%, 0.5W) resistor in series with V_{CC} as shown later in the APPLICATIONS diagram.
2. Typical limits are: V_{CC} = 5.0V; T_A = 25°C.
3. Because of the internal pull-up resistor on the FULL and EMPTY pins, a negative current is required to force the required voltage.

AC ELECTRICAL CHARACTERISTICS

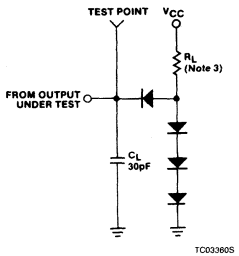
Conditions: Commercial — V_{CC} = 5.0 V (±5%) V_{BB} = 1.5 V (±5%) 0°C ≤ T_A ≤ 70°C

PARAMETERS	REFERENCES		TEST CONDITIONS	LIMITS			UNITS
	From	To		Min	Typ	Max	
Pulse widths:							
T_{LH} $\overline{SI}$ high	↑ $\overline{SI}$	↓ $\overline{SI}$	Stack approaching FULL (note 1)	25	13		ns
T_{DH} $\overline{SO}$ high	↑ $\overline{SO}$	↓ $\overline{SO}$	Stack approaching EMPTY (note 1)	30	16		ns
Write cycle timing:							
T_{LA} Address stable delay	↓ $\overline{SI}$	An	FULL = Low; $\overline{SO}$ = High		40	55	ns
T_{AW} Address lead time	An	↓ $\overline{WRITE}$		3			ns
T_{LAW} $\overline{WRITE}$ output active delay	↓ $\overline{SI}$	↓ $\overline{WRITE}$	FULL = Low; $\overline{SO}$ = High	35	51	65	ns
T_{LW} $\overline{WRITE}$ output inactive delay	↑ $\overline{SI}$	↑ $\overline{WRITE}$			3	10	ns
T_{WA} Address lag time	↑ $\overline{WRITE}$	An		20	34		ns
T_{LT} Address output disable	↑ $\overline{SI}$	An (Hi-Z)			37	60	ns
T_{LF} FULL status active delay	↓ $\overline{SI}$	↑ FULL	Stack approaching FULL; $\overline{SO}$ = High		39	65	ns
T_{LE} EMPTY status inactive delay	↓ $\overline{SI}$	↓ EMPTY	Stack = EMPTY		40	65	ns
T_{HFH} HALF-FULL status active delay	↑ $\overline{SI}$	↑ HALF FULL	Stack approaching HALF-FULL		30	45	ns
T_{DW} $\overline{WRITE}$ output active after read	↑ $\overline{SO}$	↓ $\overline{WRITE}$	Both $\overline{SI}$ & $\overline{READ}$ = Low		74	95	ns
Read cycle timing:							
T_{DA} Address stable delay	↓ $\overline{SO}$	An	EMPTY = Low; $\overline{SI}$ = High		40	55	ns
T_{AR} Address lead time	An	↓ READ		−1			ns
T_{DAR} $\overline{READ}$ output active delay	↓ $\overline{SO}$	↓ READ	EMPTY = Low; $\overline{SI}$ = High	30	48	65	ns
T_{DR} $\overline{READ}$ output inactive delay	↑ $\overline{SO}$	↑ $\overline{READ}$			5	10	ns
T_{RA} Address lag time	↑ $\overline{READ}$	An		20	32		ns
T_{DT} Address output disable	↑ $\overline{SO}$	An (Hi-Z)			37	60	ns
T_{DE} EMPTY status active delay	↓ $\overline{SO}$	↑ EMPTY	Stack approaching EMPTY; $\overline{SI}$ = High		38	50	ns
T_{DF} FULL status inactive delay	↓ $\overline{SO}$	↓ FULL	Stack = FULL		38	50	ns
T_{HFL} HALF-FULL status inactive delay	↑ $\overline{SO}$	↓ HALF FULL	Stack exactly HALF-FULL		54	75	ns
T_{LR} $\overline{READ}$ output active after write	↑ $\overline{SI}$	↓ $\overline{READ}$	Both $\overline{SO}$ & $\overline{WRITE}$ = Low		70	90	ns
Chip enable timing (write):							
T_{HEW} Chip enable hold time[2]	↓ $\overline{SI}$	↑ $\overline{CE}$	FULL = Low; $\overline{SO}$ = High	10	1		ns
T_{SEW} Chip disable set-up time[3]	↑ $\overline{CE}$	↓ $\overline{SI}$	FULL = Low; $\overline{SO}$ = High	10	1		ns
T_{PEW} Chip enable delay time	↓ $\overline{CE}$	↓ $\overline{WRITE}$	FULL = Low; $\overline{SI}$ = Low; $\overline{SO}$ = High		69	95	ns
Chip enable timing (read):							
T_{HER} Chip enable hold time[2]	↓ $\overline{SO}$	↑ $\overline{CE}$	EMPTY = Low; $\overline{SI}$ = High	10	1		ns
T_{SER} Chip disable set-up time[3]	↑ $\overline{CE}$	↓ $\overline{SO}$	EMPTY = Low; $\overline{SI}$ = High	10	1		ns
T_{PER} Chip enable delay time	↓ $\overline{CE}$	↓ $\overline{READ}$	EMPTY = Low; $\overline{SO}$ = Low; $\overline{SI}$ = High		64	95	ns
Reset timing:							
T_{RR} $\overline{RESET}$ recovery	↑ $\overline{RESET}$	↓ $\overline{WRITE}$	$\overline{SI}$ = Low		57	75	ns
T_{RL} $\overline{RESET}$ pulse width (low)	↓ $\overline{RESET}$	↑ $\overline{RESET}$		25	8		ns
Full/empty override timing:							
T_{FW} Override Recovery for FULL	↓ FULL	↓ $\overline{WRITE}$	Stack = Full; $\overline{SI}$ = Low; $\overline{SO}$ = High		70	95	ns
T_{ER} Override Recovery for EMPTY	↓ EMPTY	↓ $\overline{READ}$	Stack = EMPTY; $\overline{SO}$ = Low; $\overline{SI}$ = High		65	90	ns

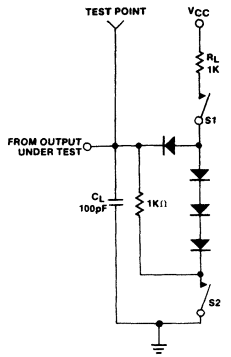
NOTES:
1. Such that write/read request is inhibited after stack becomes full/empty.
2. The earliest rising edge of $\overline{CE}$ such that the $\overline{WRITE}$ or $\overline{READ}$ output always occurs.
3. The latest rising edge of $\overline{CE}$ such that the $\overline{WRITE}$ or $\overline{READ}$ output never occurs.

TEST LOADING CIRCUITS

APPLICABLE PINS: $\overline{\text{WRITE}}$ (7), $\overline{\text{READ}}$ (8), HALF FULL (10)



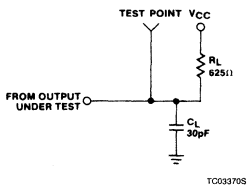
APPLICABLE PINS: A_n (15 – 20, 22 – 27)



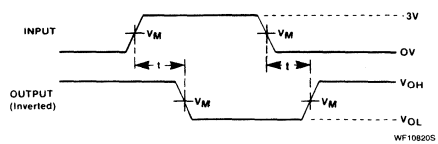
OUTPUT STATE		SWITCH POSITION	
FROM	TO	S1	S2
Low	High	Closed	Closed
High	Low	Closed	Closed
High	HI-Z	Closed	Closed
Low	HI-Z	Closed	Closed
HI-Z	High	Open	Closed
HI-Z	Low	Closed	Open

NOTES:
1. In all cases C_L includes probe and jig capacitance
2. All diodes are 1N916, 1N3064, or equivalent.
3. For $\overline{\text{READ}}$ and $\overline{\text{WRITE}}$ outputs, R_L = 280 ohms; for HALF FULL output, R_L = 2K ohms.

APPLICABLE PINS: FULL (8) AND EMPTY (11)

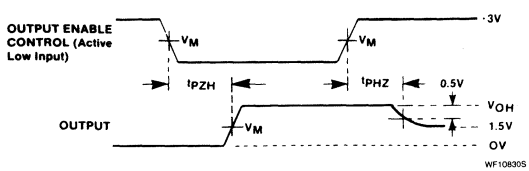


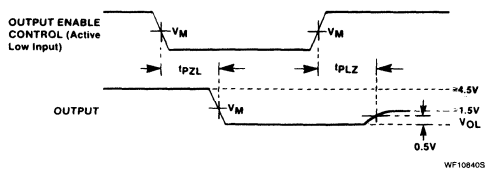
AC TEST WAVEFORMS



Propagation Delay
(Typical Example)

NOTE:
Pulse widths and Set-up/Hold times are measured using the same reference points as above waveform.



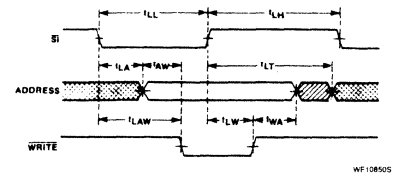


For all waveforms, V_M = 1.5V for 74 and 74S; V_M = 1.3V for 74LS.

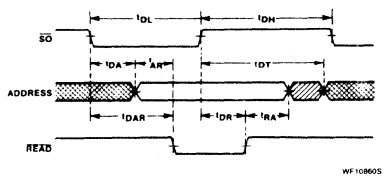
3-State Enable Time To LOW Level And Disable Time From LOW Level

3-State Enable Time To HIGH Level And Disable Time From HIGH Level

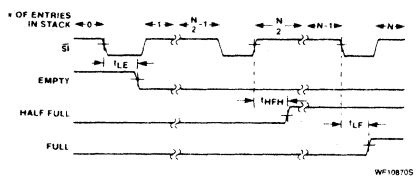
TIMING DIAGRAMS



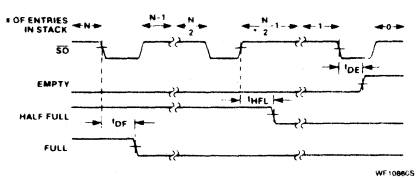
Write Cycle Timing



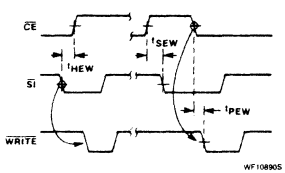
Read Cycle Timing



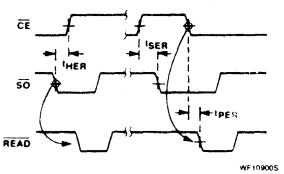
Status Output Timing-Write



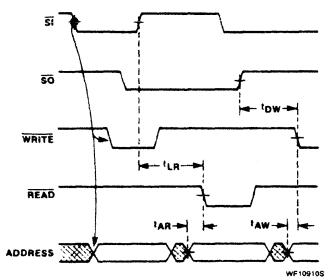
Status Output Timing-Read



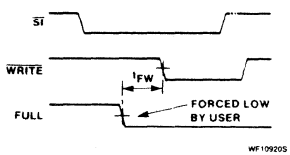
Chip Enable Timing Write*



Chip Enable Timing Read*



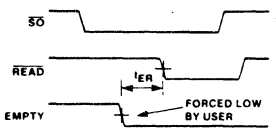
Change Of Cycle Timing



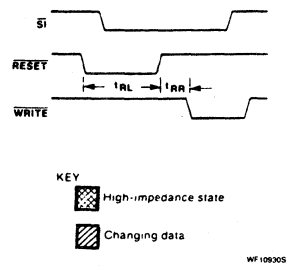
Full Override Timing

*The rising edge of $\overline{CE}$ should not occur within 10-nanoseconds before or after a falling edge of $\overline{SI}$ or $\overline{SO}$.

TIMING DIAGRAMS (Continued)

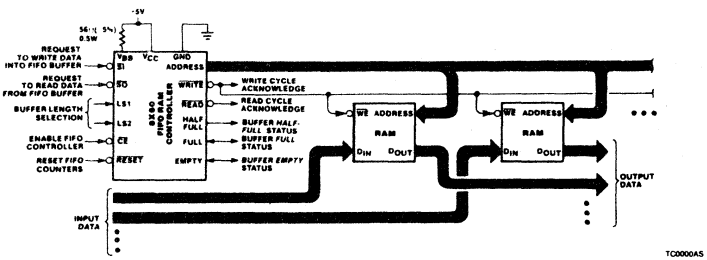


Empty Override Timing

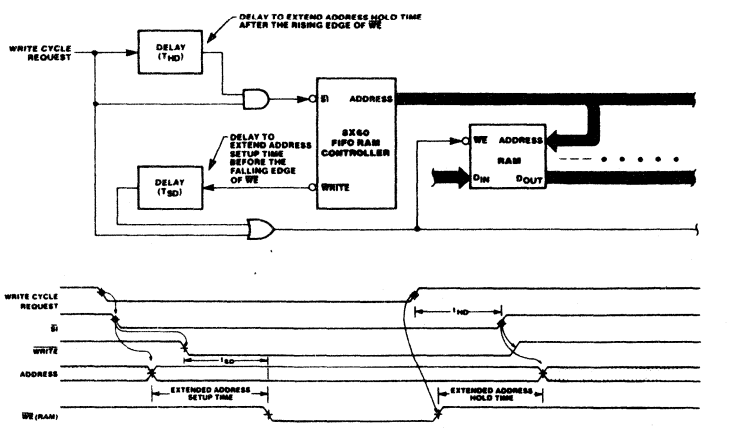


Reset Timing

APPLICATIONS



Implementation Of A FIFO Buffer Using The 8X60 And High Speed RAM



Using 8X60 With High-Density MOS RAMs

Signetics

Logic Products

2960
Error Detection and Correction (EDC) Unit

Product Specification

FEATURES

- **Boosts Memory Reliability — Corrects all single-bit errors. Detects all double and some triple-bit errors. Reliability of dynamic RAM systems is increased more than 60-fold.**
- **Very High Speed — Perfect for MOS microprocessor, minicomputer and mainframe systems.**
  - **Data in to error detect: 32ns worst case.**
  - **Data in to corrected data out: 65ns worst case.**
- **High performance systems can use the Signetics EDC in the check-only mode to avoid memory system slowdown.**
- **Replaces 25 to 50 MSI chips — All necessary features are built-in to the Signetics 2960, including diagnostics, data in, data out and check bit latches.**
- **Handles Data Words From 8 to 64 Bits — The Signetics 2960 is cascadable: 1 EDC for 8 or 16 bits, 2 for 32 bits, 4 for 64 bits.**
- **Easy Byte Operations — Separate byte enables on the data out latch simplify the steps and cuts the time required by byte writes.**
- **Built-In Diagnostics — The processor may completely exercise the EDC under software control to check for proper operation.**

DESCRIPTION

The Signetics 2960 Error Detection and Correction Unit (EDC) (Figure 1) contains the logic necessary to generate check bits on a 16-bit data field according to a modified Hamming Code, and to correct the data word when check bits are supplied. Operating on data read from memory, the EDC will correct any single bit error and will detect all double and some triple bit errors. For 16-bit words, 6 check bits are used. The 2960 can be expanded to operate on 32-bit words (7 check bits) and 64-bit words (8 check bits). In all configurations, the device makes the error syndrome available on separate outputs for data logging.

The Signetics 2960 also features two diagnostic modes, in which diagnostic data can be forced into portions of the chip to simplify device testing and to execute system diagnostic functions.

The product is supplied in a 48 lead hermetic DIP package and a 48-pin plastic package.

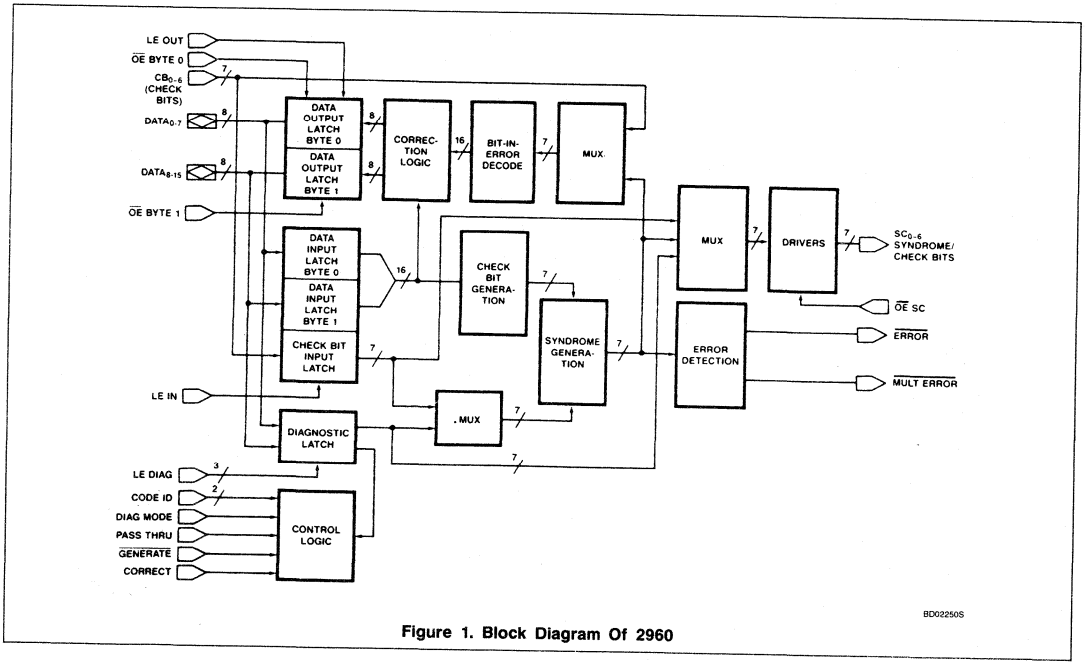
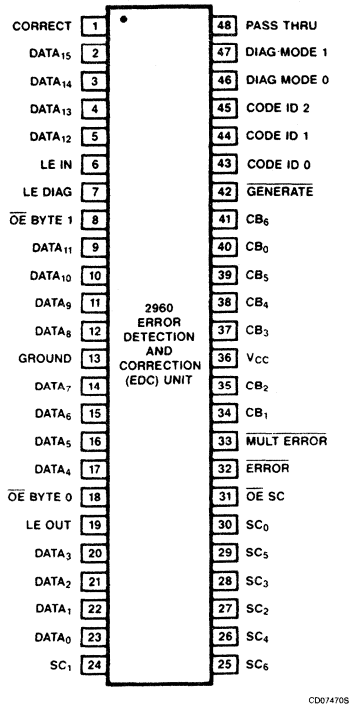


Figure 1. Block Diagram Of 2960

2960 PACKAGE AND PIN DESIGNATIONS



PIN DESCRIPTION

PIN NO.	IDENTIFIER	FUNCTION
1	Correct	**Correct input:** When HIGH this signal allows the correction network to correct any single-bit error in the Data input Latch (by complementing the bit-in-error) before putting it onto the Data Output Latch. When LOW the EDC will drive data directly from the Data input latch to the Data Output Latch without correction.
2 – 5 9 – 12 14 – 17 20 – 23	$DATA_{15-12}$ $DATA_{11-8}$ $DATA_{7-4}$ $DATA_{3-0}$	**16 Bidirectional Data Lines:** They provide input to the Data Input Latch and Diagnostic Latch, and receive output from the Data Output Latch. $DATA_0$ is the least significant bit; $DATA_{15}$ is the most significant.
6	LE IN	**Latch Enable:** Data Input Latch. Controls latching of the input data. When HIGH the Data input Latch and Check Bit Input Latch follow the input data and input check bits. When LOW, the Data Input Latch and Check Bit Input Latch are latched to their previous state.
7	LE DIAG	**Latch Enable:** Diagnostic Latch. When HIGH the Diagnostic Latch follows the 16-bit data on the input lines. When LOW the outputs of the Diagnostic Latch are latched to their previous states. The Diagnostic Latch holds diagnostic check bits, and internal control signals for CODE ID_{0-2}, DIAG $MODE_{0-1}$, CORRECT and PASS THRU.

PIN DESCRIPTION (Continued)

PIN NO.	IDENTIFIER	FUNCTION
18, 8	OE BYTE 0, OE BYTE 1	**Output Enable:** Bytes 0 and 1, Data Output Latch. These lines control the 3-state outputs for each of the two bytes of the Data Output Latch. When LOW these lines enable the Data Output Latch and when HIGH these lines force the Data Output Latch into the high impedance state. The two enable lines can be separately activated to enable only one byte of the Data Output Latch at a time.
13	GND	**Ground.**
19	LE OUT	**Latch Enable:** Data Output Latch. Controls the latching of the Data Output Latch. When LOW the Data Output Latch is latched to its previous state. When HIGH the Data Output Latch follows the output of the Data Input Latch as modified by the correction logic network. In Correct Mode, single-bit errors are corrected by the network before loading into the Data Output Latch. In Detect Mode, the contents of the Data Input Latch are passed through the correction network unchanged into the Data Output Latch. The inputs to the Data Output Latch are unspecified if the EDC is in Generate Mode.
24, 25 – 30	$SC_1S_6 - S_0$	**Syndrome/Check Bit Outputs:** These seven lines hold the check/partial-check bits when the EDC is in Generate Mode, and will hold the syndrome/partial syndrome bits when the device is in Detect or Correct Modes. These are 3-state outputs.
31	OE SC	**Output Enable:** Syndrome/Check Bits. When LOW, the 3-state output lines SC_{0-6} are enabled. When HIGH, the SC outputs are in the high impedance state.
32	ERROR	**Error Detected Output:** When the EDC is in Detect or Correct Mode, this output will go LOW if one or more syndrome bits are asserted, meaning there are one or more bit errors in the data or check bits. If no syndrome bits are asserted, there are no errors detected and the output will be HIGH. In Generate Mode, ERROR is forced HIGH. (In a 64-bit configuration, ERROR must be externally implemented.)
33	MULT ERROR	**Multiple Errors Detected Output:** When the EDC is in Detect or Correct Mode, this output if LOW indicates that there are two or more bit errors that have been detected. If HIGH this indicates that either one or no errors have been detected. In Generate mode, MULT ERROR is forced HIGH (in a 64-bit configuration, MULT ERROR must be externally implemented.)
40, 34 – 35 37 – 39, 41	CB_{0-6}	**Seven Check Bit Input Lines:** The check bit lines are used to input check bits for error detection. Also used to input syndrome bits for error correction in 32 and 64-bit configurations.
36	V_{CC}	+5V Power Supply.
42	GENERATE	**Generate Check Bits Input:** When this input is LOW the EDC is in the Check Bit Generate Mode. When HIGH the EDC is in the Detect Mode or Correct Mode. In the Generate Mode the circuit generates the check bits or partial check bits specific to the data in the Data Input Latch. The generated check bits are placed on the SC outputs. In the Detect or Correct Modes the EDC detects single and multiple errors, and generates syndrome bits based upon the contents of the Data Input Latch and Check Bit input Latch. In Correct Mode, single-bit errors are also automatically corrected - corrected data is placed at the inputs of the Data Output Latch. The syndrome result is placed on the SC outputs and indicates in a coded form the number of errors and the bit-in-error.
43 – 45	Code ID_{0-2}	**Code Identification Inputs:** These three bits identify the size of the total data word to be processed and which 16-bit slice of larger data words a particular EDC is processing. The three allowable data word sizes are 16, 32 and 64 bits and their respective modified Hamming codes are designated 16/22, 32/39 and 64/72. Special CODE ID input 001 (ID_2, ID_1, ID_0) is also used to instruct the EDC that the signals, CODE ID_{0-2}, DIAG $MODE_{0-1}$, CORRECT and PASS THRU are to be taken from the Diagnostic Latch, rather than from the input control lines.
46, 47	DIAG $MODE_{0-1}$	**Diagnostic Mode Select:** These two lines control the initialization and diagnostic operation of the EDC.
48	PASS THRU	**Pass Thru Input:** This line when HIGH forces the contents of the Check Bit Latch onto the Syndrome/Check Bit outputs (SC_{0-6}) and the unmodified contents of the Data Input Latch onto the inputs of the Data Output Latch.

ARCHITECTURAL SUMMARY

The EDC Unit is a powerful 16-bit cascadable slice used for check bit generation, error detection, error correction and diagnostics.

As shown in Figure 1, the device consists of the following:

- Data Input Latch
- Check Bit Input Latch
- Check Bit Generation Logic
- Syndrome Generation Logic
- Error Detection Logic

Error Detection and Correction (EDC) Unit 2960

Table 1. Hamming Code and Slice Identification

CODE ID_2	CODE ID_1	CODE ID_0	HAMMING CODE AND SLICE SELECTED
0	0	0	Code 16/22
0	0	1	Internal Control Mode
0	1	0	Code 32/39, Bytes 0 and 1
0	1	1	Code 32/39, Bytes 2 and 3
1	0	0	Code 64/72, Bytes 0 and 1
1	0	1	Code 64/72, Bytes 2 and 3
1	1	0	Code 64/72, Bytes 4 and 5
1	1	1	Code 64/72, Bytes 6 and 7

Table 2. EDC Operating Modes

OPERATING MODE	DIAGNOSTIC MODE**		$\overline{GENERATE}$	
	DM_1	DM_0	0	1
Normal	0	0	Generate	Correct*
Diagnostic Generate	0	1	Diagnostic Generate	Correct*
Diagnostic Correct	1	0	Generate	Diagnostic Correct*
Initialize	1	1	Initialize	Initialize
Pass Thru	When PASS THRU is asserted the Operating Mode is defaulted to the Pass Thru Mode.			

*Correct if the CORRECT Input is HIGH. Detect if the CORRECT Input is LOW.
**In Code ID_{2-0} 001 (ID_2, ID_1, ID_0) DM_1 and DM_0 are taken from the Diagnostic Latch.

- Error Correction Logic
- Data Output Latch
- Diagnostic Latch
- Control Logic

Data Input Latch

16 bits of data are loaded from the bidirectional DATA lines under control of the Latch Enable input, LE IN. Depending on the control mode the input data is either used for check bit generation or error detection/correction.

Check Bit Input Latch

Seven check bits are loaded under control of LE IN. Check bits are used in the Error Detection and Error Correction modes.

Check Bit Generation Logic

This block generates the appropriate check bit for the 16 bits of data in the Data Input Latch. The check bits are generated according to a modified Hamming code.

Syndrome Generation Logic

In both Error Detection and Error Correction modes, this logic block compares the check bits read from memory against a newly generated set of check bits produced for the data read in from memory. If both sets of check bits match, then there are no errors. If there is a mismatch, then one or more of the data or check bits is in error.

The syndrome bits are produced by an exclusive-OR of the two sets of check bits. If the two sets of check bits are identical (meaning there are no errors) the syndrome bits will be all zeroes. If there are errors, the syndrome bits can be decoded to determine the number of errors and the bit-in-error.

Error Detection Logic

This section decodes the syndrome bits generated by the Syndrome Generation Logic. If there are no errors in either the input datas or check bits, the ERROR and MULT ERROR outputs remain HIGH. If one or more errors are detected, ERROR goes LOW. If two or more errors are detected, both ERROR and MULT ERROR go LOW.

Error Correction Logic

For single errors, the Error Correction Logic complements (corrects) the single data bit in error. This corrected data is loadable into the Data Output Latch, which can then be read onto the bidirectional data lines. If the single error is one of the check bits, the correction logic does not place corrected check bits on the syndrome/check bit outputs. If the corrected check bits are needed the EDC must be switched to Generate Mode.

Data Output Latch

The Data Output Latch is used for storing the result of an error correction operation. The latch is loaded from the correction logic under control of the Data Output Latch Enable, LE OUT. The Data Output Latch may also be loaded directly from the Data Input Latch under control of the PASS THRU control input.

The Data Output Latch is split into two 8-bit (byte) latches which may be enabled independently for reading onto the bidirectional data lines.

Diagnostic Latch

This is a 16-bit latch loadable from the bidirectional data lines under control of the Diagnostic Latch Enable, LE DIAG. The Diagnostic Latch contains check bit information in one byte and control information in the other byte. The Diagnostic Latch is used for driving the device when in Internal Control Mode, or for supplying check bits when in one of the Diagnostic Modes.

Control Logic

The control logic determines the specific operating mode of the device. Normally the control logic is driven by external control inputs. However, in Internal Control Mode, the control signals are read from the Diagnostic Latch.

FUNCTIONAL OPERATION

The EDC contains the logic necessary to generate check bits on a 16-bit data field according to a modified Hamming code (Table 1). Operating on data read from memory, the EDC will correct any single-bit error and will detect all double and some triple-bit errors. The EDC can be configured to operate on 16-bit data words (with 6 check bits), 32-bit data words (with 7 check bits) and 64-bit data words (with 8 check bits). In fact, the EDC can be configured to work on data words from 8 to 64 bits. In all configurations, the device makes the error syndrome bits available on separate outputs for error data logging.

Code and Byte Specification

The EDC may be configured in several different ways and operates differently in each configuration. It is necessary to indicate to the device what size data word is involved and which bytes of the data word it is processing. This is done with input lines CODE ID_{0-2}, as shown in Table 1; the three modified Hamming codes are defined below:

- 16/22 — 16 data bits, 6 check bits, 22 bits in total.
- 32/39 — 32 data bits, 7 check bits, 39 bits in total.
- 64/72 — 64 data bits, 8 check bits, 72 bits in total.

CODE ID input 001 (ID_2, ID_1, ID_0) is a special code, described later, used to operate the device in Internal Control Mode.

Table 3. Diagnostic Mode Control

DIAG $MODE_1$	DIAG $MODE_0$	DIAGNOSTIC MODE SELECTED
0	0	**Non-diagnostic mode:** The EDC functions normally in all modes.
0	1	**Diagnostic Generate:** The contents of the Diagnostic Latch are substituted for the normally generated check bits when in the Generate Mode. The EDC functions normally in the Detect or Correct modes.
1	0	**Diagnostic Detect/Correct:** In the Detect or Correct Mode, the contents of the Diagnostic Latch are substituted for the check bits normally read from the Check Bit Input Latch. The EDC functions normally in the Generate Mode.
1	1	**Initialize:** The outputs of the Data Input Latch are forced to zeroes (and latched upon removal of the Initialize Mode) and the check bits generated correspond to the all-zero data.

Table 4. 16-Bit Modified Hamming Code

GENERATED CHECK BITS	PARITY	PARTICIPATING DATA BITS															
		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
CX	Even (XOR)		X	X	X		X			X	X		X			X	
C0	Even (XOR)	X	X	X		X		X		X		X		X			
C1	Odd (XNOR)	X			X	X			X		X	X			X		X
C2	Odd (XNOR)	X	X				X	X	X				X	X	X		
C4	Even (XOR)			X	X	X	X	X	X							X	X
C8	Even (XOR)									X	X	X	X	X	X	X	X

NOTE:
The check bit is generated as either an XOR or XNOR of the eight data bits noted by an "X" in the table.

Control Mode Selection

The device control lines are GENERATE, CORRECT, PASS THRU, DIAG $MODE_{0-1}$ and CODE ID_{0-2}. Table 2 indicates the operating modes selected by various combinations of the control line inputs.

Diagnostics

Table 3 shows specifically how DIAG $MODE_{0-1}$ select between normal operation, initialization and one of two diagnostic modes. The Diagnostic Modes allow the user to operate the EDC under software control in order to verify proper functioning of the device.

Check and Syndrome Bit Labelling

The check bits generated in the EDC are designed as follows:

- 16-bit configuration - CX, C0, C1, C2, C4, C8;
- 32-bit configuration - CX, C0, C1, C2, C4, C8, C16;
- 64-bit configuration - CX, C0, C1, C2, C4, C8, C16, C32.

Syndrome bits are similarly labeled SX through S32. There are only 6 syndrome bits in the 16-bit configuration, 7 for 32 bits and 8 syndrome bits in the 64-bit configuration.

The error correction code can be selected independent of the processor with the exception of diagnostic software.

Diagnostic software run by a processor to checkout the EDC system must know specifically which code is being used. This is only a problem when the EDC replaces an existing MSI implementation on an existing computer. In this case, the computer's software must first determine which of two codes (the old one used by the MSI implementation or the new one used by the EDC) is used by the computer's memory system.

This is easily determined by writing a test data word into memory and then examining whether the generated check bits are typical of the old or the new code. From then on the software runs only the diagnostic appropriate for the code used on that particular computer's memory system.

Initialize Mode

The inputs of the Data Output Latch are forced to zeroes. The check bit outputs (SC) are generated to correspond to the all-zero data. ERROR and MULT ERROR are forced HIGH in the Initialize Mode.

Initialize Mode is useful after power up when RAM contents are random. The EDC may be placed in initialize mode and its outputs written in to all memory locations by the processor.

Code Selections

The Signetics 2960 EDC uses a modified Hamming Code which provides the following functions:

- Cascading of EDC Units
- Detection of all double-bit errors
- Detection of gross error conditions (all "0s" or all "1s").

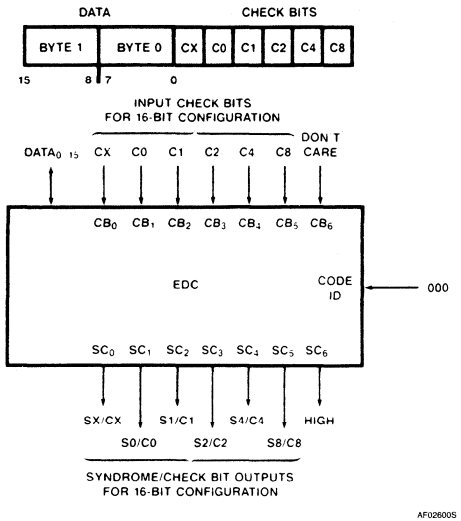


Figure 2. 16-Bit Data Format And I/O

16-Bit Data Word Configuration

The 16-bit format consists of 16 data bits, six check bits and, as previously indicated, is designated as the 16/22 code. The data format and I/O configuration for a 16-bit word is shown in Figure 2.

Generate Mode

In this mode check bits will be generated that correspond to the contents of the Data Input Latch. The check bits generated are placed on the outputs SC_{0-5} (SC_6 is a logical one, or high).

Check bits are generated according to a modified Hamming code. Details of the code for check bit generation are contained in Table 4.

Each check bit is generated as either an XOR or XNOR of eight of the 16 data bits as indicated in the table. The XOR function results in an even parity check bit; the XNOR is an odd parity check bit. Data flow for the Generate Mode is shown in Figure 3.

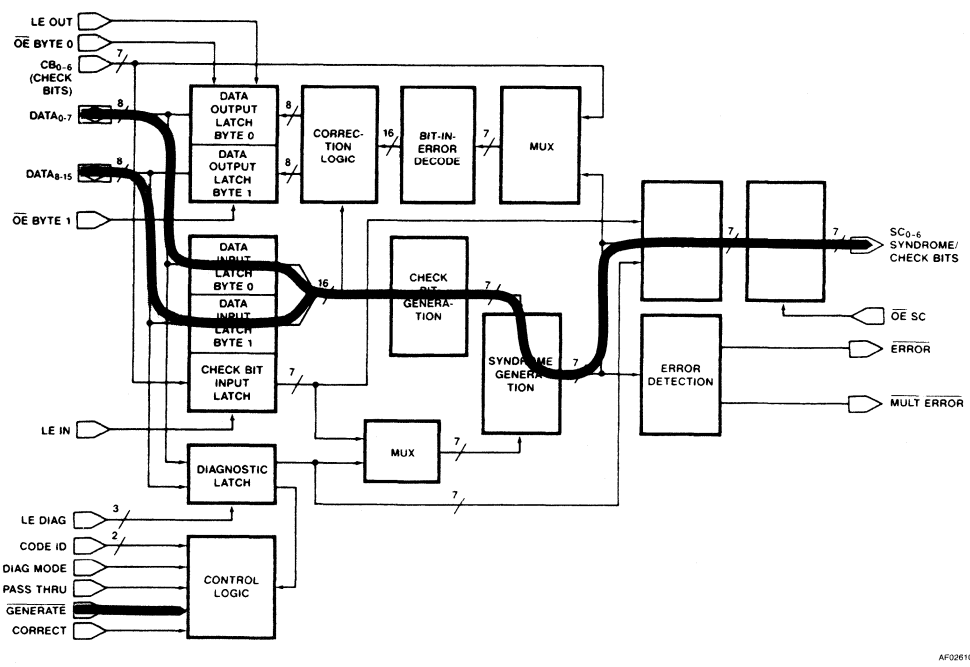


Figure 3. Check Bit Generation

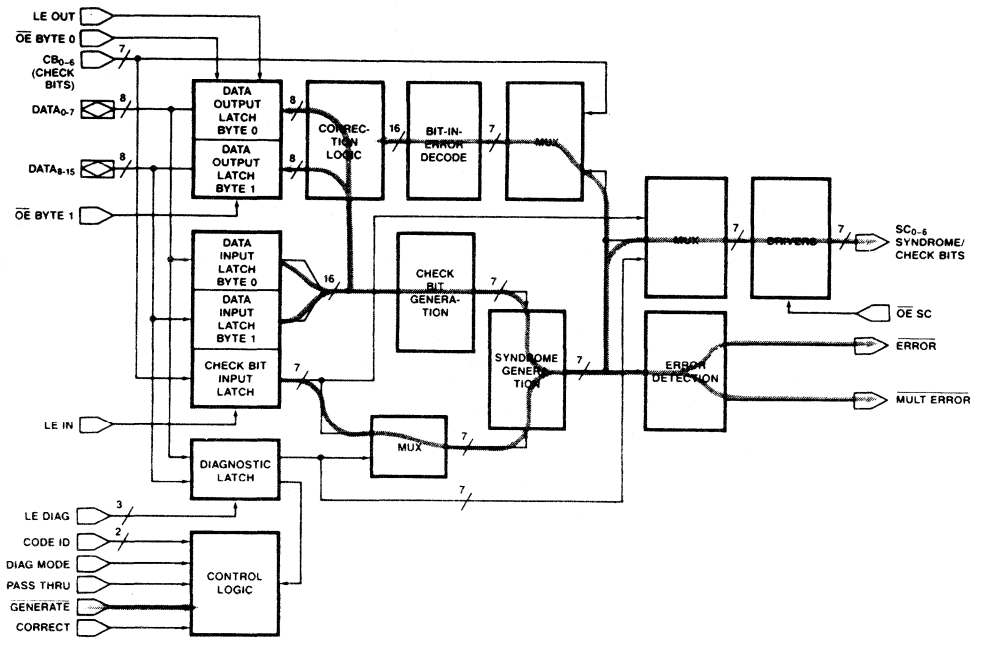


Figure 4. Error Detection And Correction

Table 5. Syndrome Decode to Bit-In-Error

SYNDROME BITS SX	S0	S1		S8	0	1	0	1	0	1	0	1
				S4	0	0	1	1	0	0	1	1
				S2	0	0	0	0	1	1	1	1
0	0	0			*	C8	C4	T	C2	T	T	M
0	0	1			C1	T	T	15	T	13	7	T
0	1	0			C0	T	T	M	T	12	6	T
0	1	1			T	10	4	T	0	T	T	M
1	0	0			CX	T	T	14	T	11	5	T
1	0	1			T	9	3	T	M	T	T	M
1	1	0			T	8	2	T	1	T	T	M
1	1	1			M	T	T	M	T	M	M	T

* — no errors detected
Number — the location of the single bit-in-error
T — two errors detected
M — three or more errors detected

Detect Mode

In this mode the device examines the contents of the Data Input Latch against the Check Bit Input Latch, and will detect all single-bit errors, all double-bit errors and some triple-bit errors. If one or more errors are detected, ERROR goes LOW. If two or more errors are detected, MULT ERROR goes LOW. Both error indicators are HIGH if there are no errors.

Also available on device outputs SC_{0-5} are the syndrome bits generated by the error detection step. The syndrome bits may be decoded to determine if a bit error was detected and, for single-bit errors, which of the data or check bits is in error. Table 5 provides decoding data for the syndrome bits generated by the 16-bit configuration (as an example, if the syndrome bits SX/S0/S1/S2/S4/S8 were 101001 this would be decoded to indicate that there is a single-bit error at data bit 9). If no error is detected the syndrome bits will all be zeroes.

In Detect Mode, the contents of the Data Input Latch are driven directly to the inputs of the Data Output Latch without correction.

Table 6. Diagnostic Latch Loading

DATA BIT	INTERNAL FUNCTION
0	Diagnostic Check Bit X
1	Diagnostic Check Bit 0
2	Diagnostic Check Bit 1
3	Diagnostic Check Bit 2
4	Diagnostic Check Bit 4
5	Diagnostic Check Bit 8
6, 7	Don't Care
8	CODE ID 0
9	CODE ID 1
10	CODE ID 2
11	DIAG MODE 0
12	DIAG MODE 1
13	CORRECT
14	PASS THRU
15	Don't Care

Correct Mode

In this mode, the EDC functions the same as in Detect Mode except that the correction network is allowed to correct (complement) any single-bit error of the Data Input Latch before putting it onto the inputs of the Data Output Latch — see Figure 4. If multiple errors are detected, the output of the correction network is unspecified. If the single-bit error is a check bit there is no automatic correction. If check bit correction is desired, this can be done by placing the device in Generate Mode to produce a correct check bit sequence for the data in the Data Input Latch.

Pass Thru Mode

In this mode, the unmodified contents of the Data Input Latch are placed on the inputs of the Data Output Latch and the contents of the Check Bit Input Latch are placed on outputs SC_{0-5}. ERROR and MULT ERROR are forced HIGH in this mode.

Diagnostic Latch

The diagnostic Latch serves both for diagnostic uses and internal control uses. It is loaded from the DATA lines under the control of LE DIAG. Table 6 shows the loading definitions for the DATA lines.

Diagnostic Generate/Detect/ Correct

These are special diagnostic modes selected by DIAG $MODE_{0-1}$ where either normal check bit inputs or outputs are substituted for by check bits loaded into the Diagnostic Latch — See Table 2 for details. Figures 5 and 6 illustrate the flow of data during the two diagnostic modes.

Internal Control Mode

This mode is selected by CODE ID_{0-2} input 001 (ID_2, ID_1, ID_0).

When in Internal Control Mode, the EDC takes the CODE ID_{0-2}, DIAG $MODE_{0-1}$, CORRECT, and PASS THRU control signals from the internal Diagnostic Latch rather than from the external input lines.

Table 6 gives the format for loading the Diagnostic Latch.

32-Bit Data Word Configuration

The 32-bit format consists of 32 data bits, seven check bits and, as previously indicated, is designated as the 32/39 code. The data format and I/O configuration for a 32-bit word is shown in Figure 7.

The upper EDC (Slice 0/1) handles the least significant bytes 0 and 1 — the external DATA lines 0 to 15 are connected to the same numbered inputs of the upper device. The lower EDC (Slice 2/3) handles the most significant bytes 2 and 3 — the external DATA lines for bits 16 to 31 are connected to inputs $DATA_0$ through $DATA_{15}$ respectively.

The valid syndrome and check bit outputs are those of Slice 2/3 as shown in the diagram. In Correct Mode these must be read into Slice 0/1 via the CB inputs and are selected by the MUX as inputs to the bit-in-error decoded (see block diagram), thus requiring external buffering and output enabling of the check bit lines as shown. The OE SC signal can be used to control enabling of check bit inputs - when syndrome outputs are enabled, the external check bit inputs will be disabled.

The valid ERROR and MULT ERROR outputs are those of the Slice 2/3. The ERROR and MULT ERROR outputs of Slice 0/1 are unspecified. All of the latch enables and control signals must be input to both of the devices.

Generate Mode

In this mode check bits will be generated that correspond to the contents of the Data Input Latch. The check bits generated are placed on the outputs SC_{0-6} of Slice 2/3.

Check bits are generated according to a modified Hamming code. Details of the code for check bit generation are contained in Table 7. Check bits are generated as either an XOR or XNOR or 16 of the 32 data bits as indicated in the table. The XOR function results in an even parity check bit, the XNOR in an odd parity check bit.

Detect Code

In this mode the device examines the contents of the Data Input Latch against the Check Bit Input Latch, and will detect all single-bit errors, all double-bit errors and some triple-bit errors. If one or more errors are detected, ERROR goes LOW. If two or more errors are detected MULT ERROR goes LOW. Both error indicators are HIGH if there are no errors. The valid ERROR and MULT ERROR signals are those of Slice 2/3 — those of Slice 0/1 are undefined.

Also available on Slice 2/3 outputs $SC_{0/6}$ are the syndrome bits generated by the error detection step. The syndrome bits may be decoded to determine if a bit error was detected and, for single-bit errors, which of the data or check bits is in error. Table 8 gives the chart for decoding the syndrome bits generated for the 32-bit configuration (as an example, if the syndrome bits SX/S0/S1/S2/S4/S8/S16 were 0010011 this would be decoded to indicate that there is a single-bit error at data bit 25). If no error is detected the syndrome bits will be all zeroes.

In Detect Mode, the contents of the Data Input Latch are driven directly to the inputs of the Data Output Latch without corrections.

Correct Mode

In this mode, the EDC functions the same as in Detect Mode except that the correction network is allowed to correct (complement) any single-bit error of the Data Input Latch before putting it onto the inputs of the Data Output Latch. If multiple errors are detected, the output of the correction network is unspecified. If the single-bit error is a check bit there is no automatic correction — if desired this would be done by placing the device in Generate Mode to produce a correct check bit sequence for the data in the Data Input Latch.

For data correction, both Slices 0/1 and 2/3 require access to the syndrome bits on Slice 2/3's outputs SC_{0-6}. Slice 2/3 has access to these syndrome bits through internal data paths, but for Slice 0/1 they must be read through the inputs CB_{0-6}. The device connections for this are shown in Figure 7. When in Correct Mode the SC outputs must be enabled so that they are available for reading in through the CB inputs.

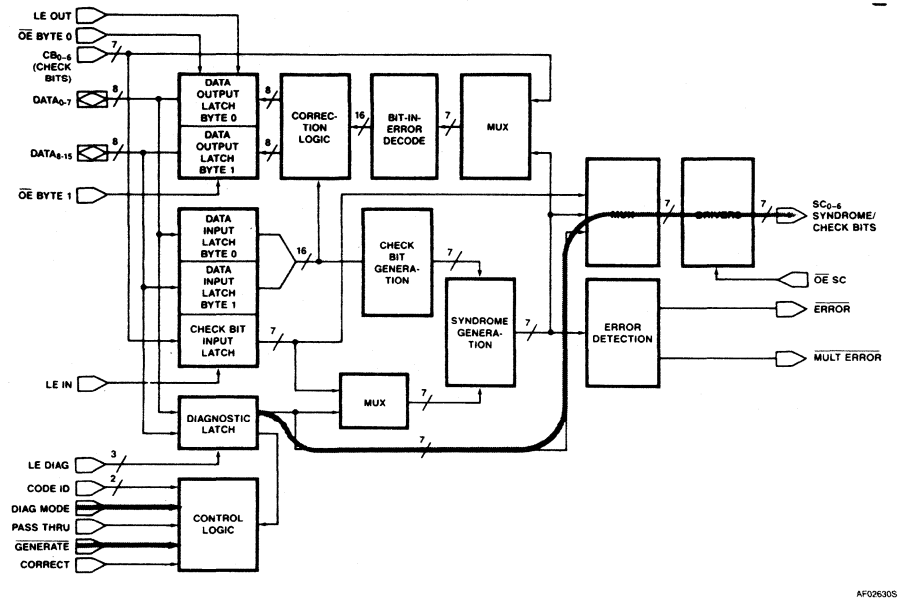


Figure 5. Diagnostic Check Bit Generation

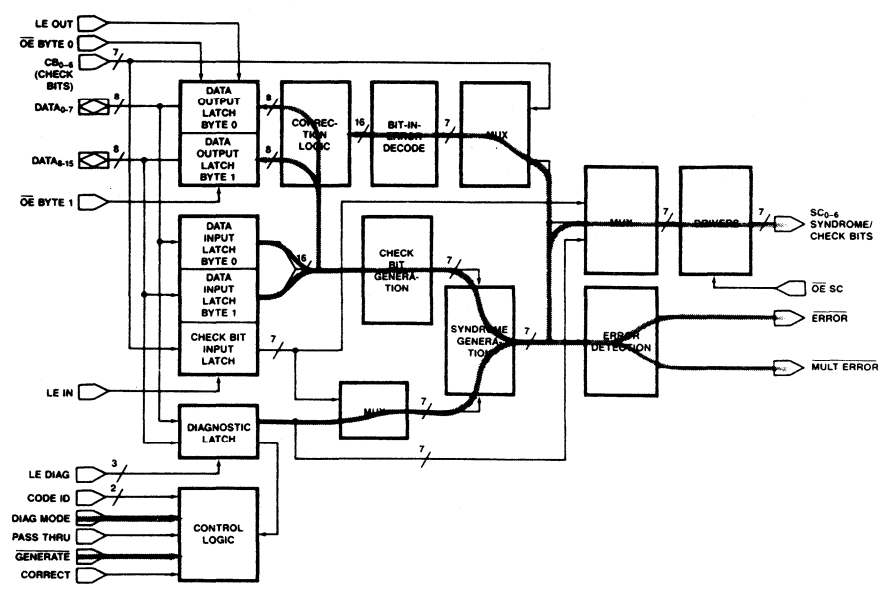


Figure 6. Diagnostic Detect And Correct

Table 7. 32-Bit Modified Hamming Code

GENERATED CHECK BITS	PARITY	PARTICIPATING DATA BITS															
		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
CS	Even (XOR)	X				X		X	X	X	X		X			X	
C0	Even (XOR)	X	X	X		X		X		X		X		X			
C1	Odd (XNOR)	X			X	X			X		X	X			X		X
C2	Odd (XNOR)	X	X				X	X	X				X	X	X		
C4	Even (XOR)			X	X	X	X	X	X							X	X
C8	Even (XOR)									X	X	X	X	X	X	X	X
C16	Even (XOR)	X	X	X	X	X	X	X	X								

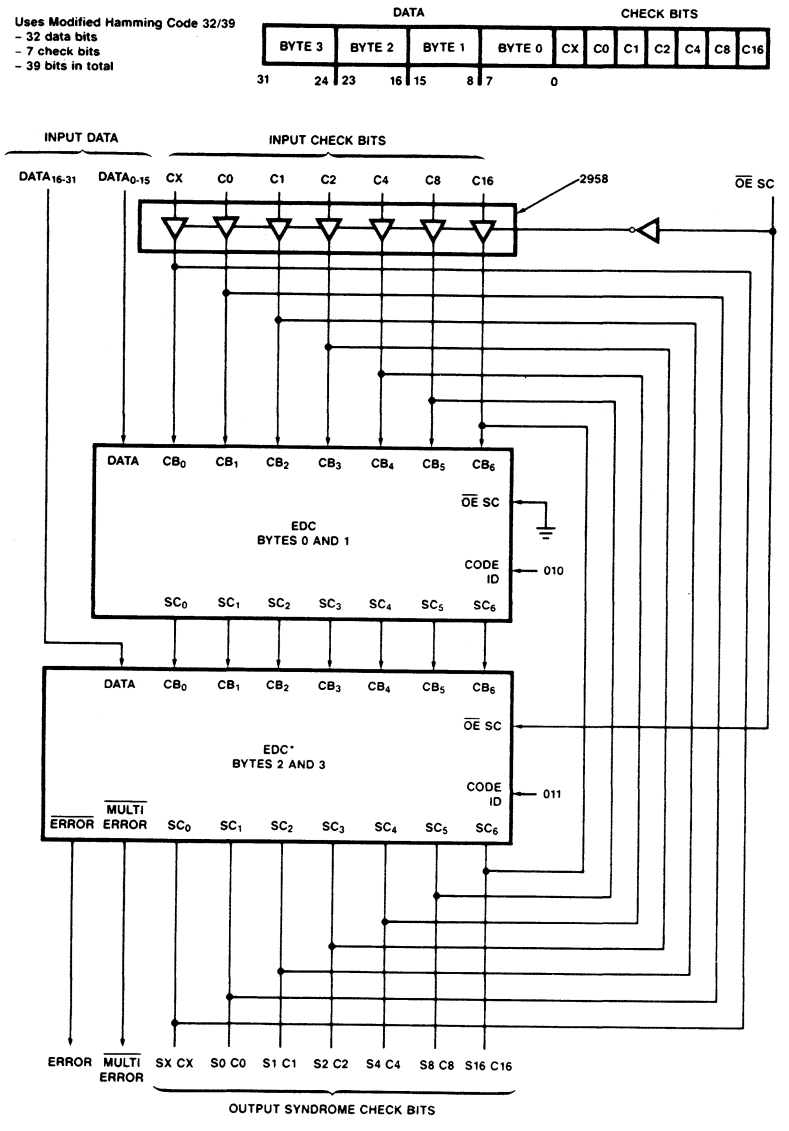
GENERATED CHECK BITS	PARITY	PARTICIPATING DATA BITS															
		16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
CS	Even (XOR)		X	X	X		X					X		X	X		X
C0	Even (XOR)	X	X	X		X		X		X		X		X			
C1	Odd (XNOR)	X			X	X			X		X	X			X		X
C2	Odd (XNOR)	X	X				X	X	X				X	X	X		
C4	Even (XOR)			X	X	X	X	X	X							X	X
C8	Even (XOR)									X	X	X	X	X	X	X	X
C16	Even (XOR)									X	X	X	X	X	X	X	X

NOTE:
The check bit is generated as either an XOR or XNOR of the sixteen data bits noted by an "X" in the table.

Table 8. Syndrome Decode to Bit-In-Error

SYNDROME BITS															
				S16	0	1	0	1	0	1	0	1
				S8	0	0	1	1	0	0	1	1
				S4	0	0	0	0	1	1	1	1
SX	S0	S1	S2									
0	0	0	0		*	C16	C8	T	C4	T	T	30
0	0	0	1		C2	T	T	27	T	5	M	T
0	0	1	0		C1	T	T	25	T	3	15	T
0	0	1	1		T	M	13	T	23	T	T	M
0	1	0	0		C0	T	T	24	T	2	M	T
0	1	0	1		T	1	12	T	22	T	T	M
0	1	1	0		T	M	10	T	20	T	T	M
0	1	1	1		16	T	T	M	T	M	M	T
1	0	0	0		CX	T	T	M	T	M	14	T
1	0	0	1		T	M	11	T	21	T	T	M
1	0	1	0		T	M	9	T	19	T	T	31
1	0	1	1		M	T	T	29	T	7	M	T
1	1	0	0		T	M	8	T	18	T	T	M
1	1	0	1		17	T	T	28	T	6	M	T
1	1	1	0		M	T	T	26	T	4	M	T
1	1	1	1		T	0	M	T	M	T	T	M

* — no errors detected
Number — the location of the single bit-in-error
T — two errors detected
M — three or more errors detected



*Check Bit Latch is forced transparent in this CODE ID combination for this slice.

Figure 7. 32-Bit Data Format And I/O

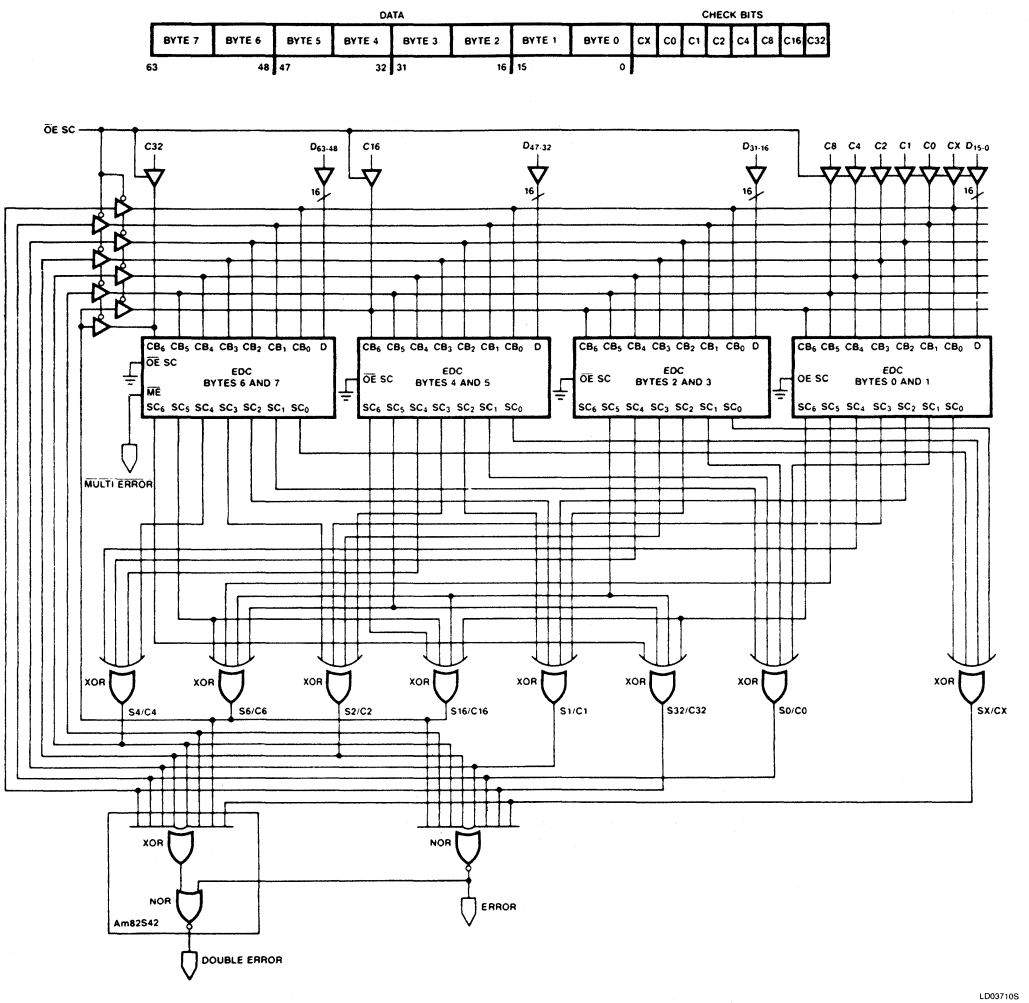


Figure 8. 64-Bit Data Format and I/O Configuration

Table 9. Diagnostic Latch Loading

DATA BIT	INTERNAL FUNCTION
0	Diagnostic Check Bit X
1	Diagnostic Check Bit 0
2	Diagnostic Check Bit 1
3	Diagnostic Check Bit 2
4	Diagnostic Check Bit 4
5	Diagnostic Check Bit 8
6	Diagnostic Check Bit 16
7	Don't Care
8	Slice 0/1 — CODE ID 0
9	Slice 0/1 — CODE ID 1
10	Slice 0/1 — CODE ID 2
11	Slice 0/1 — DIAG MODE 0
12	Slice 0/1 — DIAG MODE 1
13	Slice 0/1 — CORRECT
14	Slice 0/1 — PASS THRU
15	Don't Care
16 – 23	Don't Care
24	Slice 2/3 — CODE ID 0
25	Slice 2/3 — CODE ID 1
26	Slice 2/3 — CODE ID 2
27	Slice 2/3 — DIAG MODE 0
28	Slice 2/3 — DIAG MODE 1
29	Slice 2/3 — CORRECT
30	Slice 2/3 — PASS THRU
31	Don't Care

Pass Thru Mode

In this mode, the unmodified contents of the Data Input Latch are placed on the inputs of the Data Output Latch and the contents of the Check Bit Input Latch are placed on outputs SC_{0-6} of Slice 2/3. ERROR and MULT ERROR are forced HIGH in this mode.

Diagnostic Latches/AC Calculations

Table 9 shows how the latches (Slice 1 and Slice 2) are loaded for code 32/39 (32-bit format). Table 10 shows key AC parameters for the 32-bit configuration.

64-Bit Data Word Configuration

The 64-bit format consists of 64 data bits, eight check bits and, as previously indicated, is designated as the 64/72 code. The data format and I/O configuration for a 64-bit word is shown in Figure 8.

The configuration to process the 64-bit format is similar to that shown in Figure 2. In this configuration a portion of the syndrome generation and error detection is implemented externally of the EDCs in MSI. For error correction the syndrome bits generated must be read back into all four EDCs through the CB inputs. This necessitates the check bit buffering shown in Figure 8. The OE SC signal can control the check bit enabling — when syndrome bit outputs are enabled the external check bit lines will be disabled so that the syndrome bits may be read onto the CB inputs.

The error detection signals for the 64-bit configuration differ from the 16 and 32-bit configurations. The ERROR signal functions the same: it is LOW if one or more errors are detected, and HIGH if no errors are detected.

The DOUBLE ERROR signal is HIGH if and only if a double-bit error is detected — it is LOW otherwise. All of the MULT ERROR outputs of the four devices are valid. MULT ERROR is LOW for all three ERROR cases and some DOUBLE ERROR combinations — See Table 15. It is HIGH if either zero or one errors are detected.

This is a different meaning for MULT ERROR than in other configurations.

Generate Mode

In this mode check bits will be generated that correspond to the contents of the Data Input Latch. The check bits generated appear at the outputs of the XOR gates as indicated in Figure 8.

Check bits are generated according to a modified Hamming code. Details of the code for check bit generation are contained in Table 11. Check bits are generated as either an XOR or XNOR of 32 of the 64 bits as indicated in the table. The XOR function results in an even parity check bit, the XNOR in an odd parity check bit.

Table 10. Key AC Calculations for the 32-Bit Configuration

32-BIT PROPAGATION DELAY		COMPONENT DELAY FROM 2960 AC SPECIFICATIONS, TABLE C
From	To	
DATA	Check Bits Out	(DATA to SC) + (CB to SC, CODE ID 011)
DATA In	Corrected DATA Out	(DATA to SC) + (CB to SC, CODE ID 011) + (CB to DATA, CODE ID 010)
DATA	Syndromes Out	(DATA to SC) + (CB to SC, CODE ID 011)
DATA	$\overline{\text{ERROR}}$ for 32 Bits	(DATA to SC) + (CB to $\overline{\text{ERROR}}$, CODE ID 011)
DATA	$\overline{\text{MULT ERROR}}$ for 32 Bits	(DATA to SC) + (CB to $\overline{\text{MULT ERROR}}$, CODE ID 011)

Table 11. 64-Bit Modified Hamming Code Check Bit Encoding

GENERATED CHECK BITS	PARITY	PARTICIPATING DATA BITS															
		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
CX	Even (XOR)		X	X	X		X			X	X		X			X	
C0	Even (XOR)	X	X	X		X		X		X		X		X			
C1	Odd (XNOR)	X			X	X			X		X	X			X		X
C2	Odd (XNOR)	X	X				X	X	X				X	X	X		
C4	Even (XOR)			X	X	X	X	X	X							X	X
C8	Even (XOR)									X	X	X	X	X	X	X	X
C16	Even (XOR)	X	X	X	X	X	X	X	X								
C32	Even (XOR)	X	X	X	X	X	X	X	X								

GENERATED CHECK BITS	PARITY	PARTICIPATING DATA BITS															
		16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
CX	Even (XOR)		X	X	X		X			X	X		X			X	
C0	Even (XOR)	X	X	X		X		X		X		X		X			
C1	Odd (XNOR)	X			X	X			X		X	X			X		X
C2	Odd (XNOR)	X	X				X	X	X				X	X	X		
C4	Even (XOR)			X	X	X	X	X	X							X	X
C8	Even (XOR)									X	X	X	X	X	X	X	X
C16	Even (XOR)									X	X	X	X	X	X	X	X
C32	Even (XOR)									X	X	X	X	X	X	X	X

GENERATED CHECK BITS	PARITY	PARTICIPATING DATA BITS															
		32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
CX	Even (XOR)	X				X		X	X			X		X	X		X
C0	Even (XOR)	X	X	X		X		X		X		X		X			
C1	Odd (XNOR)	X			X	X			X		X	X			X		X
C2	Odd (XNOR)	X	X				X	X	X				X	X	X		
C4	Even (XOR)			X	X	X	X	X	X							X	X
C8	Even (XOR)									X	X	X	X	X	X	X	X
C16	Even (XOR)	X	X	X	X	X	X	X	X								
C32	Even (XOR)									X	X	X	X	X	X	X	X

GENERATED CHECK BITS	PARITY	PARTICIPATING DATA BITS															
		48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
CX	Even (XOR)	X				X		X	X			X		X	X		X
C0	Even (XOR)	X	X	X		X		X		X		X		X			
C1	Odd (XNOR)	X			X	X			X		X	X			X		X
C2	Odd (XNOR)	X	X				X	X	X				X	X	X		
C4	Even (XOR)			X	X	X	X	X	X							X	X
C8	Even (XOR)									X	X	X	X	X	X	X	X
C16	Even (XOR)									X	X	X	X	X	X	X	X
C32	Even (XOR)	X	X	X	X	X	X	X	X								

NOTE:
The check bit is generated as either an XOR or XNOR of the 32 data bits noted by an "X" in the table.

Table 12. Syndrome Decode to Bit-In-Error

SYNDROME BITS																		
				S32	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
				S16	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
				S8	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
				S4	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
SX	S0	S1	S2																	
0	0	0	0		*	C32	C16	T	C8	T	T	M	C4	T	T	M	T	46	62	T
0	0	0	1		C2	T	T	M	T	43	59	T	T	53	37	T	M	T	T	M
0	0	1	0		C1	T	T	M	T	41	57	T	T	51	35	T	15	T	T	31
0	0	1	1		T	M	M	T	13	T	T	29	23	T	T	7	T	M	M	T
0	1	0	0		C0	T	T	M	T	40	56	T	T	50	34	T	M	T	T	M
0	1	0	1		T	49	33	T	12	T	T	28	22	T	T	6	T	M	M	T
0	1	1	0		T	M	M	T	10	T	T	26	20	T	T	4	T	M	M	T
0	1	1	1		16	T	T	0	T	M	M	T	T	M	M	T	M	T	T	M
1	0	0	0		CX	T	T	M	T	M	M	T	T	M	M	T	14	T	T	30
1	0	0	1		T	M	M	T	11	T	T	27	21	T	T	5	T	M	M	T
1	0	1	0		T	M	M	T	9	T	T	25	19	T	T	3	T	47	63	T
1	0	1	1		M	T	T	M	T	45	61	T	T	55	39	T	M	T	T	M
1	1	0	0		T	M	M	T	8	T	T	24	18	T	T	2	T	M	M	T
1	1	0	1		17	T	T	1	T	44	60	T	T	54	38	T	M	T	T	M
1	1	1	0		M	T	T	M	T	42	58	T	T	52	36	T	M	T	T	M
1	1	1	1		T	48	32	T	M	T	T	M	M	T	T	M	T	M	M	T

* — no errors detected
Number — the location of the single bit-in-error
T — two errors detected
M — three or more errors detected

Detect Mode

In this mode the device compares the contents of the Data Input Latch against the contents of the Check Bit Input Latch and will detect all single-bit errors, all double-bit errors and some triple-bit errors. If one or more errors are detected, ERROR goes LOW. If exactly two errors are detected, DOUBLE ERROR goes HIGH. If three or more errors are detected, MULT ERROR goes LOW - the MULT ERROR output of any of the four EDCs may be used.

Available as XOR gate outputs are the generated syndrome bits — see Figure 8. The syndrome bits may be decoded to determine if a bit error was detected and for single-bit errors, which of the data or check bits is in error. Table 12 gives the chart for encoding the syndrome bits generated for the 64-bit configuration (as an example, if the syndrome bits SX/S1/S2/S4/S8/S16/S32 were 00100101 this would be decoded to indicate that there is a single-bit error at data bit 41). If no error is detected the syndrome bits will all be zeroes. In Detect Mode the contents of the Data Input Latch are driven directly to the Inputs of the Data Output Latch without corrections.

Correct Mode

In this mode, the EDC functions the same as in Detect Mode except that the correction network is allowed to correct (complement) any single-bit error of the Data Input Latch before putting it onto the inputs of the Data Output Latch. If multiple errors are detected, the output of the correction network is unspecified, if the single bit error is a check bit there is no automatic correction. Check bit correction can be done by placing the device in generate mode to produce a correct check bit sequence for the data in the Data Input Latch.

To perform the correction step, all four slices require access to the syndrome bits which are generated externally of the devices. This access is provided by reading the syndrome bits in through the CB inputs where they are selected as inputs to the bit-in-error decoded by the multiplexer (see block diagram). The device connections for this operation are shown in Figure 8. When in Correct Mode the SC outputs must be enabled so that the syndrome bits are available at the CB inputs.

Pass Thru Mode

In this mode, the unmodified contents of the Data Input Latch are placed on the inputs of the Data Output Latch, and the contents of the Check Bit Input Latch are passed through the external XOR network and appear inverted at the XOR gate outputs labeled CX to C32 — see Figure 8.

Diagnostic Latch

The Diagnostic Latch is used for both diagnostic and internal control of the EDC. Table 13 provides bit definitions and shows the 64-bit loading format.

Diagnostic Generate/Detect/Correct

These are special diagnostic modes selected by DIAG $MODE_{0-1}$ where either normal check bit inputs or outputs are substituted for by check bits from the Diagnostic Latch — see Table 2 for details.

Table 13. Diagnostic Latch Loading

DATA BIT	INTERNAL FUNCTION
0	Diagnostic Check Bit X
1	Diagnostic Check Bit 0
2	Diagnostic Check Bit 1
3	Diagnostic Check Bit 2
4	Diagnostic Check Bit 4
5	Diagnostic Check Bit 8
6, 7	Don't Care
8	Slice 0/1 — CODE ID 0
9	Slice 0/1 — CODE ID 1
10	Slice 0/1 — CODE ID 2
11	Slice 0/1 — DIAG MODE 0
12	Slice 0/1 — DIAG MODE 1
13	Slice 0/1 — CORRECT
14	Slice 0/1 — PASS THRU
15	Don't Care
16 – 23	Don't Care
24	Slice 2/3 — CODE ID 0
25	Slice 2/3 — CODE ID 1
26	Slice 2/3 — CODE ID 2
27	Slice 2/3 — DIAG MODE 0
28	Slice 2/3 — DIAG MODE 1
29	Slice 2/3 — CORRECT
30	Slice 2/3 — PASS THRU
31	Don't Care
32 – 37	Don't Care
38	Diagnostic Check Bit 16
39	Don't Care
40	Slice 4/5 — CODE ID 0
41	Slice 4/5 — CODE ID 1
42	Slice 4/5 — CODE ID 2
43	Slice 4/5 — DIAG MODE 0
44	Slice 4/5 — DIAG MODE 1
45	Slice 4/5 — CORRECT
46	Slice 4/5 — PASS THRU
47	Don't Care
48 – 54	Don't Care
55	Diagnostic Check Bit 32
56	Slice 6/7 — CODE ID 0
57	Slice 6/7 — CODE ID 1
58	Slice 6/7 — CODE ID 2
59	Slice 6/7 — DIAG MODE 0
60	Slice 6/7 — DIAG MODE 1
61	Slice 6/7 — CORRECT
62	Slice 6/7 — PASS THRU
63	Don't Care

Internal Control Mode

This mode is selected by CODE ID_{0-2}, input 001 (ID_2, ID_2, ID_0).

When in Internal Control Mode the EDC takes the CODE ID_{0-2}, DIAG $MODE_{0-1}$, CORRECT and PASS THRU signals from the internal Diagnostic Latch rather than from the external control lines — see Table 13 for latch loading.

AC Calculations

Table 14 shows key AC parameters for the 64-bit configuration.

Functional Equations

The following equations and tables describe in detail how the output values of the Signetics 2960 are determined as a function of input values and internal states of the chip. Before examining the tables, the following symbol definitions should be carefully studied.

Table 14. Key AC Calculations for the 64-Bit Configuration

64-BIT PROPAGATION DELAY		COMPONENT DELAYS FROM 2960 AC SPECIFICATIONS, TABLE C (PLUS MSI)
From	To	
DATA	Check Bits Out	(DATA to SC) + (XOR Delay)
DATA In	Corrected DATA Out	(DATA to SC) + (XOR Delay) + (Buffer Delay) + (CB to DATA, CODE ID 1xx)
DATA	Syndromes	(DATA to SC) + (XOR Delay)
DATA	$\overline{\text{ERROR}}$ for 64 Bits	(DATA to SC) + (XOR Delay) + (NOR Delay)
DATA	$\overline{\text{MULT ERROR}}$ for 64 Bits	(DATA to SC) + (XOR Delay) + (Buffer Delay) + (CB to $\overline{\text{MULT ERROR}}$, CODE ID 1xx)
DATA	DOUBLE ERROR for 64 Bits	(DATA to SC) + (XOR Delay) + (XOR/NOR Delay)

Table 15. TOME (Three or More Errors)*

			S0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
			**S6	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
			S5	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
			S4	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
S1	S2	S3																	
0	0	0		0	0	0	1	0	1	1	1	0	1	1	1	0	0	0	0
0	0	1		0	1	1	1	0	0	0	0	0	0	0	0	1	1	1	1
0	1	0		0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
0	1	1		1	1	1	1	0	0	0	0	0	0	0	0	1	1	1	1
1	0	0		0	1	1	1	0	0	0	0	0	0	0	0	1	1	1	1
1	0	1		0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1
1	1	0		1	1	1	1	0	0	0	0	0	0	0	0	1	1	1	1
1	1	1		0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1

*S6, S5, ... S0 are internal syndromes except in Modes 010, 100, 101, 110, 111, (CODE ID_2, ID_1, ID_0). In these modes the syndromes are input over the Check-Bit lines S6 – C6, S5 – C5, S1 – C1, S0 – C0.

**The S6 internal syndrome is always forced to 0 in CODE ID 000.

SC Outputs

Tables 16 through 20 show how outputs SC_{0-6} are generated in each control mode for various CODE IDs (internal control mode not applicable).

DEFINITIONS

$D_i \leftarrow$ ($DATA_i$ if LE IN is HIGH or the output of bit i of the Data input Latch if LE IN is LOW)
$C_i \leftarrow$ (CB_i if LE IN is HIGH or the output of bit i of the Check Bit Latch if LE IN is LOW)
$DL_i \leftarrow$ Output of the bit i of the Diagnostic Latch
$S_i \leftarrow$ Internally generated syndromes (same as outputs of SC_i if outputs enabled)
PA $\leftarrow$ D0 ⊕ D1 ⊕ D2 ⊕ D4 ⊕ D6 ⊕ D8 ⊕ D10 ⊕ D12
PB $\leftarrow$ D0 ⊕ D1 ⊕ D2 ⊕ D3 ⊕ D4 ⊕ D5 ⊕ D6 ⊕ D7
PC $\leftarrow$ D8 ⊕ D9 ⊕ D10 ⊕ D11 ⊕ D12 ⊕ D13 ⊕ D14 ⊕ D15
PD $\leftarrow$ D0 ⊕ D3 ⊕ D4 ⊕ D7 ⊕ D9 ⊕ D10 ⊕ D13 ⊕ D15
PE $\leftarrow$ D0 ⊕ D1 ⊕ D5 ⊕ D6 ⊕ D7 ⊕ D11 ⊕ D12 ⊕ D13
PF $\leftarrow$ D2 ⊕ D3 ⊕ D4 ⊕ D5 ⊕ D6 ⊕ D7 ⊕ D14 ⊕ D15
$PG_1 \leftarrow$ D0 ⊕ D4 ⊕ D6 ⊕ D7
$PG_2 \leftarrow$ D1 ⊕ D2 ⊕ D3 ⊕ D5
$PG_3 \leftarrow$ D8 ⊕ D9 ⊕ D11 ⊕ D14
$PG_4 \leftarrow$ D10 ⊕ D12 ⊕ D13 ⊕ D15

ERROR SIGNALS

$$\text{ERROR} \leftarrow \overline{(\overline{S6} \cdot (ID_1 + ID_2))} \cdot \overline{S5} \cdot \overline{S4} \cdot \overline{S3} \cdot \overline{S2} \cdot \overline{S1} \cdot \overline{S0} + \text{GENERATE} + \text{INITIALIZE} + \text{PASSTHRU}$$

$$\overline{\text{MULT ERROR}} \text{ (16 and 32-Bit Modes)} \leftarrow \overline{((\overline{S6} \cdot ID_1) \cdot S5 \cdot S4 \cdot S3 \cdot S2 \cdot S1 \cdot S0)\ (\overline{\text{ERROR}})} + \text{TOME} + \text{GENERATE} + \text{PASSTHRU} + \text{INITIALIZE}$$

$$\overline{\text{MULT ERROR}} \text{ (64-Bit Modes)} \leftarrow \text{TOME} + \text{GENERATE} + \text{PASSTHRU} + \text{INITIALIZE}$$

Table 16. Syndrome/Check Bit Generation in GENERATE Mode

GENERATE MODE (CHECK BITS)	CODE ID_{2-0}						
	000	010	011	100	101	110	111
SC_0*	$PG_2 \oplus PG_3$	$PG_1 \oplus PG_3$	$PG_2 \oplus PG_4$	$PG_2 \oplus PG_3$	$PG_2 \oplus PG_3$	$PG_1 \oplus PG_4$	$PG_1 \oplus PG_4$
SC_1*	PA	PA	PA	PA	PA	PA	PA
SC_2*	$\overline{PD}$	$\overline{PD}$	PD	$\overline{PD}$	PD	PD	PD
SC_3*	$\overline{PE}$	$\overline{PE}$	PE	$\overline{PE}$	PE	PE	PE
SC_4*	PF	PF	PF	PF	PF	PF	PF
SC_5*	PC	PC	PC	PC	PC	PC	PC
SC_6*	1	PB	PC	PB	PB	PB	PB

Table 17. Syndrome/Check Bit Generation in Detect/Correct Modes

DETECT AND CORRECT MODES (SYNDROMES)	CODE ID_{2-0}						
	000	010	011*	100	101	110	111
SC_0*	$PG_2 \oplus PG_3 \oplus C0$	$PG_1 \oplus PG_3 \oplus C0$	$PG_2 \oplus PG_4 \oplus CB_0$	$PG_2 \oplus PG_3 \oplus C0$	$PG_2 \oplus PG_3$	$PG_1 \oplus PG_4$	$PG_1 \oplus PG_4$
SC_1*	$PA \oplus C1$	$PA \oplus C1$	$PA \oplus CB_1$	$PA \oplus C1$	PA	PA	PA
SC_2*	$\overline{PD} \oplus C2$	$\overline{PD} \oplus C2$	$PD \oplus CB_2$	$\overline{PD} \oplus C2$	PD	PD	PD
SC_3*	$\overline{PE} \oplus C3$	$\overline{PE} \oplus C3$	$PE \oplus CB_3$	$\overline{PE} \oplus C3$	PE	PE	PE
SC_4*	$PF \oplus C4$	$PF \oplus C4$	$PF \oplus CB_4$	$PF \oplus C4$	PF	PF	PF
SC_5*	$PC \oplus C5$	$PC \oplus C5$	$PC \oplus CB_5$	$PC \oplus C5$	PC	PC	PC
SC_6*	1	$PB \oplus C6$	$PC \oplus CB_6$	PB	PB	$PB \oplus C6$	$PB \oplus C6$

*In CODE ID_{2-0} 011 the Check-Bit Latch is forced transparent, the Data Latch operates normally.

Table 18. Syndrome/Check Bit Generation in Diagnostic Read Mode

DIAGNOSTIC READ MODE	CODE ID_{2-0}						
	000	010	011*	100	101	110	111
SC_0*	$PG_2 \oplus PG_3 \oplus DL_0$	$PG_1 \oplus PG_3 \oplus DL_0$	$PG_2 \oplus PG_4 \oplus CB_0$	$PG_2 \oplus PG_3 \oplus DL_0$	$PG_2 \oplus PG_3$	$PG_1 \oplus PG_4$	$PG_1 \oplus PG_4$
SC_1*	$PA \oplus DL_1$	$PA \oplus DL_1$	$PA \oplus CB_1$	$PA \oplus DL_1$	PA	PA	PA
SC_2*	$\overline{PD} \oplus DL_2$	$\overline{PD} \oplus DL_2$	$PD \oplus CB_2$	$\overline{PD} \oplus DL_2$	PD	PD	PD
SC_3*	$\overline{PE} \oplus DL_3$	$\overline{PE} \oplus DL_3$	$PE \oplus CB_3$	$\overline{PE} \oplus DL_3$	PE	PE	PE
SC_4*	$PF \oplus DL_4$	$PF \oplus DL_4$	$PF \oplus CB_4$	$PF \oplus DL_4$	PF	PF	PF
SC_5*	$PC \oplus DL_5$	$PC \oplus DL_5$	$PC \oplus CB_5$	$PC \oplus DL_5$	PC	PC	PC
SC_6*	1	$PB \oplus DL_6$	$PC \oplus CB_6$	PB	PB	$PB \oplus DL_6$	$PB \oplus DL_6$

*In CODE ID_{2-0} 011 the Check-Bit Latch is forced transparent, the Data Latch operates normally.

Table 19. Syndrome/Check Bit Generation in Diagnostic Write Mode

DIAGNOSTIC WRITE MODE	CODE ID_{2-0}						
	000	010	011*	100	101	110	111
SC_0 ←	DL_0	DL_0	CB_0	DL_0	1	1	1
SC_1 ←	DL_1	DL_1	CB_1	DL_1	1	1	1
SC_2 ←	DL_2	DL_2	CB_2	DL_2	1	1	1
SC_3 ←	DL_3	DL_3	CB_3	DL_3	1	1	1
SC_4 ←	DL_4	DL_4	CB_4	DL_4	1	1	1
SC_5 ←	DL_5	DL_5	CB_5	DL_5	1	1	1
SC_6 ←	1	DL_6	CB_6	1	1	DL_6	DL_7

*In CODE ID_{2-0} 011 the Check-Bit Latch is forced transparent; the Data Latch operates normally.

Table 20. Syndrome/Check Bit Generation in PASS THRU Mode

PASS THRU MODE	CODE ID_{2-0}						
	000	010	011*	100	101	110	111
SC_0 ←	C0	C0	CB_0	C0	1	1	1
SC_1 ←	C1	C1	CB_1	C1	1	1	1
SC_2 ←	C2	C2	CB_2	C2	1	1	1
SC_3 ←	C3	C3	CB_3	C3	1	1	1
SC_4 ←	C4	C4	CB_4	C4	1	1	1
SC_5 ←	C5	C5	CB_5	C5	1	1	1
SC_6 ←	1	C6	CB_6	1	1	C6	C6

*In CODE ID_{2-0} 011 the Check-Bit Latch is forced transparent; the Data Latch operates normally.

Table 21. CODE ID_{2-0} = 000*

		S5	0	0	0	0	1	1	1	1
		S4	0	0	1	1	0	0	1	1
		S3	0	1	0	1	0	1	0	1
S2	S1									
0	0		—	—	—	5	—	11	14	—
0	1		—	1	2	6	8	12	—	—
1	0		—	—	3	7	9	13	15	—
1	1		—	0	4	—	10	—	—	—

*Unlisted S combinations are no correction.

Table 22. CODE ID_{2-0} = 010*

		CB_6	0	0	0	0	1	1	1	1
		CB_5	1	1	1	1	0	0	0	0
		CB_4	0	0	1	1	0	0	1	1
		CB_3	0	1	0	1	0	1	0	1
CB_2	CB_1									
0	0		—	11	14	—	—	—	—	5
0	1		8	12	—	—	—	1	2	6
1	0		9	13	15	—	—	—	3	7
1	1		10	—	—	—	—	0	4	—

*Unlisted CB combinations are no correction.

Table 23. CODE ID_{2-0} = 011*

		S6	0	0	0	0	1	1	1	1
		S5	0	0	0	0	1	1	1	1
		S4	0	0	1	1	0	0	1	1
		S3	0	1	0	1	0	1	0	1
S2	S1									
0	0		—	—	—	5	—	11	14	—
0	1		—	1	2	6	8	12	—	—
1	0		—	—	3	7	9	13	15	—
1	1		—	0	4	—	10	—	—	

*Unlisted S combinations are no correction.

Table 24. CODE ID_{2-0} = 100*

		CB_0	0	0	0	0	1	1	1	1
		CB_6	0	0	0	0	1	1	1	1
		CB_5	1	1	1	1	0	0	0	0
		CB_4	0	0	1	1	0	0	1	1
		CB_3	0	1	0	1	0	1	0	1
CB_2	CB_1									
0	0		—	11	14	—	—	—	—	5
0	1		8	12	—	—	—	1	2	6
1	0		9	13	15	—	—	—	3	7
1	1		10	—	—	—	—	0	4	—

*Unlisted CB combinations are no correction.

Data Correction

Tables 21 through 27 show which data output bits are corrected (inverted) depending upon the syndromes and the CODE ID position. Note that the syndromes that determine data correction are in some cases syndromes input externally via the CB inputs and in some cases syndromes generated internally by that EDC (S_i are the internal syndromes and are the same as the value of the SC_i output of that EDC if enabled).

The tables show the number of data bit inverted (corrected) if any for the CODE ID and syndrome combination.

Table 25. CODE ID_{2-0} = 101*

	CB_0	0	0	0	0	1	1	1	1
	CB_6	0	0	0	0	1	1	1	1
	CB_5	0	0	0	0	1	1	1	1
	CB_4	0	0	1	1	0	0	1	1
	CB_3	0	1	0	1	0	1	0	1
CB_2	CB_1								
0	0	—	—	—	5	—	11	14	—
0	1	—	1	2	6	8	12	—	—
1	0	—	—	3	7	9	13	15	—
1	1	—	0	4	—	10	—	—	—

*Unlisted CB combinations are no correction.

Table 26. CODE ID_{2-0} = 110*

	CB_0	0	0	0	0	1	1	1	1
	CB_6	1	1	1	1	0	0	0	0
	CB_5	0	0	0	0	1	1	1	1
	CB_4	0	0	1	1	0	0	1	1
	CB_3	0	1	0	1	0	1	0	1
CB_2	CB_1								
0	0	—	—	—	5	—	11	14	—
0	1	—	1	2	6	8	12	—	—
1	0	—	—	3	7	9	13	15	—
1	1	—	0	4	—	10	—	—	—

*Unlisted CB combinations are no correction.

Table 27. CODE ID_{2-0} = 111*

	CB_0	0	0	0	0	1	1	1	1
	CB_6	1	1	1	1	0	0	0	0
	CB_5	1	1	1	1	0	0	0	0
	CB_4	0	0	1	1	0	0	1	1
	CB_3	0	1	0	1	0	1	0	1
CB_2	CB_1								
0	0	—	11	14	—	—	—	—	5
0	1	8	12	—	—	—	1	2	6
1	0	9	13	15	—	—	—	3	7
1	1	10	—	—	—	—	0	4	—

*Unlisted CB combinations are no correction.

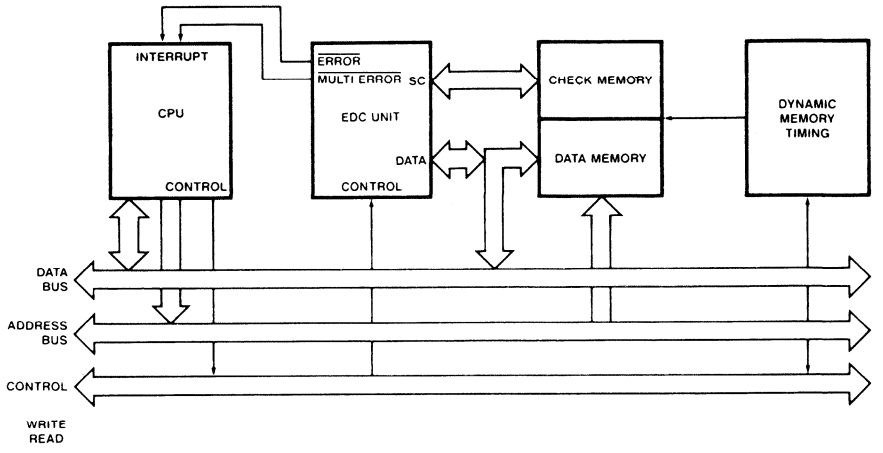


Figure 9. Check-Only Configuration

SYSTEM DESIGN CONSIDERATIONS

High Performance Parallel Operation

For maximum memory system performance the EDC should be used in the Check-Only configuration shown in Figure 9. With this configuration the memory system operates as fast with EDC as it would without.

On reads from memory, data is read out from the RAMs directly to the data bus (same as in a non-EDC system). At the same time, the data is read into the EDC to check for errors.

If an error exists the EDC's error flags are used to interrupt the CPU and/or to stretch the memory cycle. If no error is detected, no slowdown is required.

If an error is detected, the EDC generates corrected data for the processor. At the designer's option the correct data may be written back into memory; error logging and diagnostic routines may also be run under processor control.

The Check-Only configuration allows data reads to proceed as fast with EDC as without. Only if an error is detected is there any slowdown. But even if the memory system had an error every hour this would mean only one error every 3 – 4 billion memory cycles. So even with a very high error rate, EDC in a Check-Only configuration has essentially zero impact on memory system speed.

On writes to memory, check bits must be generated before the full memory word can be written into memory. The data word is frequently buffered while the check bits are generated. This makes the check bit generate time transparent to the processor.

EDC in the Data Path

The simplest configuration for EDC is to have the EDC directly in the data path as shown in Figure 10. Correct-Always Configuration). In the configuration data read from memory is always corrected prior to putting the data on the data bus. The advantages are simpler operation and no need for mid-cycle interrupts. The disadvantages is that memory system speed is slowed by the amount of time it takes for error correction on every cycle.

Usually the Correct-Always Configuration will be used with MOS microprocessors which have ample memory timing budgets. Most high performance processors will use the high performance parallel configuration shown in Figure 9.

Scrubbing Avoids Double Errors

Single-bit errors are by far the most common in a memory system and are always correctable by the EDC.

Double bit memory errors are far less frequent than single bit errors (50 to 1, or 100 to 1) and are always detected by the EDC but not corrected.

In a memory system, soft errors occur only one at a time. A double bit error in a data word occurs when a single soft error is left uncorrected and is followed by another error in the data word hours, days, or weeks after the first.

"Scrubbing" memory periodically avoids almost all double-bit errors. In the scrubbing operation, every data word in a memory is periodically checked by the EDC for single-bit errors. If one is found, it is corrected and the data word written back into memory. Errors are not allowed to pile up and so most double-bit errors are avoided.

The scrubbing operation is generally done as a background routine when the memory is not being used by the processor.

If memory is scrubbed frequently, errors are detected and corrected during processor accesses need not be immediately written back into memory. Instead the error will be corrected in memory during scrubbing. This reduces the time delay involved in a processor access of an incorrect memory word.

Correction of Double-Bit Errors

In some cases, double-bit memory errors can be corrected. This is possible when one of the two bit errors is a hard error.

When a double bit error is detected the data word should be checked to determine if one of the errors is a hard error. If so the hard error bit may be corrected by inverting it leaving only a single, correctable error. The time for this operation is negligible since it will occur infrequently.

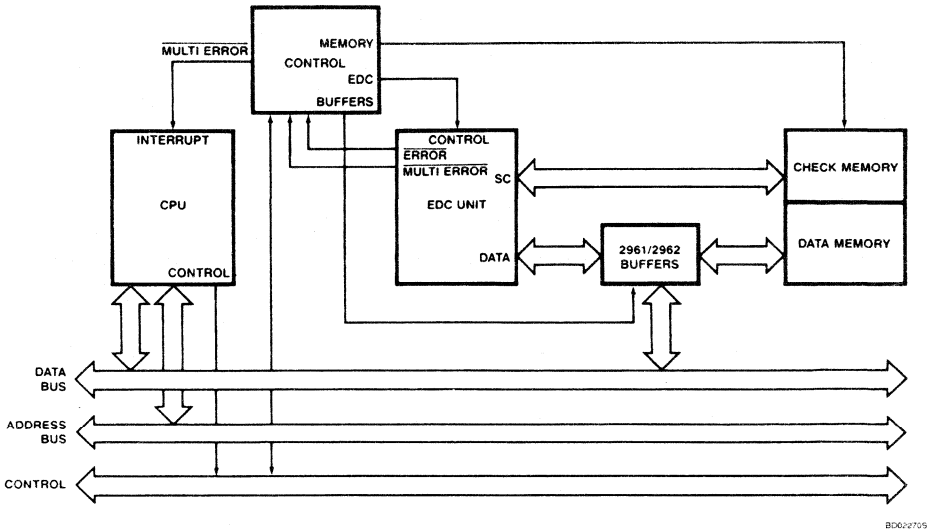


Figure 10. Correct-Always Configuration

The procedure after detection of a double error is as follows:

- Invert the data bits read from memory.
- Write the inverted data back into the same memory word.
- Re-read the memory location and XOR the newly read out value with the old. If there is no hard error then the XOR result will be 1's. If there is a hard error, it will have the same bit value regardless of what was written in. So it will show as a 0 after the XOR operation.
- Invert the hard error bit (this will "correct" it) leaving only one error in the data.
- The EDC can then correct the single bit error.
- Rewrite the correct data word into memory. This does not change the hard error but does eliminate the soft error. So the next memory access will find only a single-bit, correctable error.

An example helps to illustrate the procedure:

Example of Double Bit Error Correction When One is a Hard Error

		16 DATA BITS	6 CHECK BITS
1)	Data read from memory ($D_2$2)	1111111100000011	011010
2)	EDC detects a multiple error Syndromes:		011000
3)	Syndrome decode indicates a double bit error.		
4)	Invert the bits read from memory ($D_1$1)	0000000011111100	100101
5)	Write D_1 back to the same memory location.		
6)	Read back the memory location (D_2)	0000000011111101	100101
7)	XOR D_1 and D_2	1111111100000010	111111
8)	So the last data bit is the hard error. Use this to modify D_1	1111111100000010	011010
9)	Pass the modified D_1 through the EDC. The EDC detects a single bit correctable error and outputs corrected data.	1111111100000000	011010
10)	Write the corrected data back to memory to fix the soft error		

Error Logging and Preventative Maintenance

The effectiveness of preventative maintenance can be increased by logging information on errors detected by the EDC. This is called error logging.

The EDC provides syndromes when errors are detected. The syndromes indicate which bit is in error. In most memory systems, each individual RAM supplies only one bit of the memory word. So the syndrome and data word address specify which RAM was in error.

Typically a permanent/hard RAM failure is preceded by a period of time where the RAM displays an increasing frequency of intermittent, soft errors. Error logging statistics can be used to detect an increasing intermittent error frequency so that the RAM can be replaced before a permanent failure occurs.

Error logging also records the location of already hard failed RAMs. With EDC a hard failure will not halt system operation. EDC always can correct single bit errors even if it is a hard error. EDC can also correct double bit errors where one is hard and one soft. The ability to continue operation despite hard errors can greatly reduce the need for emergency field maintenance. The hard-failed RAMs can be instead replaced at low cost during a regularly scheduled preventative maintenance session.

Reducing Check Bit Overhead

Memory word widths need not be same as the data word width of the processor. There is a substantial reduction in check bit overhead if wider memory words are used. (See Table 28.)

This reduction in check-bit overhead lowers cost and increases the amount of data that can be packed on to each board.

The tradeoff is that when writing data pieces into memory that are narrower than the memory word width, *more steps are required.* These steps are exactly the same as those described in Byte Write in the Applications section. No penalty exists for reads from memory.

Table 28. Reducing Check Bit Overhead

MEMORY WORD		CHECK BIT OVERHEAD
#Data Bits	#Check Bits	
8	5	38%
16	6	27%
32	7	14%
64	8	11%

EDC per Board vs EDC Per System

The choice of an EDC per system or per board depends on the economics and the architecture of the system.

Certainly the cheaper approach is to have only one EDC per system and this is a viable solution if only one memory location is accessed at a time.

This solution does require that the system have both data and check bit lines - see Figure 11. This makes retrofitting a system difficult and creates complications if static or ROM memory, which do not require check bits, are mixed in with dynamic RAM.

If the system has an advanced architecture it is quite likely that it is necessary to simultaneously access memory locations on different memory boards — see Figure 12. Architectural features that require this are interleaved memory, cache memory, and DMA that is done simultaneously with processor memory accesses. EDC per board is a simpler system from a design standpoint.

The EDC is designed to work efficiently in either the per system or per board configurations.

Test Information

Incoming test procedures on this device should be carefully planned, taking into account the complexity and power levels of the part. The following notes may be useful.

1. Insure the part is adequately decoupled at the test head. Large changes in V_{CC} current as the device switches may cause erroneous function failures due to V_{CC} changes.
2. Do not leave inputs floating during any tests, as they may start to oscillate at high frequency.
3. Do not attempt to perform threshold tests at high speed. Following an input transition, ground current may change by as much as 400mA in 5 – 8ns. Inductance in the ground cable may allow the ground pin at the device to rise by 100's of millivolts momentarily.
4. Use extreme care in defining input levels for AC tests. Many inputs may be changed at once, so there will be significant noise at the device pins and they may not actually reach V_{IL} or V_{IH} until the noise has settled. Signetics recommends using V_{IL} 0.4V and V_{IH} −2.4V for AC tests.
5. To simplify failure analysis, programs should be designed to perform DC, Function, and AC tests as three distinct groups of tests.
6. To assist in testing, Signetics offers documentation on our test procedures and, in most cases, can provide Fairchild Sentry programs, under license.

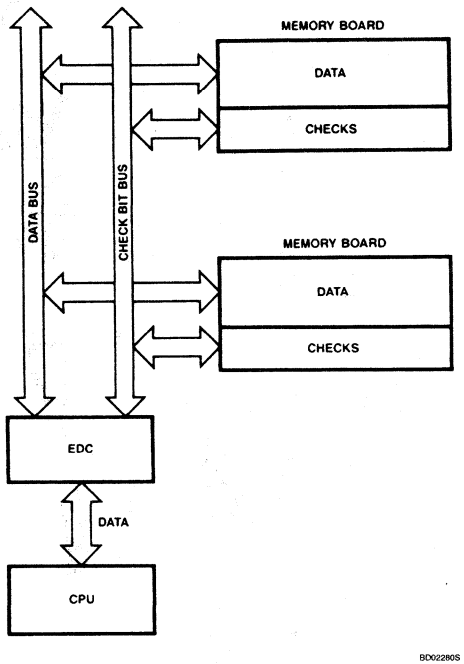


Figure 11. EDC Per System

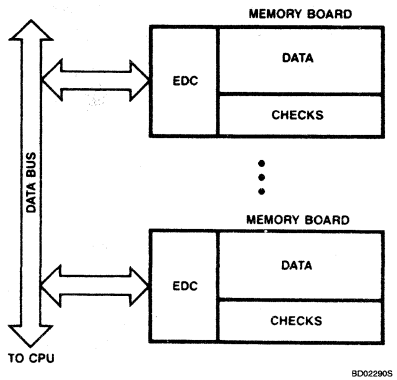


Figure 12. EDC Per Board

ABSOLUTE MAXIMUM RATINGS (Above which the useful life may be impaired)

PARAMETER	RATING	UNIT
Storage Temperature	−65 to +150	°C
Temperature (Case) Under Bias	−55 to +125	°C
Supply Voltage to Ground Potential	−0.5 to +7.0	V
DC Voltage Applied to Outputs for High Output State	−0.5 to V_{CC} max.	V
DC Input Voltage	−0.5 to +5.5	V
DC Output Current, Into Outputs	30	mA
DC Input Current	−30 to +5.0	mA

OPERATING RANGE

PART NO.	TEMPERATURE	V_{CC}
N2960N N2960I	T_A = 0 to +70°C	5V (± 5%)

DC CHARACTERISTICS V_{CC} MIN = 4.75V, V_{CC} MAX = 5.25V

PARAMETER		TEST CONDITIONS[1]			2960 Min	2960 Typ[2]	2960 Max	UNIT
V_{OH}	Output HIGH voltage	V_{CC} = MIN, V_{IN} = V_{IH} or V_{IL}	I_{OH} = −0.8mA		2.7			V
V_{OL}	Output LOW voltage	V_{CC} = MIN, V_{IN} = V_{IH} or V_{IL}	I_{OL} = 8mA				0.5	V
V_{IH}	Input HIGH voltage	Guaranteed input logical HIGH voltage for all inputs[6]			2.0			V
V_{IL}	Input LOW voltage	Guaranteed input logical LOW voltage for all inputs[6]					0.8	V
V_I	Input clamp voltage	V_{CC} = MIN, I_{IN} = −18mA					−1.5	V
I_{IL}	Input LOW current	V_{CC} = MAX, V_{IN} = 0.5V	$DATA_{0-15}$				−410	µA
			All other inputs				−360	
I_{IH}	Input HIGH current	V_{CC} = MAX, V_{IN} = 2.7V	$DATA_{0-15}$				70	µA
			All other inputs				50	
I_I	Input HIGH current	V_{CC} = MAX, V_{IN} = 5.5V					1.0	mA
I_{OZH}[4] I_{OZL}[4]	Off state (high impedance) output current	V_{CC} = MAX	$DATA_{0-15}$	V_O = 2.4V			70	µA
				V_O = 0.5V			−410	
			SC_{0-6}	V_O = 2.4V			50	
				V_O = 0.5V			−50	
I_{OS}	Output short circuit current[3]	V_{CC} = V_{CC} MAX + 0.5V, V_O = 0.5V			−25		−85	mA
I_{CC}	Power supply current[5]	V_{CC} = MAX	T_A = 25°C			300	360	mA
			T_A = 0 to +70°C					
			T_A = +70°C					

NOTES:
1. For conditions shown as MIN or MAX, use the appropriate value specified under Electrical Characteristics for the applicable device type.
2. Typical limits are at V_{CC} = 5.0V, 25°C ambient and maximum loading.
3. Not more than one output should be shorted at a time. Duration of the short circuit test should not exceed one second.
4. These are three-state outputs internally connected to TTL inputs. Input Characteristics are measured with output enables HIGH.
5. Worst case I_{CC} is at minimum temperature.
6. These input levels provide zero noise immunity and should only be tested in a static, noise-free environment.

GUARANTEED PERFORMANCE OVER COMMERCIAL TEMPERATURE RANGE OF 0 TO +70°C

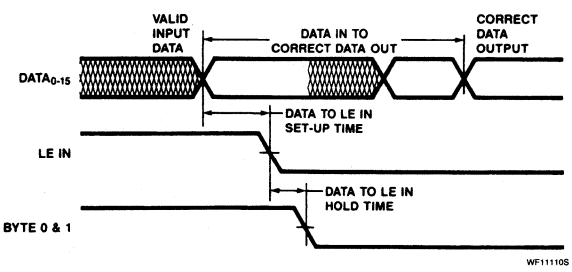
The tables that follow specify the guaranteed performance of the 2960 over the commercial operating range of 0 to +70°C, with V_{CC} from 4.75V to 5.25V. All data are in ns with inputs switching between 0V and 3V at 1V/ns and measurements made at 1.5V. All outputs have maximum DC load.

This data applies to the following part numbers: N2960N and N2960I.

COMBINATIONAL PROPAGATION DELAYS — C_L = 50pF

FROM INPUT \ TO OUTPUT	SC_{0-6}	$DATA_{0-15}$	$\overline{ERROR}$	$\overline{MULT\ ERROR}$
$DATA_{0-15}$	32	65*	32	50
CB_{0-6} (CODE ID_{2-0} 000, 011)	28	56	29	47
CB_{0-6} (CODE ID_{2-0} 010, 100, 101, 110, 111)	28	45	29	34
$\overline{GENERATE}$	35	63	36	55
CORRECT (Not internal control mode)	—	45	—	—
DIAG MODE (Not internal control mode)	50	78	59	75
PASS THRU (Not internal control mode)	36**	44	29	46
CODE ID_{2-0}	61	90	60	80
LE IN (From latched to transparent)	39	72*	39	59
LE OUT (From latched to transparent)	—	31	—	—
LE DIAG (From latched to transparent; not internal control mode)	45	78	45	65
Internal control mode: LE DIAG (from latched to transparent)	67	96	66	86
Internal control mode: $DATA_{0-15}$ (via diagnostic latch)	67	96	66	86

*Data In (or LE In) to Correct Data Out measurement requires timing as shown below.



** Device not tested in GENERATE mode.

Error Detection and Correction (EDC) Unit 2960

SET-UP AND HOLD TIMES RELATIVE TO LATCH ENABLES

FROM INPUT	TO (LATCHING UP DATA)	SET-UP TIME	HOLD TIME
$DATA_{0-15}$	LE IN	6	7
CB_{0-6}	LE IN	5	6
$DATA_{0-15}$	LE OUT	44	5
CB_{0-6} (CODE ID 000, 011)	LE OUT	35	0
CB_{0-6} (CODE ID 010, 100, 101, 110, 111)	LE OUT	27	0
$\overline{GENERATE}$	LE OUT	42	0
CORRECT	LE OUT	26	1
DIAG MODE	LE OUT	69	0
PASS THRU	LE OUT	26	0
CODE ID_{2-0}	LE OUT	81	0
LE IN	LE OUT	51	5
$DATA_{0-15}$	LE DIAG	6	8

MINIMUM PULSE WIDTHS

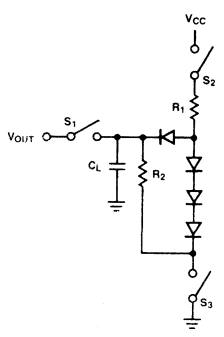
LE IN, LE OUT, LE DIAG	15

OUTPUT ENABLE/DISABLE TIMES Output disable tests performed with C_L = 5pF and measured to 0.5V change of output voltage level.

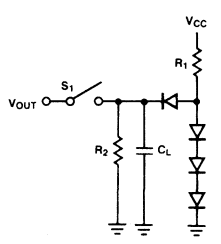
INPUT	OUTPUT	ENABLE	DISABLE
$\overline{OE}$ BYTE 0, $\overline{OE}$ BYTE 1	$DATA_{0-15}$	30	30
$\overline{OE}$ SC	SC_{0-6}	30	30

TEST LOADING

Three-State Outputs:



Normal Outputs:



NOTES:
1. C_L = 50pF includes scope probe, wiring and stray capacitances without devices in test fixture.
2. S_1, S_2, S_3 are closed during function test and all AC tests, except output enable tests.
3. S_1 and S_3 are closed while S_2 is open for t_{PZH} test. S_1 and S_2 are closed while S_3 is open for t_{PZL} test.
4. R_2 = 1K for three-state output. R_2 is determined by the I_{OH} at V_{OH} = 2.4V for non-three-state output.
5. R_1 is determined by I_{OL} (MIL) with V_{CC} = 5.0V minus the current to ground through R_2.
6. C_L = 5.0pF for output disable tests.

TEST OUTPUT LOADS

PIN #	PIN LABEL	TEST CIRCUIT	R_1	R_2
—	$D_0 - D_{15}$	Fig. 11	430Ω	1kΩ
24 – 30	$SC_0 - SC_6$	Fig. 11	430Ω	1kΩ
32	$\overline{ERROR}$	Fig. 12	470Ω	3kΩ
33	$\overline{MULTERROR}$	Fig. 12	470Ω	3kΩ

PACKAGE DATA

TYPE: Plastic and Ceramic
Configuration: DIP
Width: C = 0.6″ P = 0.55″
Length: 2.4″
Pin Centers: 2.54 BSC

ORDERING INFORMATION

Commercial:
N2960N (Plastic)
N2960I (Ceramic)

Error Detection and Correction (EDC) Unit 2960

APPLICATIONS

Byte Write

Byte operations are increasingly common for 16 and 32-bit processors. These complicate memory operations because check bits are generated for a complete 16 or 32 or 64-bit memory word, not for a single byte.

To write a byte into memory with EDC requires the following steps — See Figures 13 and 14.

- Latch the byte into the bus buffers
- Read the complete word from memory
- Correct the complete data word if necessary
- Insert the byte to be written into the data word
- Generate new check bits for the entire data word
- Store the data word back into memory.

(In fact these steps must be taken for any piece of data being written into memory that is not as wide as a full memory word).

The EDC is designed with the intent of keeping byte operations simple in error detection/correction systems. The EDC has separate output enables for each byte in the Data Output Latch. As shown in Figures 13 and 14, this allows the data word to be read from memory, the new byte to be inserted among the old, and new check bits to be generated using less time and less hardware than if separate byte enables were not available.

Diagnostics

EDC is used to boost the reliability of the overall system. It is necessary to also be able to check the operation of the EDC itself. For this reason the EDC has an internal control mode, a diagnostic latch, and two diagnostic modes.

To check that the EDC is functioning properly, the processor can put the EDC under software control by setting CODE ID_{2-0} to 001. This puts the EDC into Internal Control Mode. In Internal Control Mode the EDC is controlled by the contents of the Diagnostic Latch, which is loaded from the DATA inputs under processor control.

DATA BITS	CHECK BITS REQUIRED	
	Single Error Corrections Only	Single Error Correct & Double Error Detect
8	4	5
16	5	6
32	6	7
64	7	8

The EDC is set into CORRECT Mode. The processor loads in a known set of check bits into the Diagnostic Latch, a known set of data bits into the Data in Latch, and forces data errors. The output of the EDC (syndromes, error flags, corrected data) is then compared against the expected responses. By exercising the EDC with a string of data/check combinations and comparing the output against the expected responses, the EDC can be fully checked out.

Eight Bit Data Word

Eight bit MOS microprocessors can use EDC too. Only five check bits are required. The EDC configuration for eight bits is shown in Figure 15. It operates as does the normal 16-bit configuration with the upper byte fixed at 0.

Other Word Widths

EDC on data words other than 8, 16, 32, of 64 bits can be accomplished with the 2960. In most cases the extra data bits can be forced to a constant and EDC will proceed as normal. For example a 24-bit data word is shown in Figure 16.

Single Error Correction Only

The EDC normally corrects all single bit errors and detects all double bit and some triple bit errors. To save one check bit per word the ability to detect double bit errors can be sacrificed — single errors are still detected and corrected.

Figure 17 shows single error correction only configurations for 8, 16, 32, and 64-bit data words respectively.

Check Bit Correction

The EDC detects single bit errors whether the error is a data bit or a check bit. Data bit errors are automatically corrected by the EDC. To generate corrected check bits once a single check bit error is detected, the EDC need only be switched to GENERATE mode (data in the DATA INPUT LATCH is valid).

The syndromes generated by the EDC may be decoded to determine whether the single bit error is a check bit.

In many memory systems, a check bit error will be ignored on the memory read and corrected during a periodic "scrubbing" of memory — see System Design Considerations).

Multiple Errors

The bit-in-error decode logic uses syndrome bits SO through S32 to correct errors, SX is only used in developing the multiple error signal. This means that some multiple errors will cause a data bit to be inverted.

For example, in the 16-bit mode if data bits 8 and 13 are in error the syndrome 111100 (SX, S0, S1, S2, S4 S8) is produced. This is flagged a double error by the error detection logic, but the decoded bit-in-error only receives syndrome 11100 (S0, S1, S2, S4, S8) which it decodes as a single error in data bit 0 and inverts that bit. If it is desired to inhibit this inversion, the multiple error output may be connected to the correct input as in Figure 18. This will inhibit correction when a multiple error occurs. Extra time delay may be introduced in the data to correct data path when this is done.

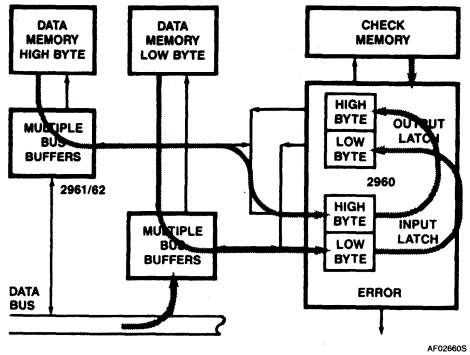


Figure 13. Byte Write, Phase 1: Read Out The Old Word And Correct

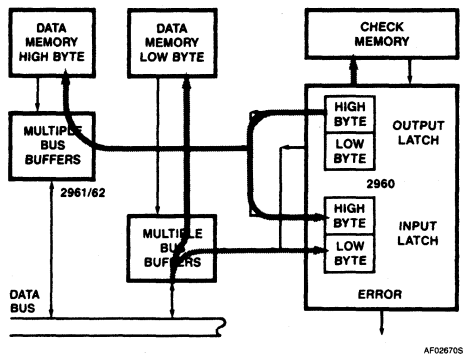


Figure 14. Byte Write, Phase 2: Insert The New Byte Generate Checks And Write Into Memory

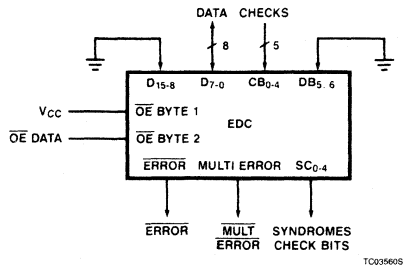


Figure 15. 8-Bit Configuration

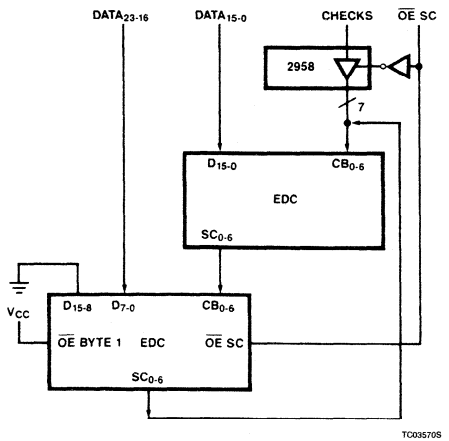
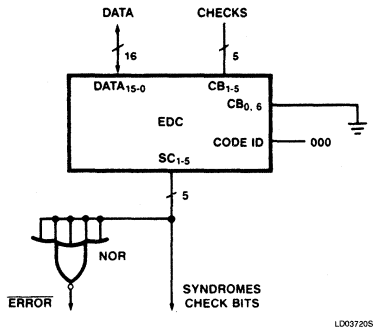
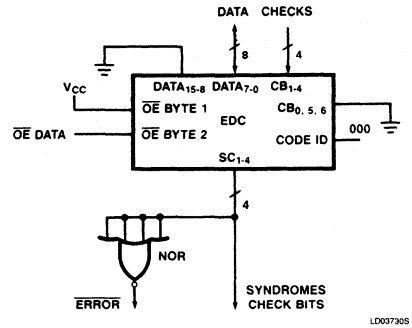


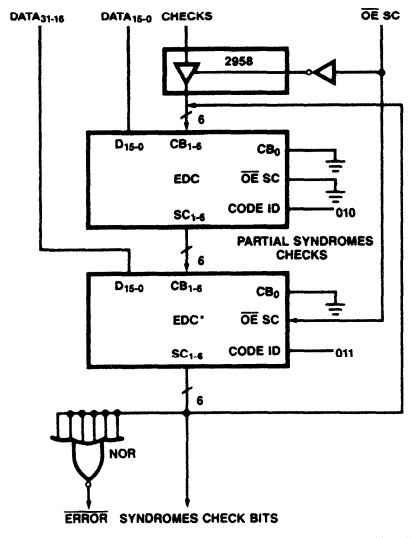
Figure 16. 24-Bit Configuration



a. 8-Bit Data Word



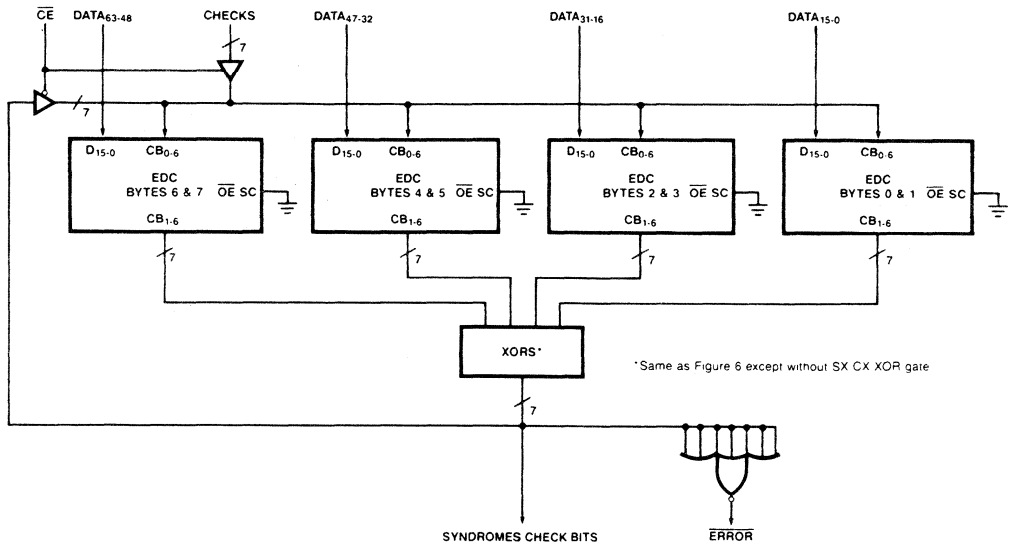
b. 16-Bit Data Word



c. 32-Bit Data Word

NOTE:
*The code ID combination for this slice forces the check bit latch transparent.

Figure 17. Single Error "Correction Only" Configurations



d. 64-Bit Data Word

NOTES:
1. In Pass Thru Mode the contents of the check latch appear on the XOR outputs inverted.
2. In Diagnostic GENERATE Mode the contents of the diagnostic latch appear on the XOR outputs inverted.

Figure 17. (Continued)

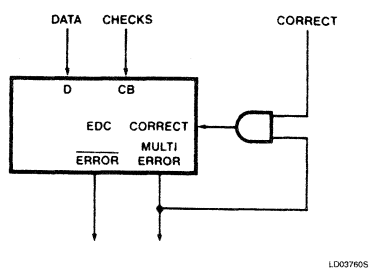


Figure 18. Inhibition Of Data Modification

Signetics

2964B
Dynamic Memory Controller

Product Specification

Logic Products

FEATURES

- **Operating Options — controls 16K or 64K DRAMs**
- **8-Bit Refresh Counter — refresh address generation, clear input, and selectable terminal count (128 or 256) output**
- **Row Address Decoder — four Active *R*ow *A*ddress *S*elect (RAS) outputs during refresh**
- **On-Chip Latches — dual 8-bit address latches and RAS decoder latches**
- **User-Selectable Refresh Modes — burst, distributed or transparent**
- **3-port, 8-bit address multiplexer with Schottky speed**
- **Non-inverting address for RAS and CAS signal paths**

PRODUCT DESCRIPTION

The Signetics 2964B **D**ynamic **M**emory **C**ontroller (DMC) provides address multiplexing, refresh address generation, and RAS/CAS control for dynamic RAMs of any data width. The eight-bit address path is designed for 64K DRAMs but can be used equally well with 16K DRAMs. Sixteen address input latches and two row address select latches (for higher order addresses) allow the DMC to control up to 256K words of memory (with 64K DRAMs) by using the internal RAS decoder to select from one-of-four banks of DRAMs.

FUNCTIONAL OPERATION

The Signetics 2964B Dynamic Memory Controller (Figure 1) replaces a dozen MSI devices by grouping several unique functions. Two 8-bit latches capture and hold the memory address. These latches and a clearable, 8-bit refresh counter feed into an 8-bit, 3-input, Schottky speed MUX, for output to the DRAM address lines.

The 2964B also includes a special RAS decoder and CAS buffer. Placing these functions on the same chip minimizes the time skew between output functions which would otherwise be separate MSI chips, and therefore, allows a faster memory cycle time by the amount of skew eliminated.

The RAS Decoder allows upper addresses to select one-of-four banks of DRAM by determining which bank receives an RAS input. During refresh (RFSH = LOW), the decoder mode is changed to four-of-four and all banks of

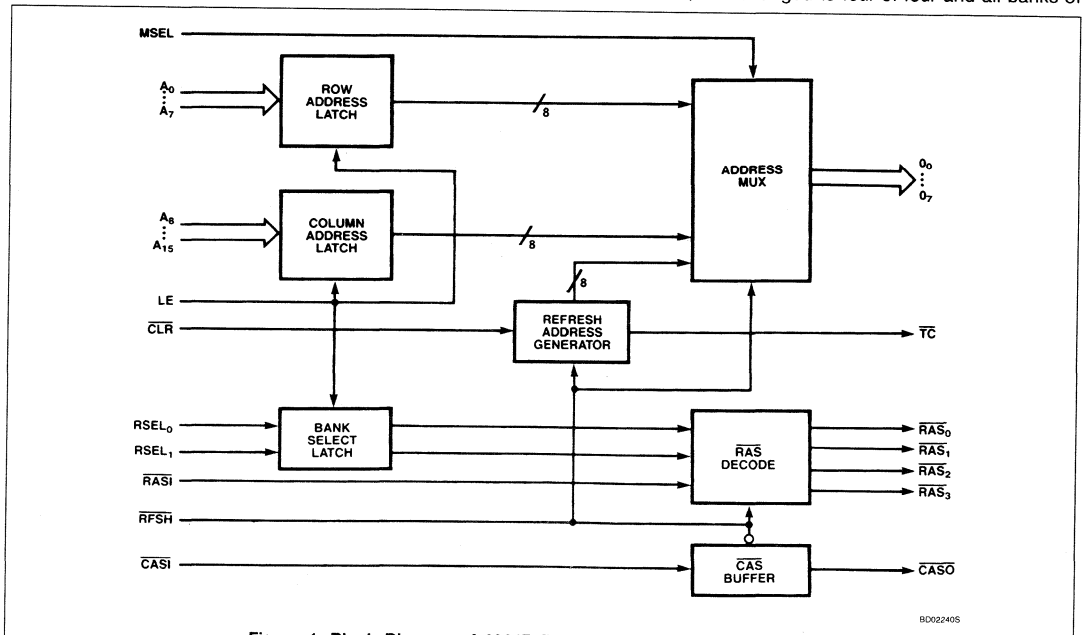
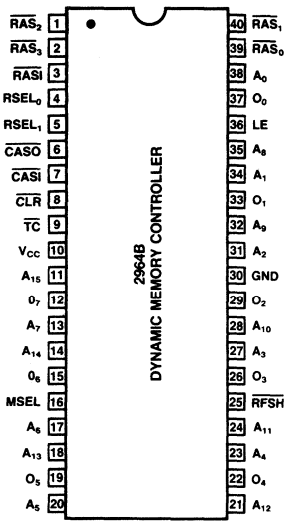


Figure 1. Block Diagram of 2964B Dynamic Memory Controller (DMC)

2964B PACKAGE/PIN DESIGNATIONS



PIN NO.	IDENTIFIER	FUNCTION
1, 2, 39 and 40	$\overline{RAS}_2$, $\overline{RAS}_3$ $\overline{RAS}_0$, $\overline{RAS}_1$	Row Address Strobe outputs ($\overline{RAS}_i$). Each provides a Row Address Strobe for one of the four banks of memory. Each will go active LOW only when selected by $RSEL_0$ and $RSEL_1$ and only when $\overline{RASI}$ goes active LOW. All $\overline{RAS}_{0\text{-}3}$ outputs go active low in response to $\overline{RASI}$ when $\overline{RFSH}$ goes LOW.
3	$\overline{RASI}$	The Row Address Strobe input. During normal memory cycles, the selected $\overline{RAS}$ Decoder output, $\overline{RAS}_0$, $\overline{RAS}_1$, $\overline{RAS}_2$ or $\overline{RAS}_3$, will go active LOW in response to an active LOW input at $\overline{RASI}$. During refresh (RFSH = LOW), all $\overline{RAS}$ outputs go LOW in response to $\overline{RASI}$ = LOW.
4 and 5	$RSEL_0$ and $RSEL_1$	The $\overline{RAS}$ decoder Select inputs. Data (latched) at these inputs (normally higher-order addresses) is decoded by the $\overline{RAS}$ Decoder to "$\overline{RAS}$ Select" one of four banks of memory with $\overline{RAS}_0$, $\overline{RAS}_1$, $\overline{RAS}_2$ or $\overline{RAS}_3$.
6	$\overline{CASO}$	The Column Address Strobe output. The active LOW $\overline{CASO}$ output strobes the Column Address into the dynamic RAM. $\overline{CASO}$ is inhibited during refresh ($\overline{RFSH}$ = LOW).
7	$\overline{CASI}$	The Column Address Strobe. An active LOW input at $\overline{CASI}$ will result in an active LOW output at $\overline{CASO}$, unless a refresh cycle is in progress ($\overline{RFSH}$ = LOW).
8	$\overline{CLR}$	The refresh counter Clear input. An active LOW input at $\overline{CLR}$ resets the refresh counter to all LOW (refresh address output to all HIGH).
9	$\overline{TC}$	The Terminal Count output. A LOW output at $\overline{TC}$ indicates that the refresh counter has been sequenced through either 128 or 256 refresh addresses depending on A_{15}. The $\overline{TC}$ output remains active LOW until the refresh counter is advanced by the rising edge of $\overline{RASI}$ or $\overline{RFSH}$.
10	V_{CC}	+5V power supply
11, 14, 18, 21, 24, 28, 32 and 35	$A_{15} - A_8$	The high-order Address inputs are used to latch eight Column Address inputs for the DRAM. These inputs drive the outputs $O_0 - O_7$ when MSEL is LOW — see next paragraph. A_{15} is a dual input. With normal TTL level inputs. A_{15} acts as an address input for 64K DRAMs. If A_{15} is pulled up to +12V through a 1K resistor, the terminal count output $\overline{TC}$ will go LOW every 128 counts (for 16K DRAMs) instead of every 256 counts.
12, 15, 19, 22, 26, 29, 33 and 37	$O_7 - O_0$	The DRAM address outputs. The eight-bit width is designed for DRAMs up to 64K.
13, 17, 20, 23, 27, 31, 34 and 38	$A_7 - A_0$	The low-order Address inputs are used to latch eight Row Address inputs for the DRAM. These inputs drive the outputs $O_0 - O_7$ when MSEL is HIGH.
16	MSEL	The Multiplexer-SELect input determines whether low-order or high-order address inputs appear at the multiplexer outputs $O_0 - O_7$. When MSEL is HIGH, the low-order address latches ($A_0 - A_7$) are connected to the outputs. When MSEL is LOW, the high-order address latches are connected to the outputs.

2964B PACKAGE/PIN DESIGNATIONS (Continued)

PIN NO.	IDENTIFIER	FUNCTION
25	$\overline{RFSH}$	The Refresh control input. When active LOW, the $\overline{RFSH}$ input switches the address output multiplexer to output the inverted contents of the 8-bit refresh counter. $\overline{RFSH}$ LOW also inhibits the $\overline{CAS}$ buffer and changes the mode of the $\overline{RAS}$ decoder from one-of-four to four-of-four so that all four $\overline{RAS}$ decoder outputs, $\overline{RAS_0}$, $\overline{RAS_1}$, $\overline{RAS_2}$ and $\overline{RAS_3}$, go LOW in response to a LOW input at $\overline{RASI}$. This action refreshes one row address in each of the four $\overline{RAS}$ decoded memory banks. The refresh counter is advanced at the end of each cycle by the LOW-to-HIGH transition of $\overline{RFSH}$ or $\overline{RASI}$ (whichever occurs first). In burst mode refresh. $\overline{RFSH}$ may be held LOW and refresh accompanied by toggling $\overline{RASI}$.
30	GND	Ground.
36	LE	The address latch enable input. An active HIGH input at LE causes the two 8-bit address latches and the 2-bit $\overline{RAS}$ Select input latch to go transparent, accepting new input data. A LOW input on LE latches the input data which meets set-up and hold time requirements.

memory receive an RAS input for refresh in response to an RASI active LOW input. CAS is inhibited during refresh.

Burst mode refresh is accomplished by holding RFSH low and toggling RASI.

A_{15} is a dual function input which controls the refresh counter's range. For 64K DRAMs, it is an address input. For 16K DRAMs, it can be pulled to +12V through 1K to terminate the refresh count at 128 instead of 256.

TRUTH TABLES: $\overline{RAS}$ OUTPUT FUNCTIONS

$\overline{RFSH}$	$\overline{RASI}$	$RSEL_1$	$RESEL_0$	$\overline{RAS_0}$	$\overline{RAS_1}$	$\overline{RAS_2}$	$\overline{RAS_3}$
L	H	X	X	H	H	H	H
L	L	X	X	L	L	L	L
H	H	X	X	H	H	H	H
H	L	L	L	L	H	H	H
H	L	L	H	H	L	H	H
H	L	H	L	H	H	L	H
H	L	H	H	H	H	H	L

$\overline{CASO}$ FUNCTION

$\overline{RFSH}$	$\overline{CASI}$	$\overline{CASO}$
H	L	L
H	H	H
L	X	H

ADDRESS OUTPUT FUNCTIONS

MSEL	$\overline{RFSH}$	$O_0 - O_7$
H	H	$A_0 - A_7$
L	H	$A_8 - A_{15}$
X	L	Refresh Address

REFRESH ADDRESS COUNTER FUNCTION

A_{15}	CLR	$\overline{RFSH}$	$\overline{RASi}$	$\overline{TC}$	REFRESH COUNT	FUNCTION
X	L	X	X	X	FF_H	Clear counter
X	H		X	X	NC	Output refresh address no change for counter
X	H		L	X	Count – 1	Return to memory cycle mode and decrement counter
X	H	L		X	NC	Output all RASI to RAM no change for counter
X	H	L		X	Count – 1	Return RAS_i to HIGH and decrement counter
L or H	H	X	X	L	00_H	Terminal count for 256 line refresh
+12V*	H	X	X	L	00_H and 80_H	Terminal count for 128 line refresh

*Through 1KΩ resistor.

ABSOLUTE MAXIMUM RATINGS (Above which useful life may be impaired)

Storage temperature	−65 to +150°C
Temperature (ambient) under bias	−55 to +125°C
Supply voltage to ground potential	−0.5 to +7.0V
DC voltage applied to outputs for high output state	+0.5V to $+V_{CC}$ MAX
DC input voltage	−0.5 to 5.5V
DC output current, into outputs	30mA
DC input current	−30 to +5.0mA

DC ELECTRICAL CHARACTERISTICS

Commercial: T_A = 0 to + 70°C, V_{CC} = 5.0V (±5%), (Min = 4.75V), (Max = 5.25V)

DESCRIPTION		TEST CONDITIONS[1]		2964B			UNIT
				Min	Typ[2]	Max	
V_{OH}	Output HIGH voltage	V_{CC} = MIN, V_{IN} = V_{IH} or V_{IL}, I_{OH} = −1mA	$\overline{TC}$	2.5			V
			Others	3.0			V
V_{OH}	Output HIGH voltage	V_{CC} = MIN, V_{IN} = V_{IH} or V_{IL}, I_{OH} = −15mA	All outputs except $\overline{TC}$	2.0			V
V_{OL}	Output LOW voltage	V_{CC} = MIN, V_{IN} = V_{IH} or I_{IL}	All outputs except $\overline{TC}$, I_{OL} = 16mA			0.5	V
			$\overline{TC}$, I_{OL} = 8mA			0.5	V
V_{IH}	Input HIGH level	Guaranteed input logical HIGH voltage for all inputs		2.0			V
V_{IL}	Input LOW level	Guaranteed input logical LOW voltage for all inputs				0.8	V
V_I	Input clamp voltage	V_{CC} = MIN, I_{IN} = −18mA				−1.5	V
I_{IL}	Input LOW current	V_{CC} = MAX, V_{IN} = 0.4V	$\overline{RAS}$I			−3.2	mA
			$\overline{CAS}$I, MSEL, $\overline{RFSH}$			−1.6	mA
			A_0 − A_{15}, $\overline{CLR}$, $RSEL_{0,1}$, LE			−0.4	mA
I_{IH}	Input HIGH current	V_{CC} = MAX, V_{IN} = 2.7V	$\overline{RAS}$I			100	μA
			$\overline{CAS}$I, MSEL, $\overline{RFSH}$			50	μA
			A_0 − A_{15}, $\overline{CLR}$, $RSEL_{0,1}$, LE			20	μA
I_I	Input HIGH current	V_{CC} = MAX, V_{IN} = −5.5V	$\overline{RAS}$I			2.0	mA
			$\overline{CAS}$I, MSEL, $\overline{RFSH}$			1.0	mA
		V_{CC} = MAX, V_{IN} = 5.5V	A_0 − A_{15}, $\overline{CLR}$, $RSEL_{0,1}$, LE			0.1	mA
I_{SC}	Output short circuit current	V_{CC} = MAX (note 3)		−40		−100	mA
I_{CC}	Power supply current (note 4)	25°C, 5V	Com'l		122		mA
		0 to 70°C				173	mA
		70°C				165	mA
I_T	A_{15} Enable current	A_{15} connected to +12V through 1KΩ ±10%				5	mA

NOTES:

1. For conditions shown as MIN or MAX, use the appropriate value specified under DC Electrical Characteristics for the applicable device type.
2. Typical limits are at V_{CC} = 5.0V, 25°C ambient and maximum loading.
3. Not more than one output should be shorted at a time. Duration of the short circuit test should not exceed one second.
4. I_{CC} is worst case when the Address inputs are latched HIGH, the refresh counter is at terminal count (255), $\overline{RAS}$I and $\overline{CAS}$I are HIGH and all other inputs are LOW.

AC ELECTRICAL CHARACTERISTICS

Tables 1 and 2 specify performance characteristics of the Signetics 2964B over the operating range for capacitive loads of 50 and 150 picofarads, respectively. Note that the minimum specified limits for t_{PW}, t_S, and t_H are for minimum system operating requirements and that limits for t_{SKEW} and t_{PD} are guaranteed test limits for the device. All AC parameters are specified at 1.5 volts.

Table 1. Performance Characteristics for Capacitive Loading of 50 Picofarads

PARAMETER — See Figure 2.	DESCRIPTION	$T_A = +25°C$ $V_{CC} = 5.0V$	COMMERCIAL $T_A = 0°C$ to $+70°C$ $V_{CC} = 5.0V \pm 5\%$		UNIT
		Typ	Min	Max	
1 t_{PD}	A_i to O_i Delay	14		19	ns
2 t_{PHL}	$\overline{RASI}$ to $\overline{RAS}_i$ ($\overline{RFSH}$ = H)	14		20	ns
3 t_{PHL}	$\overline{RASI}$ to $\overline{RAS}_i$ ($\overline{RFSH}$ = L)	14		20	ns
4 t_{PD}	MSEL to O_i	17	9		ns
5 t_{PD}	MSEL to O_i	17		21	ns
6 t_{PHL}	$\overline{CASI}$ to $\overline{CASO}$ ($\overline{RFSH}$ = H)	12		17	ns
7 t_{PHL}	$RSEL_i$ to $\overline{RAS}_i$ (LE = H, $\overline{RASI}$ = L)	15		20	ns
8 t_{PLH}	$\overline{RFSH}$ to $\overline{TC}$ ($\overline{RASI}$ = L)	30		40	ns
9 t_{PLH}	$\overline{RASI}$ to $\overline{TC}$ ($\overline{RFSH}$ = L)	25		35	ns
10 t_{PW}	$\overline{RASI}$ = L ($\overline{RFSH}$ = L)	10	50		ns
11 t_{PW}	$\overline{RASI}$ = H ($\overline{RFSH}$ = L)	10	50		ns
12 t_{PD}	$\overline{RFSH}$ to O_i ($\overline{RASI}$ = X)	17		21	ns
13 t_{PHL}	$\overline{RFSH}$ to $\overline{RAS}_i$ ($\overline{RASI}$ = L)	19		26	ns
14 t_{PW}	$\overline{CLR}$ = L	10	30		ns
15 t_{PLH}	$\overline{RFSH}$ to $\overline{CASO}$ ($\overline{RASI}$ = L $\overline{CASI}$ = L, Note 1)	16		21	ns
16 t_{PD}	LE to O_i	25		35	ns
17 t_{PHL}	LE to $\overline{RAS}_i$	30		40	ns
18 t_{PLH}	$\overline{CLR}$ to $\overline{TC}$	35		45	ns
19 t_{PLH}	$\overline{CLR}$ to O_i ($\overline{RFSH}$ = L)	31		44	ns
20 t_S	A_i to LE Set-up time	0	5		ns
21 t_H	A_i to LE Hold time	5	12		ns
22 t_S	$RSEL_i$ to LE Set-up time	0	5		ns
23 t_H	$RSEL_i$ to LE Hold time	10	17		ns
24 t_S	$\overline{CLR}$ Recovery time	10	16		ns
25 t_{SKEW}	O_i to $\overline{RAS}_i$ ($\overline{RFSH}$ = H, Note 2)	2		5	ns
26 t_{SKEW}	O_i to $\overline{CASO}$ (Note 2)	6		8	ns
27 t_{SKEW}	O_i to $\overline{RAS}_i$ ($\overline{RFSH}$ = L, Note 3)	6		8	ns
28 t_{SKEW}	O_i to $\overline{RAS}_i$ (MSEL = Z, Note 4)	1		5	ns

AC ELECTRICAL CHARACTERISTICS (Continued)

Table 2. Performance Characteristics for Capacitive Loading of 150 Picofarads

PARAMETER — See Figure 2.	DESCRIPTION	$T_A = +25°C$ $V_{CC} = 5.0V$	COMMERCIAL $T_A = 0°C$ to $+70°C$ $V_{CC} = 5.0V \pm 5\%$		UNIT
		Typ	Min	Max	
1 t_{PD}	A_i to O_i Delay	20		25	ns
2 t_{PHL}	$\overline{RASI}$ to $\overline{RAS_i}$ ($\overline{RFSH}$ = H)	18		24	ns
3 t_{PHL}	$\overline{RASI}$ to $\overline{RAS_i}$ ($\overline{RFSH}$ = L)	18		24	ns
4 t_{PD}	MSEL to O_i	23	12		ns
5 t_{PD}	MSEL to O_i	23		27	ns
6 t_{PHL}	$\overline{CASI}$ to $\overline{CASO}$ ($\overline{RFSH}$ = H)	17		24	ns
7 t_{PHL}	$RSEL_i$ to $\overline{RAS_i}$ (LE = H, $\overline{RASI}$ = L)	19		27	ns
8 t_{PLH}	$\overline{RFSH}$ to $\overline{TC}$ ($\overline{RASI}$ = L)	34		45	ns
9 t_{PLH}	$\overline{RASI}$ to $\overline{TC}$ ($\overline{RFSH}$ = L)	32		45	ns
10 t_{PW}	$\overline{RASI}$ = L ($\overline{RFSH}$ = L)	10	50		ns
11 t_{PW}	$\overline{RASI}$ = H ($\overline{RFSH}$ = L)	10	50		ns
12 t_{PD}	$\overline{RFSH}$ to O_i ($\overline{RASI}$ = X)	21		27	ns
13 t_{PHL}	$\overline{RFSH}$ to $\overline{RAS_i}$ ($\overline{RASI}$ = L)	25		33	ns
14 t_{PW}	$\overline{CLR}$ = L	10	30		ns
15 t_{PLH}	$\overline{RFSH}$ to $\overline{CASO}$ ($\overline{RASI}$ = L $\overline{CASI}$ = L, Note 1)	21		27	ns
16 t_{PD}	LE to O_i	30		40	ns
17 t_{PHL}	LE to $\overline{RAS_i}$	34		45	ns
18 t_{PLH}	$\overline{CLR}$ to $\overline{TC}$	39		55	ns
19 t_{PLH}	$\overline{CLR}$ to O_i ($\overline{RFSH}$ = L)	38		50	ns
20 t_S	A_i to LE Set-up time	0	5		ns
21 t_H	A_i to LE Hold time	5	12		ns
22 t_S	$RSEL_i$ to LE Set-up time	0	5		ns
23 t_H	$RSEL_i$ to LE Hold time	10	17		ns
24 t_S	$\overline{CLR}$ Recovery time	10	16		ns
25 t_{SKEW}	O_i to $\overline{RAS_i}$ ($\overline{RFSH}$ = H, Note 2)	3		6	ns
26 t_{SKEW}	O_i to $\overline{CASO}$ (note 2)	6		8	ns
27 t_{SKEW}	O_i to $\overline{RAS_i}$ ($\overline{RFSH}$ = L, Note 3)	6		9	ns
28 t_{SKEW}	O_i to $\overline{RAS_i}$ ($\overline{MSEL}$ ⌝⌞, Note 4)	1		5	ns

NOTES:

1. $\overline{RFSH}$ inhibits $\overline{CASO}$ during refresh. Specification is for $\overline{CASO}$ inhibit time.
2. O_i to $\overline{RAS_i}$, ($\overline{RFSH}$ = HIGH) skew is guaranteed maximum difference between fastest $\overline{RASI}$ to $\overline{RAS_i}$ delay and slowest A_i to O_i delay within a single device. O_i to $\overline{CASO}$ skew is maximum difference between fastest $\overline{CASI}$ to $\overline{CASO}$ delay and slowest MSEL to O_i delay within a single device. See application section entitled Memory Cycle Timing for correlation to System Timing requirements.
3. O_i to $\overline{RAS_i}$, ($\overline{RFSH}$ = LOW) skew is guaranteed maximum difference between fastest $\overline{RASI}$ to $\overline{RAS_i}$ delay and slowest $\overline{RFSH}$ to O_i delay within a single device. See application section on Refresh Timing for correlation to system refresh timing requirements.
4. O_i to $\overline{RAS_i}$ (MSEL ⌝⌞) skew is guaranteed maximum difference between fastest MSEL ⌝⌞ to O_i delay and slowest $\overline{RASI}$ to $\overline{RAS_i}$ delay within a single device.

TIMING DIAGRAM

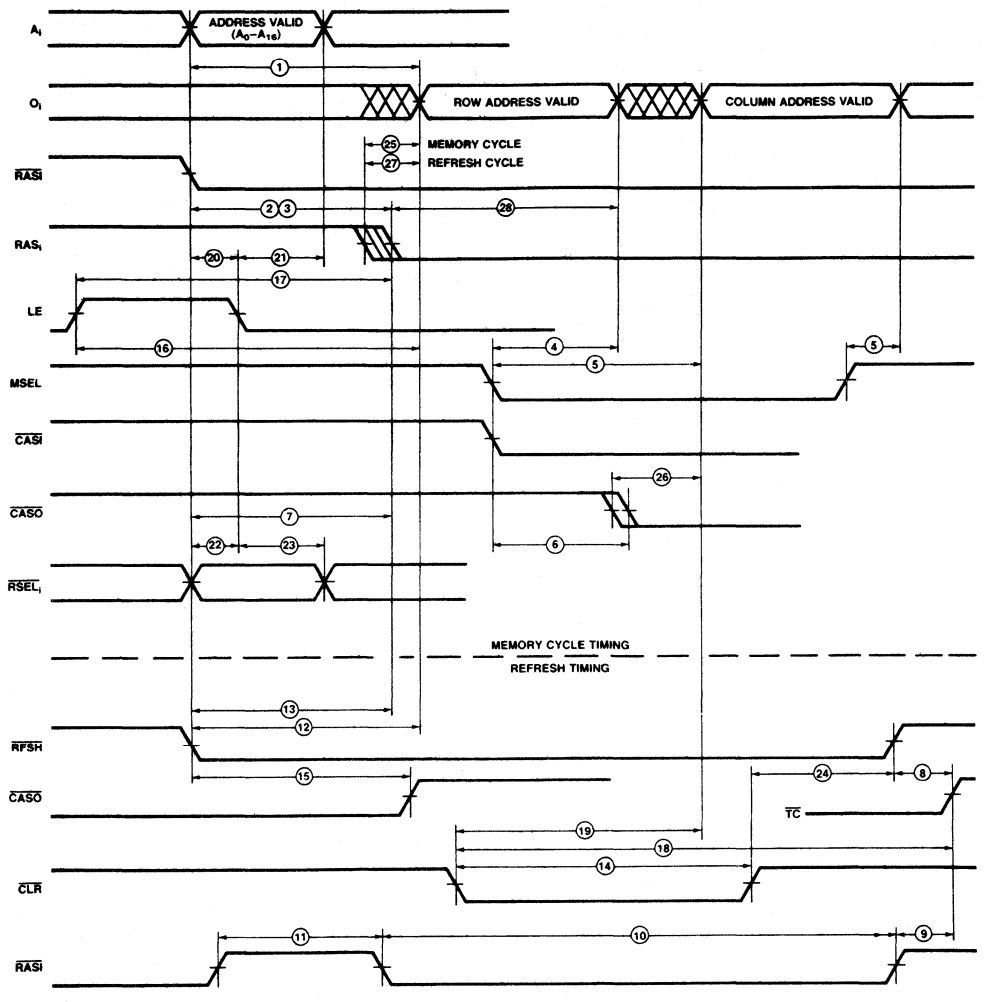


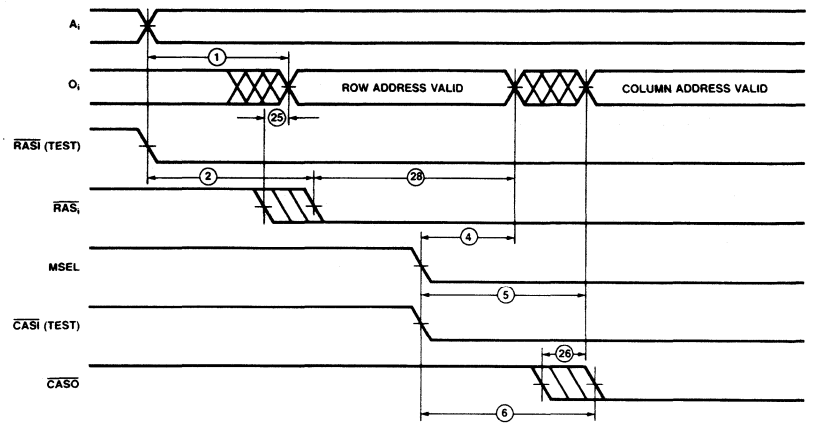
Figure 2. Dynamic Memory Controller Timing

MEMORY CYCLE TIMING

The relationship between DMC specifications and system timing requirements are shown in Figure 3. T_1, T_2, and T_3 represent the minimum timing requirements at the DMC inputs to guarantee that DRAM timing requirements are met and that maximum system performance is achieved.

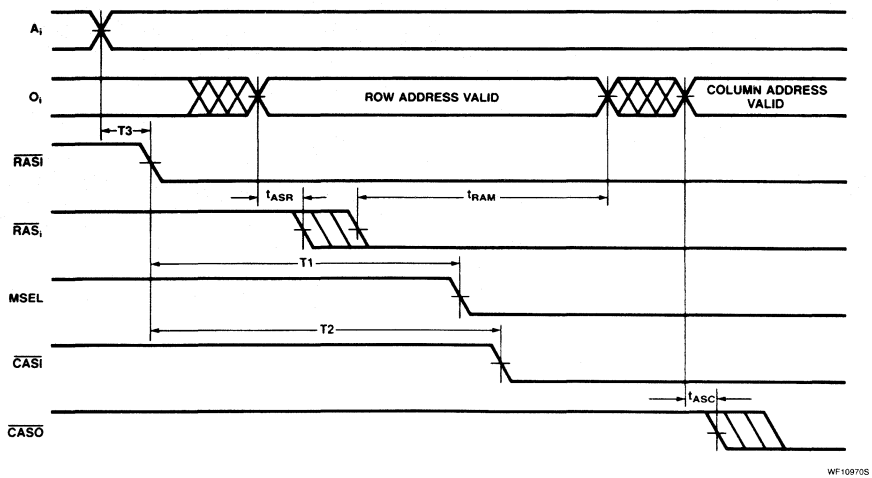
The minimum requirements for T_1, T_2, and T_3 are as follows:

$T_1MIN = t_{RAH} + t_{28}$
$T_2MIN = t_1 + t_{26} + t_{ASC}$
$T_3MIN = t_{ASR} + t_{25}$ where,
t_{RAH} = Row Address Hold Time
t_{ASC} = Column Address Set-up Time
t_{ASR} = Row Address Set-up Time



Legend
㉕ = Guaranteed maximum difference between fastest $\overline{RASI}_1$ to $\overline{RAS}$ delay and the slowest A_1 to O_1 delay on any single device.
㉖ = Guaranteed maximum difference between fastest $\overline{CASI}$ to $\overline{CASO}$ delay and the slowest MSEL to 0_1 delay on any single device.
㉘ = Guaranteed maximum difference between fastest MSEL to O_i delay and the slowest RASI to RAS_i delay on any single device.

a. Specifications Applicable to Memory Cycle Timing



b. Desired System Timing

Figure 3. Memory Cycle Timing

REFRESH CYCLE TIMING

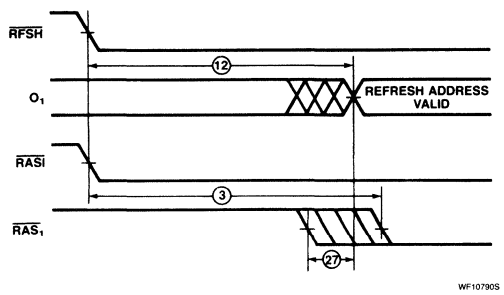
The timing relationships for refresh are shown in Figure 4.

T_4 minimum is calculated as follows:

$$T_4 = t_{ASR} + t_{27}$$

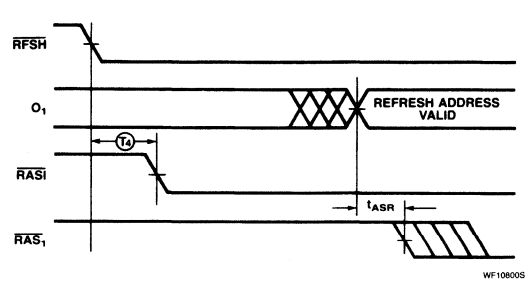
Burst refresh timing is shown in Figure 5.

AC WAVEFORMS



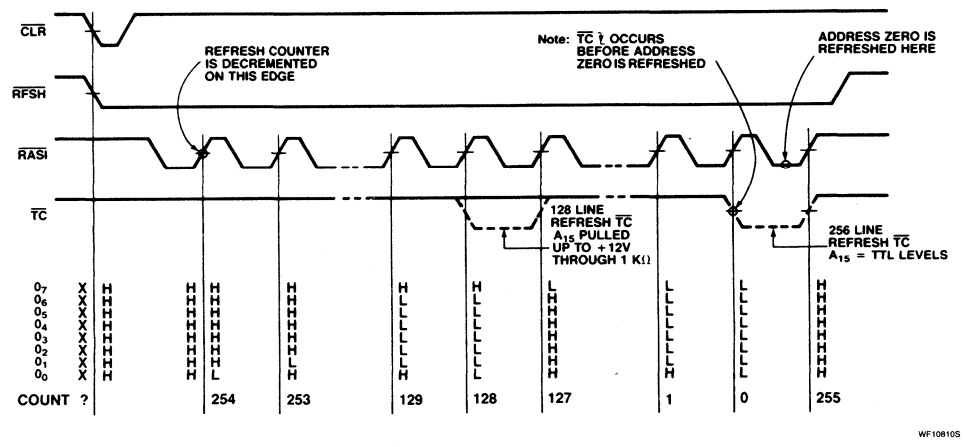
㉗ = Guaranteed maximum difference between fastest $\overline{RASI}_1$ to $\overline{RAS}$ delay and the slowest $\overline{RFSH}$ to O_1 delay on any single device.

Figure 4a. Refresh Timing — Test Waveforms



㉗ = Guaranteed maximum difference between fastest $\overline{RASI}_1$ to $\overline{RAS}$ delay and the slowest $\overline{RFSH}$ to O_1 delay on any single device.

Figure 4b. Refresh Timing — Desired System Timing



NOTE:
The timing shown assumes that burst mode applications may power-down the 2964B with the RAM. Therefore, the counter is cleared prior to executing the refresh sequence

Figure 5. Burst Refresh Timing

Dynamic Memory Controller

2964B

ORDERING INFORMATION

Commercial:

N2964BN (Plastic)

N2964BI (Ceramic)

PACKAGE DATA

Type: Plastic or Ceramic
Configuration: DIP
Width: 0.6 in.
Length: 2.0 in.
Pin Centers: 0.1 in.

APPLICATIONS

Speed with Minimum Skew

The DMC provides Schottky speed in all of the critical paths. In addition, time skew between the Address, RAS and CAS paths is minimized (and specified) by placing these functions on the same chip. The inclusion of the CAS buffer allows matching of its propagation delay, and also provides the CAS inhibit function during RAS — only refresh.

Input Latches

The eighteen input latches are transparent when LE is HIGH and latch the input data, meeting the set-up and hold time requirements when LE goes LOW. In systems with separate address and data buses, LE may be permanently enabled HIGH.

Refresh Counter

The 8-bit refresh counter provides both 128 and 256 line refresh capability. Refresh control is external to allow maximum user flexibility. Transparent (hidden), burst, synchronous or asynchronous refresh modes are all possible.

The refresh counter is advanced at the LOW-to-HIGH transition of RFSH (or RASI). This assures a stable counter output for the next refresh cycle. The counter will continue to cycle through 256 addresses unless reset to zero by CLR. This actually causes all output to go HIGH since the output MUX is inverting. (Address inputs to outputs are non-inverting since both the input latches and output MUX are inverting.)

Refresh Terminal Count

The refresh counter also provides a Terminal Count output for burst mode refresh applications. TC normally occurs at count 255 (00_7 to 0_7 all LOW when RFSH is LOW). TC can be made to occur at count 127 for 128 line burst mode refresh by pulling A_{15} up to +12V through a 1KΩ ±10% resistor. The counter actually cycles through 256 with TC determined by A_{15}. Otherwise A_{15} functions as an address input when driven at normal TTL levels.

Three-Input 8-Bit Address Multiplexer

The address MUX is 8-bits wide (for 64K DRAMs) and has three data sources, the lower address input latch (A_0 to A_7), the upper address input latch (A_8 to A_{15}), and the internal refresh counter. The lower address latch is selected when MSEL is HIGH. This is normally the Row address. The upper address latch is selected when MSEL is LOW. This is normally the Column address. The third source — the refresh counter is selected when RFSH is LOW and overrides MSEL.

When RFSH goes LOW, the MUX selects the refresh counter address and CASO is inhibited. Also, the RAS Decoder function is changed from one-of-four to four-of-four so all RAS outputs $RAS_0 - RAS_3$ go low to refresh all banks of memory when RASI goes LOW. When RFSH is HIGH, only one RAS output goes low, determined by the RAS Select inputs, $RSEL_0$ and $RSEL_1$. In either case the RAS Decoder output timing is controlled by RASI to make sure the refresh count appears at $00_0 - 0_7$ before $RAS_0 - RAS_3$ goes LOW. This assures meeting Row address Set-up time requirement of the DRAM (t_{ASR}).

Maximum Performance System

The typical organization of a maximum performance 16-bit system including Error Detection and Correction is shown in Figure 6. Delay lines provide the most accurate timing and are recommended for RAS, MSEL, and CAS timing in this type of system.

Controlling 16K RAMS or Smaller Systems

16K DRAMs require seven address inputs and 128 line refresh. Also A_0 is often used to designate upper or lower byte transactions in 16-bit systems. These modifications are shown in Figure 7.

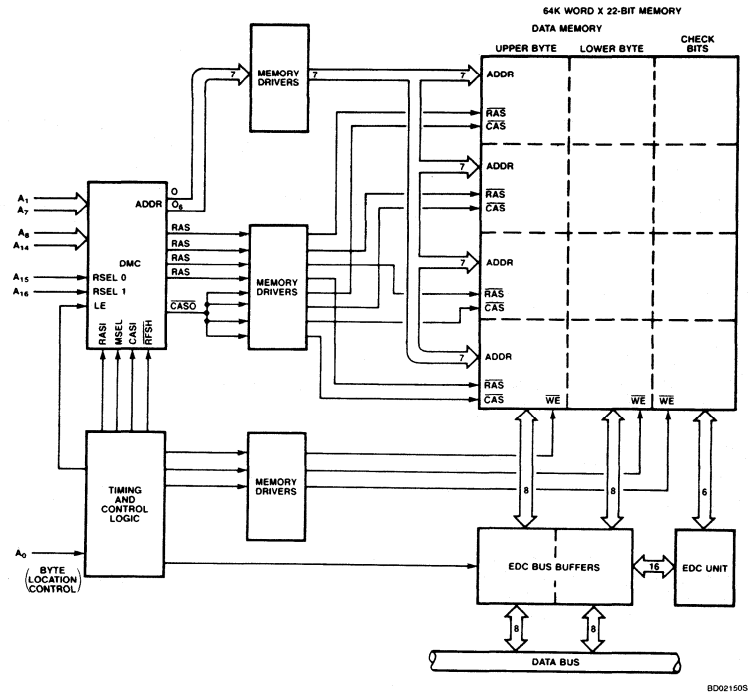
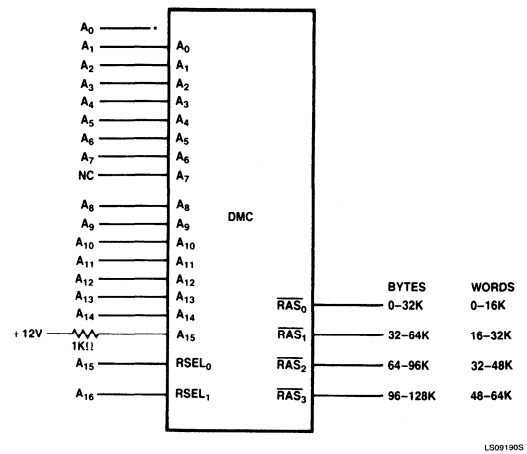


Figure 6. Dynamic Memory Control with Error Detection and Correction



*A_0 controls byte select logic.

Figure 7. Word Organized Memory Using 16K RAMs

Signetics

9403
64-Bit FIFO Buffer Memory

Product Specification

Logic Products

FEATURES

- **10MHz Serial or Parallel Data Rate**
- **Serial or Parallel Input and Output**
- **Expandable Without External Logic**
- **Three-State Outputs**
- **Fully TTL-Compatible**
- **Slim (0.4 In.) 24-Pin DIP**

PRODUCT DESCRIPTION

The 9403 is an expandable fall-through type First-In First-Out (FIFO) Buffer Memory that is optimized for high-speed disc/tape controllers and communication-buffer applications. In multiples of four, the device can be expanded to any number of bits and subsequently, to any number of words. Serial or parallel data can be asynchronously entered or retrieved which makes the 9403 *the* cost-effective solution for implementing buffer memories.

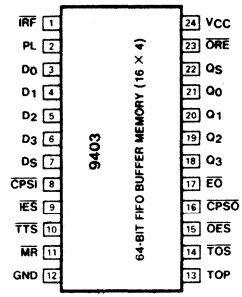
FUNCTIONAL DESCRIPTION

As shown in Figure 1, the 9403 consists of three parts which operate asynchronously and are virtually independent. These parts are:

- **Input Register —** with serial and parallel data inputs and control signals that permit easy expansion and a handshake interface.
- **FIFO Stack —** 4-bit wide, 14-word deep fall-through type with self-contained control logic.
- **Output Register —** with serial and parallel data outputs and control signals that permit easy expansion and a handshake interface.

PIN DESIGNATIONS & DESCRIPTIONS

N PACKAGE



TOP VIEW

ORDER NUMBER

N9403N

MNEMONIC AND FUNCTION		DESCRIPTION
$\overline{IRF}$	Input register full output	Low when input register is full
PL	Parallel load input	High on PL enables $D_0 - D_3$; not edge-triggered, 1's catching
$D_0 - D_3$	Parallel data input	—
D_S	Serial data input	—
$\overline{CPSI}$	Serial input clock	Edge-triggered and activates on falling edge
$\overline{IES}$	Serial input enable	When low, serial input is enabled
$\overline{TTS}$	Transfer to stack input	When low, initiates fall-through
$\overline{MR}$	Master Reset	Active low
TOP	Transfer out parallel input	When high and $\overline{TOS}$ is low, enables word transfer from stack to output register — not edge-triggered
$\overline{TOS}$	Transfer out serial input	When low and TOP is high, enables word transfer from stack to output register — not edge-triggered
$\overline{OES}$	Serial output enable input	When low, enables serial output
$\overline{CPSO}$	Serial output clock input	Edge-triggered and activates on falling edge
$\overline{EO}$	Output enable	Active low
$Q_0 - Q_3$	Parallel data ouput	
Q_S	Serial data output	
$\overline{ORE}$	Output register empty output	When high, output register contains valid data
GND	Ground	—
V_{CC}	Supply voltage	+5 volts

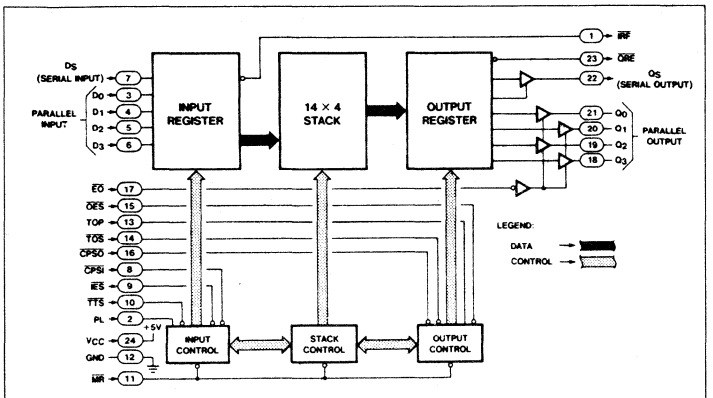


Figure 1. Simplified Block Diagram of 9403 Buffer Memory

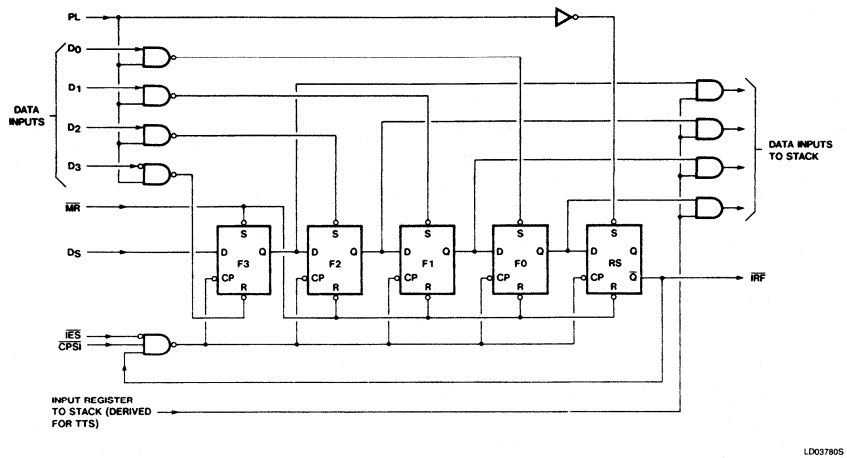


Figure 2. Functional Equivalent of Input Register

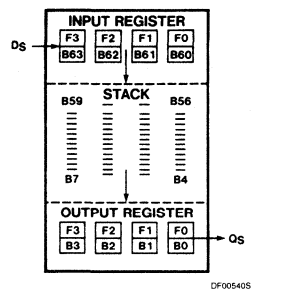


Figure 3. Final Bit Positions Resulting from a Serial Train of 64-Bits

INPUT REGISTER

Data can be entered serially or, using the parallel mode of operation, data is entered in 4-bit increments. In either case, the data is subsequently transferred to the fall-through stack; the functional equivalent of this register is shown in Figure 2. The **I**nput **R**egister **F**ull ($\overline{IRF}$) status signal is internally generated by the **R**egister **S**tatus (RS) flip-flop; when initialized, the Q ($\overline{IRF}$) output of this flip-flop is high.

Serial Entry (Input Register)

Serial data is entered via the D_S input and is handled by a 5-bit shift register consisting of flip-flops F3, F2, F1, F0, and RS. With $\overline{IES}$ and PL both low, each high-to-low transition of the serial input clock ($\overline{CPSI}$) shifts the input data in domino order from F3 to F2 to F1 to F0. After the fourth clock transition, the four bits of serial data are aligned in F3 through F0 and RS is set, forcing $\overline{IRF}$ low and inhibiting $\overline{CPSI}$ until contents of the input register are transferred to the stack. Figure 3 shows how a serial train of 64-bits would appear in the 9403 — four bits (B60 – B63) in the input register, 56 bits (B4 – B59) in the stack, and four bits (B0 – B3) in the output register.

Parallel Entry (Input Register)

When PL is high and $\overline{CPSI}$ is low (Figure 2), flip-flops F0 – F3 are loaded with data and $\overline{IRF}$ is forced low. This condition remains until current data is transferred to the stack. Once the data is transferred, $\overline{IRF}$ is driven high and new data can again be clocked into the input flip-flops. If parallel expansion is not being implemented, $\overline{IES}$ must be low to establish row mastership — refer to discussion of parallel expansion.

STACK OPERATION

As shown in Figure 2, the outputs of F0 – F3 are applied to the stack under control of a signal derived from $\overline{TTS}$. When $\overline{TTS}$ is low, an attempt to inititate a fall-through action is made. If the top location of stack is empty, data is loaded and the input register is re-initialized provided PL is low. Note that initialization is postponed until PL is again low. Thus, automatic FIFO action is achieved by connecting the $\overline{TTS}$ input to the $\overline{IRF}$ output.

The RS flip-flop (Figure 2) records the fact that data has been transferred to the stack; this flip-flop is not cleared until PL goes low. Therefore, if a particular data word is transferred to the stack and falls to the second location before PL goes low, the same word will not be re-transferred even though $\overline{IRF}$ and $\overline{TTS}$ are still low. Once data enters the stack, "fall-through" is automatic; a delay is necessary only when waiting for the next stack location to empty. In the 9403, as in most modern FIFO designs, the $\overline{MR}$ input initializes the stack control section and does not clear the data.

OUTPUT REGISTER

This register receives and stores 4-bits of data from the bottom stack location and, on demand, outputs data onto a three-state 4-bit parallel data bus or a three-state serial data bus. The **O**utput **R**egister Full ($\overline{ORE}$) status signal is internally-generated by the FX flip-flop; when data is transferred from the stack to the output register, $\overline{ORE}$ goes high. The functional equivalent of the output register is shown in Figure 4.

Retrieval of Serial Data

When the FIFO stack is empty and $\overline{MR}$ is driven low, the $\overline{ORE}$ output goes low to indicate that the output register is ready to accept new data from the stack. After new data is entered and falls through to the bottom stack location, it is transferred to the output register provided $\overline{TOS}$ is low and TOP is high. As a result of the data transfer, $\overline{ORE}$ goes high indicating valid data in the output register. Subsequently, the $\overline{Q}_S$ output is automatically enabled and the first data bit is transmitted to the three-state serial data bus. Henceforth, a serial shift of data occurs on each high-to-low transition of $\overline{CSPO}$. On the fourth transition, the register is emptied, $\overline{ORE}$ is forced low, and serial output $\overline{Q}_S$ is diabled. To request a new word from the stack, the $\overline{TOS}$ input can be connected to the $\overline{ORE}$ output.

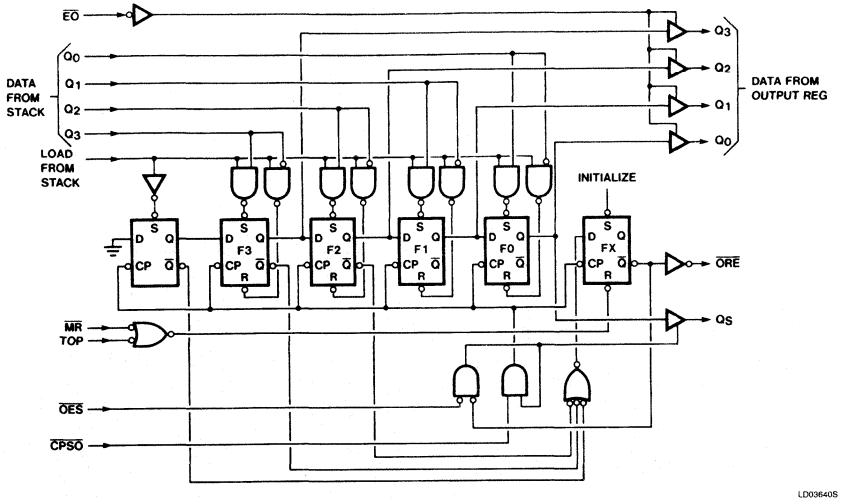


Figure 4. Functional Equivalent of Output Register

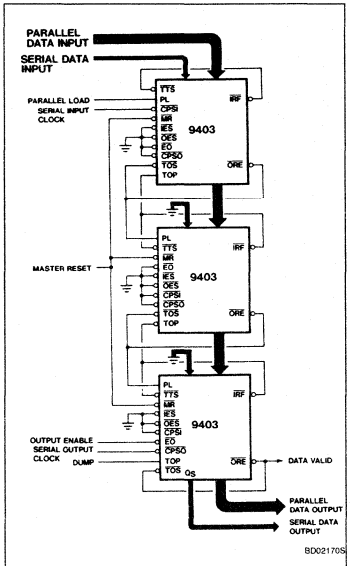


Figure 5. Word Expansion

Retrieval of Parallel Data

With the stack empty and $\overline{MR}$ in the active-low state, the $\overline{ORE}$ output goes low, signifying that the output register is also empty. When new data is entered and has fallen through to bottom location of the stack, it is automatically transferred to the output register, provided the Transfer Out Parallel (TOP) input is high. When the data is transferred from stack-to-register, $\overline{ORE}$ goes high and valid data appears at $Q_0 - Q_3$ (Figure 4), provided the three-state buffers are enabled, that is, $\overline{EO}$ is active-low. When TOP goes low, $\overline{ORE}$ is driven low which indicates that the data output cycle is complete; however, the original data remains latched in the flip-flops until the next word (if available) is transferred from the stack to the output register.

For parallel operation, $\overline{CPSO}$ must be low, whereas, $\overline{TOS}$ should be grounded for single-slice operation or connected to the appropriate $\overline{ORE}$ for expanded operation. The TOP input is not edge-triggered; therefore, if it goes high before data is available from stack but data becomes available before it goes low, the data will be transferred to the output register. However, internal control circuits prevent the same data from being transferred twice. If TOP goes high and returns to low before data is available from the stack, $\overline{ORE}$ will remain low, indicating the absence of valid output data.

VERTICAL EXPANSION

In a vertical structure, the 9403 can be expanded to achieve greater word capacity without any external parts; a 46-word by 4-bit FIFO is shown in Figure 5. Using the same technique and similar connections, any FIFO of 15n + 1 words (where *n* is the number of devices) can be constructed. Observe that word expansion does not sacrifice flexibility of the 9403 FIFO as regards serial/parallel input and output.

HORIZONTAL EXPANSION

The 9403 can be horizontally expanded to store long words in multiples of 4-bits, again without external logic. Connections required to form a 16-word by 12-bit FIFO are shown in Figure 6, using similar techniques, any 16-word by 4n-bit FIFO (where *n* is the number of devices) can be constructed.

For horizontal or bit expansion, it is good practice to connect, respectively, the $\overline{IRF}$ and $\overline{ORE}$ ouputs of the right-most device (most significant device) to the $\overline{TTS}$ and $\overline{TOS}$ inputs of all devices to the left (least significant devices) to guarantee that no operation is initiated before each and every device is ready. Word expansion does not affect the ability of the 9403 to handle serial/parallel inputs and outputs; however, the ripple form of expansion shown in Figure 6 does extract a penalty in speed of operation. Whereas a single 9403 is guaranteed to operate at 10MHz, an array of four FIFOs connected as shown is guaranteed to operate at 4.3MHz.

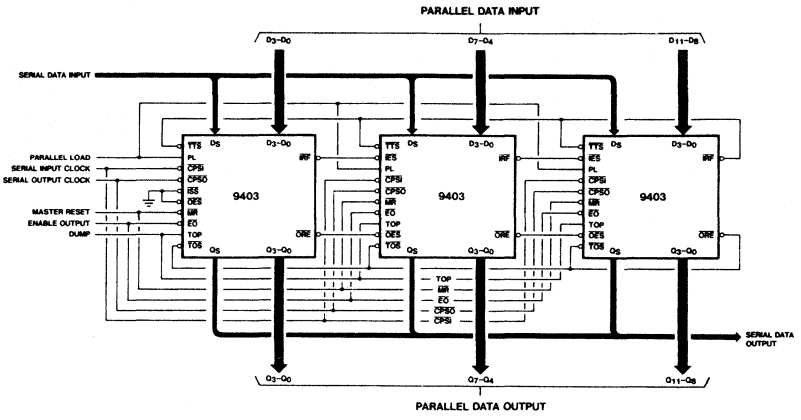
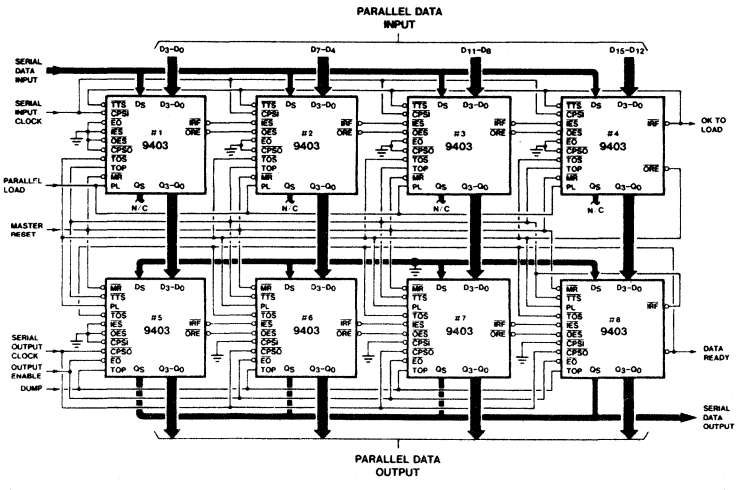


Figure 6. Bit Expansion



INPUT REGISTER	UNIT NUMBER & ORDER OF SERIAL BITS #1	#2	#3	#4	OUTPUT REGISTER	UNIT NUMBER & ORDER OF SERIAL BITS #5	#6	#7	#8
D_3	BIT 483	BIT 487	BIT 491	BIT 495	Q_3	BIT 3	BIT 7	BIT 11	BIT 15
D_2	482	486	490	494	Q_2	2	6	10	14
D_1	481	485	489	493	Q_1	1	5	9	13
D_0	480	484	488	492	Q_0	0	4	8	12

Figure 7. Horizontal and Vertical Expansion — 31 x 16 FIFO

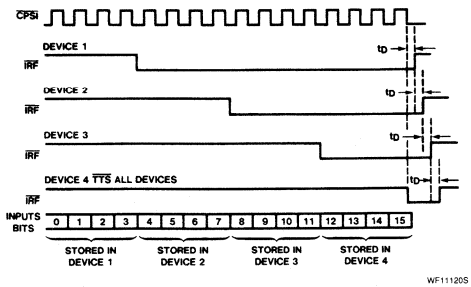


Figure 8. Entry of Serial Data for Array of Figure 7

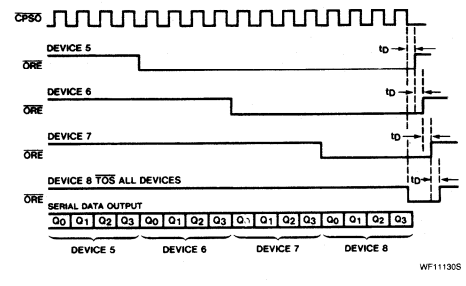


Figure 9. Retrieval of Serial Data for Array of Figure 7

HORIZONTAL AND VERTICAL EXPANSION

In addition to bit-or-word expansion, the 9403 can be used to expand in both the horizontal and vertical directions; a 31-word by 16-bit FIFO is shown in Figure 7. Using the same or similar techniques, any FIFO of 15m + 1 words by 4n-bits can be constructed, where *m* is the number of devices in a column and *n* is the number of devices in a row.

The chart appended to Figure 7 shows the final positions for a contiguous serial entry of 496 bits. Figures 8 and 9, respectively, show the timing relationships involved for data-entry and data-retrieval pertaining to the 31-word by 16-bit array.

INTERLOCKING CIRCUITS

Most conventional FIFO designs provide the status-signal counterparts of $\overline{IRF}$ and $\overline{ORE}$. However, when these devices are used in arrays, variations in unit-to-unit operating speeds require the use of external gating to ensure that all devices have, in fact, completed the last operation. The 9403 incorporates simple but effective master/slave interlocking circuits to eliminate these gating requirements.

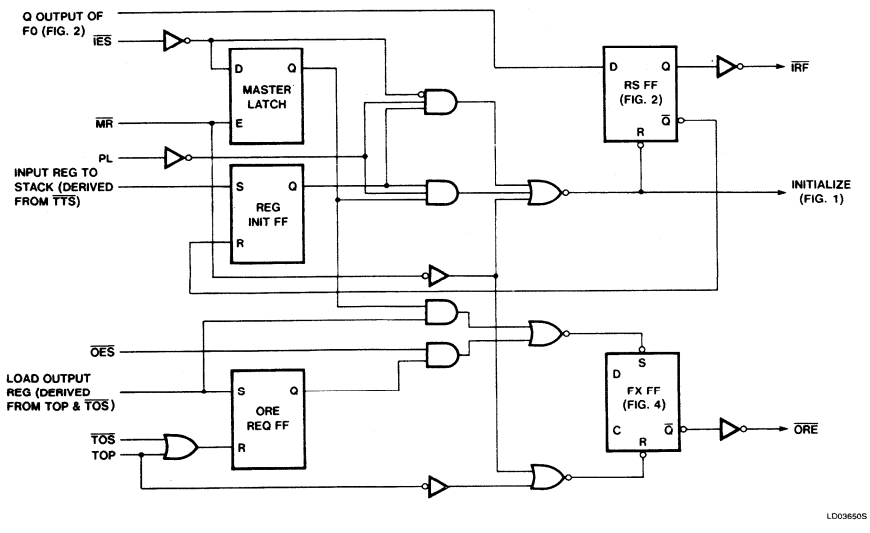


Figure 10. Functional Equivalent of Interlocking Circuits

ABSOLUTE MAXIMUM RATINGS

PARAMETER		RATING	UNIT
V_{CC}	Power supply voltage	+7	Vdc
V_{IN}	Input voltage	+5.5	Vdc
V_O	Off-state output voltage	+5.5	Vdc
T_A	Operating temperature range	0 to +70	°C
T_{stg}	Storage temperature range	−65 to +150	°C

DC ELECTRICAL CHARACTERISTICS Over operating temperature range unless otherwise noted

PARAMETER		TEST CONDITIONS[1, 2]	LIMITS			UNIT
			Min	Typ	Max	
V_{IH}	Input high voltage	Guaranteed input high voltage	2.0			V
V_{IL}	Input low voltage	Guaranteed input low voltage			0.8	V
V_{CD}	Input clamp diode voltage	V_{CC} = MIN, I_{IN} = −18mA		−0.9	−1.5	V
V_{OH}	Output high voltage, $\overline{ORE}$, $\overline{IRF}$	V_{CC} = MIN, I_{OH} = −400μA	2.4	3.4		V
V_{OH}	Output high voltage, $Q_0 - Q_3$, Q_S	I_{OH} = −5.7mA, V_{CC} = MIN	2.4	3.1		V
V_{OL}	Output low voltage, $Q_0 - Q_3$, Q_S	V_{CC} = MIN, I_{OL} = 16mA		0.35	0.5	V
V_{OL}	Output low voltage, $\overline{ORE}$, $\overline{IRF}$	V_{CC} = MIN, I_{OL} = 8.0mA		0.35	0.5	V
I_{OZH}	Output off current high, $Q_0 - Q_3$, Q_S	V_{CC} = MAX, V_{OUT} = 2.4V, V_E = 2V			100	μA
I_{OZL}	Output off current low, $Q_0 - Q_3$, Q_S	V_{CC} = MAX, V_{OUT} = 0.5V, V_E = 2V			−100	μA
I_{IH}	Input high current	V_{CC} = MAX, V_{IN} = 2.7V V_{CC} = MAX, V_{IN} = 5.5V		1.0	40 1.0	μA mA
I_{IL}	Input low current, all except $\overline{OES}$ & $\overline{IES}$	V_{CC} = MAX, V_{IN} = 0.4V			−0.36 −0.96	mA
I_{OS}	Output short circuit current, $Q_0 - Q_3$, Q_S, $\overline{ORE}$, $\overline{OES}$	V_{CC} = MAX, V_{OUT} = 0, (Note 3)	−30		−130	mA
I_{CC}	Supply Current	V_{CC} MAX, Inputs open		115	170	mA

NOTES:
1. Operating temperature ranges are guaranteed after terminal equilibrium has been reached.
2. All voltages measured with respect to ground terminal.
3. No more than one output should be shorted at a time.

AC ELECTRICAL CHARACTERISTICS V_{CC} = 5.0V, C_L = 15pF, T_A = 25°C

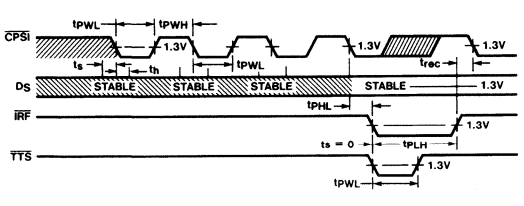
PARAMETER	FROM INPUT	TO OUTPUT	TEST CONDITIONS[1, 2, 3]	LIMITS Min	LIMITS Typ	LIMITS Max	UNIT
Fall-through time: t_{DFT}	Positive going PL	$Q_0 - Q_3$	$\overline{TTS}$ connected to $\overline{IRF}$, $\overline{TOS}$ connected to $\overline{ORE}$, $\overline{IES}$, $\overline{OES}$, $\overline{EO}$, $\overline{CPSO}$ low. TOP high (f, Fig. 11)		450	600	ns
Propagation delay							
t_{PLH} Low-to-high	Negative going $\overline{TTS}$	$\overline{IRF}$	Stack not full. PL low		48	64	ns
t_{PHL} High-to-low	Negative going $\overline{CPSI}$	$\overline{IRF}$	(a & b, Fig. 11)		18	25	
t_{PLH} Low-to-high	Negative going $\overline{CPSO}$	Q_S	Serial output $\overline{OES}$ low.		30	40	ns
t_{PHL} High-to-low			TOP high (c & d, Fig. 11)		17	28	
t_{PHL} High-to-low	Negative going $\overline{CPSO}$	$\overline{ORE}$			32	42	ns
t_{PLH} Low-to-high	Positive going TOP	$Q_0 - Q_3$	$\overline{EO}$, $\overline{CPSO}$ low (e, Fig 11)		40	56	ns
t_{PHL} High-to-low					31	45	
t_{PLH} Low-to-high	Positive going TOP	$\overline{ORE}$	Parallel output.		51	68	ns
t_{PHL} High-to-low	Negative going TOP	$\overline{ORE}$	$\overline{EO}$, $\overline{CPSO}$ low (e, Fig. 11)		40	54	
t_{PLH} Low-to-high	Negative going $\overline{TOS}$	Positive going $\overline{ORE}$	Data in stack. TOP high, (c & d, Fig. 11)		41	56	ns
t_{PHL} High-to-low	Positive going PL	Negative going $\overline{IRF}$	Stack not full (g & h, Fig. 11)		20	33	ns
t_{PLH} Low-to-high	Negative going PL	Positive going $\overline{IRF}$			33	46	ns
t_{PLH} Low-to-high	Positive going $\overline{OES}$	$\overline{ORE}$			26	44	
t_{PLH} Low-to-high	Positive going $\overline{IES}$	Positive going $\overline{IRF}$			31	40	
Enable delay:							
t_{PZH} High	$\overline{EO}$	$Q_0 - Q_3$	Out of high impedance state		9	14	ns
t_{PZL} Low						20	
t_{PZL} Low	Negative going $\overline{OES}$	Q_S			13	25	ns
t_{PZH} High						20	
Disable delay:							
t_{PLZ} Low	$\overline{EO}$	$Q_0 - Q_3$	Into high impedance state		7	14	ns
t_{PZH} High							
t_{PLZ} Low	Negative going $\overline{OES}$	Q_S			7	14	ns
t_{PHZ} High							
Appearance time:							
t_{AP} Parallel	$\overline{ORE}$	$Q_0 - Q_3$	Time elapsed between $\overline{ORE}$ going high and valid data appearing at output, negative number indicates data available before $\overline{ORE}$ goes high		−12	−5	ns
t_{AS} Serial	$\overline{ORE}$	Q_S			6	10	
Pulse width:							
t_{PWL} $\overline{CPSI}$ low			Stack not full. PL low	20	11		ns
t_{PWH} $\overline{CPSI}$ high			(a & b, Fig. 11)	33	19		
t_{PWL} TOP low			$\overline{CPSO}$ low, data available	30	17		ns
t_{PWH} TOP high			in stack (e, Fig. 11)	26	13		
t_{PWL} $\overline{CPSO}$ low			TOP high, data in stack,	30	16		ns
t_{PWH} $\overline{CPSO}$ high			(c & d, Fig. 11)	32	18		
t_{PWH} PL high			Stack not full (g & h, Fig. 11)	40	29		ns
t_{PWL} $\overline{TTS}$ low (serial or parallel mode)			Stack not full (a, b, g, & h, Fig. 11)	20	9		ns
t_{PWL} $\overline{MR}$ low			(f, Fig. 11)	25	13		ns

AC ELECTRICAL CHARACTERISTICS V_{CC} = 5.0V, C_L = 15pF, T_A = 25°C (Cont.)

PARAMETER	FROM INPUT	TO OUTPUT	TEST CONDITIONS[1, 2, 3]	LIMITS Min	LIMITS Typ	LIMITS Max	UNIT
Set-up and hold time:							
t_s Set-up time	D_S	Negative $\overline{CPSI}$	PL low (a & b, Fig. 11)	28	17		ns
t_h Hold time	D_S	$\overline{CPSI}$		0	−6		
t_s Set-up time	Parallel inputs	PL	Length of time parallel inputs must be applied prior to rising edge of PL	0	−22		ns
t_h Hold time	Parallel inputs	PL	Length of time parallel inputs must remain applied after falling edge of PL	2			ns
t_s Set-up time (serial or parallel mode)	$\overline{TTS}$	$\overline{IRF}$	(a, b, g, & h, Fig. 11)	0	−20		ns
t_s Set-up time	Negative going $\overline{ORE}$	Negative going $\overline{TOS}$	TOP high (c & d, Fig. 11)	0	−24		ns
t_s Set-up time	Negative going $\overline{IES}$	$\overline{CPSI}$	(b, Fig. 11)	45	23		ns
t_s Set-up time	Negative $\overline{TTS}$	$\overline{CPSI}$		84	58		
Recovery time:							
t_{rec}	$\overline{MR}$	Any input	(f, Fig. 11)	15	5		ns

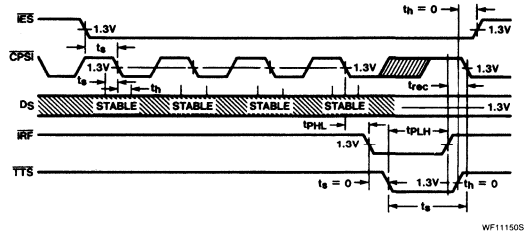
NOTES:
1. Initialization requires a master reset to occur after power has been applied.
2. $\overline{TTS}$ normally connected to $\overline{IRF}$.
3. If stack is full, $\overline{IRF}$ will stay low.

TEST CIRCUITS AND WAVEFORMS



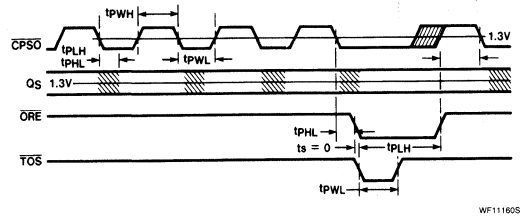
CONDITIONS:
Stack Not Full, $\overline{IES}$, PL Low

a. Serial Input, Unexpanded or Master Operation



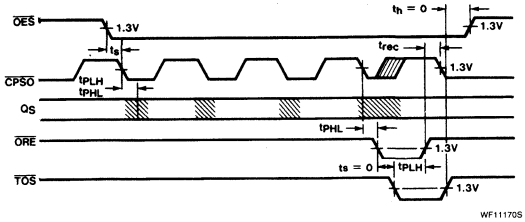
CONDITIONS:
Stack Not Full, $\overline{IES}$ High When Initialized, PL Low

b. Serial Input, Expanded Slave Operation



CONDITIONS:
Data in Stack, Top High, $\overline{IES}$ Low When Initialized, $\overline{OES}$ Low

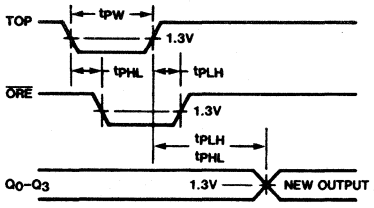
c. Serial Output, Unexpanded or Master Operation



CONDITIONS:
Data in Stack, Top High, $\overline{IES}$ High When Initialized

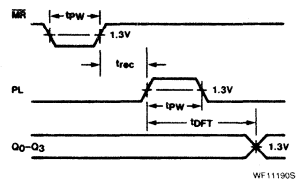
d. Serial Output, Slave Operation

TEST CIRCUITS AND WAVEFORMS (Continued)



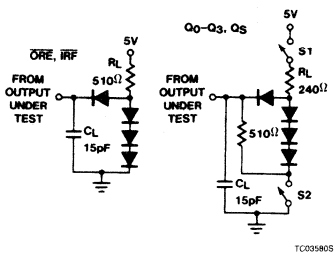
CONDITIONS:
IES Low When Initialized, EO, CPSO Low,
Data Available in Stack

e. Parallel Output, 4-Bit Word or Master/Slave in Parallel Expansion



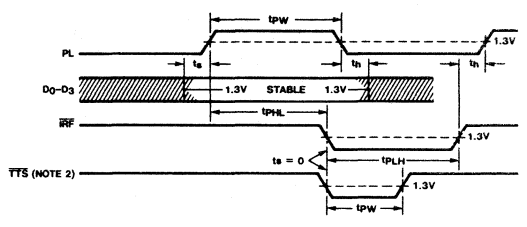
CONDITIONS:
TTS Connected to IRF, TOS Connected to
ORE, IES, OES, EO, CPSO Low, Top High

f. Fall-Through Time



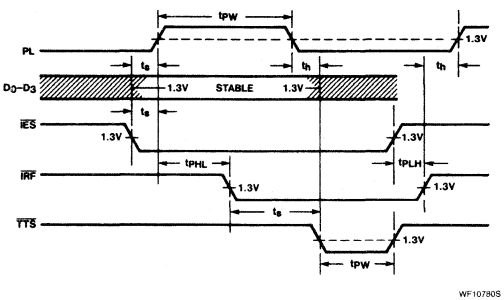
NOTES:
1. C_L includes jig and probe capacitance
2. All diodes are 1N3064 or equivalent

g. Test Circuit for Measurement of AC Parameters



CONDITIONS:
Stack Not Full, IES Low When Initialized

h. Parallel Load Mode, 4-BIT Word (Unexpanded) or Master/Slave Operation in Parallel Expansion



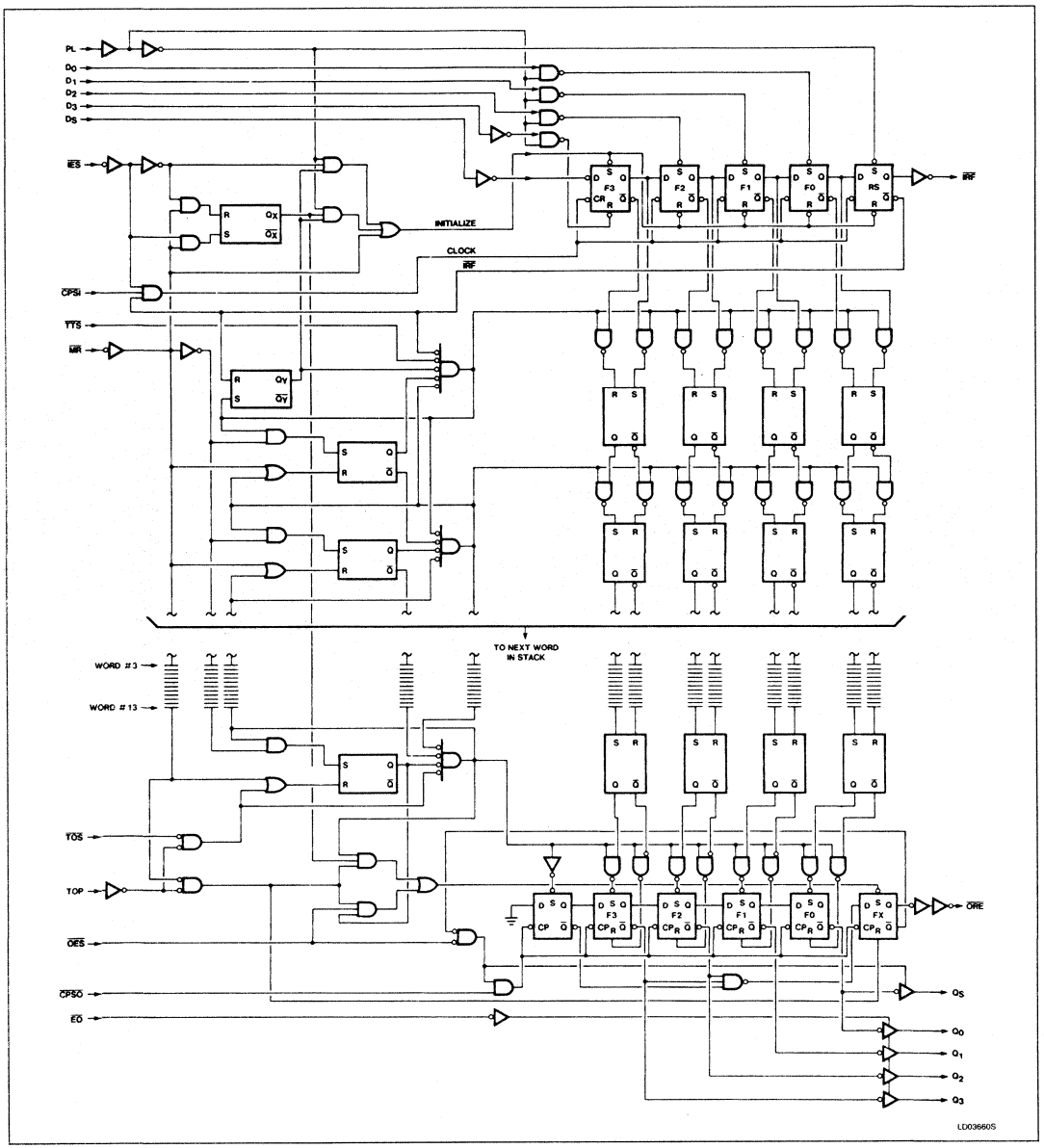
CONDITIONS:
Stack Not Full, Device Initialized With IES High

j. Parallel Load, Slave Mode

For all waveforms, V_M = 1.3V for 74S; V_M = for all other TTL families.
The shaded areas indicate when the input is permitted to change for predictable output performance.

Figure 11. 9403 Timing and Parameter-Measurement Information

LOGIC DIAGRAM



DIGITAL SIGNAL PROCESSOR (DSP)

SINGLE-CHIP DIGITAL SIGNAL PROCESSOR

HOW TO USE THIS DATA SHEET

- **Section 1** contains ordering information and the main features of the PCB5010 and PCB5011.
- **Section 2** describes the signals of the PCB5010 and PCB5011, with block diagrams and full descriptions of what functions can be performed by each block.
- **Section 3** describes how the blocks are controlled by the instructions given by the programmer. This section is used during programming, it assumes however, a full knowledge of section 2.0. Programming can be simplified by using the software tools available.
- **Section 4** describes all the electrical characteristics. This section is used during the design of system hardware.
- **Section 5** gives details of the PCB5010 and PCB5011 packages.

CONTENTS

1.0 INTRODUCTION

The PCB5010 and PCB5011 are part of our SP 50 family of digital signal processors that contains devices for various applications. These CMOS devices have a common processor structure and are accompanied by a common set of development tools.

The processor structure is characterized by a double data bus, a two operand hardware multiply/accumulate unit and a two operand ALU to improve throughput. Powerful parallel and serial interfaces enable communication with external devices. Large on-chip data memories, each with its own programmable address computation unit (ACU), offer the possibility to make systems with only a few components.

The PCB5010 and PCB5011 are the optimal solutions for implementing DSP functions in telecommunications, and can also be used to advantage in speech processing, high-speed control, image processing and many other fields.

- PCB 5010: Version with on-chip ROM (mask programmable)
- PCB 5011: ROMless bond-out version, for use with external program/data memory

DEVELOPMENT DATA

ORDERING INFORMATION

order number	speed (MHz)	operating ambient temperature (°C)	package
PCB5010WP-8	8	0 to +70	68-pin PLCC
PCF5010WP-8*	8	−40 to +85	68-pin PLCC
PCB5011YC-8	8	0 to +70	144 PGA
PCF5011YC-8*	8	−40 to +85	144 PGA

* The PCF versions will be identical to the PCB versions except that they have an extended operating ambient temperature range. However, minor variations may occur in the AC characteristics.

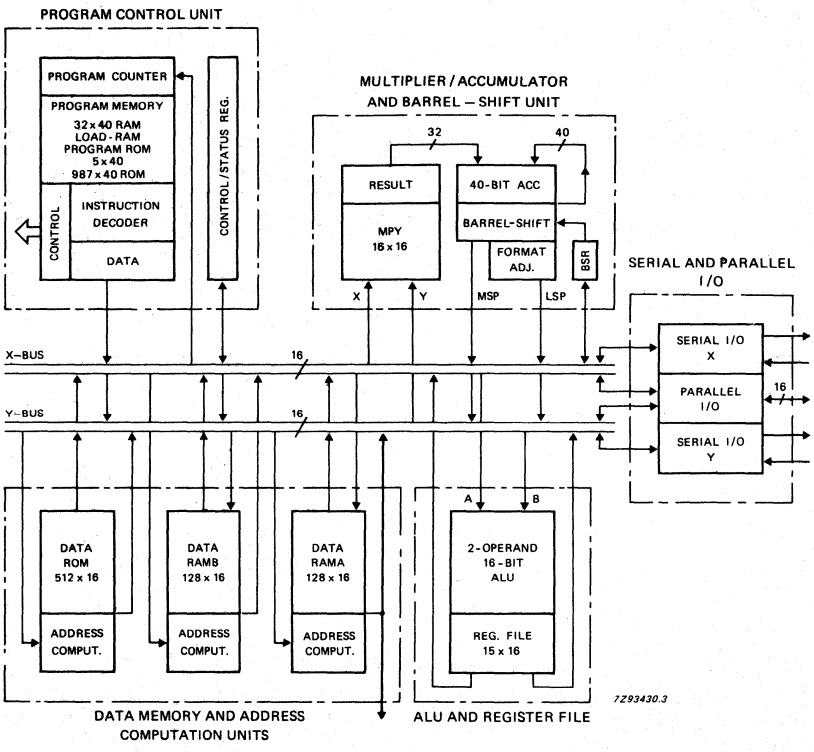


Fig. 1.0-1 Simplified block diagram of the PCB5010.

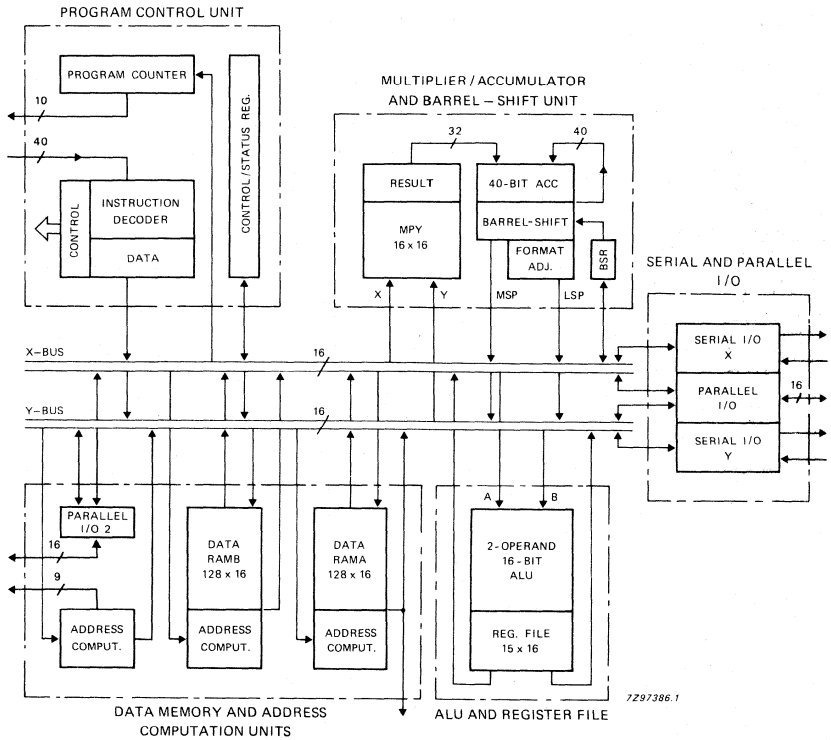


Fig. 1.0-2 Simplified block diagram of the PCB5011.

FEATURES

- Harvard architecture with two data buses of 16 bit width
- 4 instruction types:
  - multiply/accumulate operation + 2 data moves + 3 address calculations/memory read accesses
  - alu operation + 2 data moves + 3 address calculations/memory read accesses
  - load immediate data + 3 address calculations/memory read accesses
  - branch + 3 address calculations/memory read accesses

  Note: a high degree of parallel processing allows up to 6 basic operations to be performed simultaneously.
- Hardware two's complement 16 x 16 multiplier with 40-bit accumulator, full range barrel-shifter and format adjuster:
  - 45 different multiply/accumulate operations
  - multi-precision multiplication support
  - result bit-reversal possibility
  - 4 status flags
- 16-bit 2-operand ALU with:
  - 31 different operations
  - multi-precision operation support
  - 15 x 16-bit 3-port register-file
  - 5 status flags
- Program memory:
  PCB5010: 987 x 40-bit on-chip ROM (mask programmable)
  32 x 40-bit on-chip RAM (loaded via the X-bus)
  5 x 40-bit on-chip ROM (fixed load RAM program)
  PCB5011: 1024 x 40-bit external memory (or 64K x 40-bit when some external logic is added)
- Data memory:
  PCB5010: 512 x 16 bit on-chip ROM (mask programmable)
  2 x (128 x 16) on-chip static RAM
  PCB5011: 512 x 16 bit external memory (read and write possible)
  2 x (128 x 16) on-chip static RAM
- 3 powerful programmable address computation units (ACU's) for the data ROM and both data RAMs and also for 16 pages of 4096 x 16 bit external data memory
  - each ACU has 8 different operations
  - 1 status flag for each ACU
- 5 level deep hardware stack (software extendable)
- 16-bit parallel I/O to access external data memory
  - 8 million words/s
  - WAIT facility so that "slow" peripherals can be connected
- 2 independent serial inputs and outputs (one pair for each data bus), with a maximum speed of 4 million bit/s under the control of external clocks
- 4 user input flags
- Maskable interrupt
- Repeat possibility of single instruction
- Maximum clock rate 8 MHz
- Pipelined (P) and Non-pipelined (NP) modes under programmer control:
  P-mode: a new instruction can start every 125 ns
  NP-mode: a new instruction can start every 250 ns
- Single 5 V power supply (±5%)
- All I/O are TTL compatible
- Operating ambient temperature range:
  PCB5010/11: 0 to +70 °C
  PCF5010/11: −40 to +85 °C

2.0 FUNCTIONAL DESCRIPTION

2.1 GENERAL DESCRIPTION

The detailed block diagram of the PCB5010 and PCB5011 are shown in Fig. 2.1-1 and Fig. 2.1-2. The signals of the processors are described briefly in the table below.

SIGNAL DESCRIPTION

SIGNAL	DESCRIPTION
V_{DD}	Supply voltage: 5V ±5%
V_{SS}	Ground
CLK	Clock (input)
$\overline{SCAN}$	This signal must be grounded
$\overline{RST}$	Reset (input)
D15 . . . D0	16-bit wide parallel I/O port (input/output)
A15 . . . A0	16-bit wide address for 64K-words in external data memory (output)
R/$\overline{W}$	Read/write signal for control of external memory (output)
$\overline{DS}$	Data strobe (output)
$\overline{WAIT}$	Wait signal for synchronization of parallel I/O (input)
DIX	Serial data input (input) for the X-bus
$\overline{SIXEN}$	Serial data enable (input) for the X-bus
$\overline{SIXRQ}$	Serial input request (output) for the X-bus
CIX	Serial input clock (input) for the X-bus
DIY	Serial data input (input) for the Y-bus
$\overline{SIYEN}$	Serial input enable (input) for the Y-bus
$\overline{SIYRQ}$	Serial input request (output) for the Y-bus
CIY	Serial input clock (input) for the Y-bus
DOX	Serial data output (output) for the X-bus
$\overline{SOXEN}$	Serial output enable (input) for the X-bus
$\overline{SOXRQ}$	Serial output request (output) for the X-bus
COX	Serial output clock (input) for the X-bus
DOY	Serial data output (output) for the Y-bus
$\overline{SOYEN}$	Serial output enable (input) for the Y-bus
$\overline{SOYRQ}$	Serial output request (output) for the Y-bus
COY	Serial output clock (input) for the Y-bus
$\overline{INT}$	Maskable interrupt (input)
$\overline{IACK}$	Interrupt acknowledge (output)
SYNC	Synchronization signal; indicates where execution of a new instruction starts (output)
IFA	User flag (input)
IFB	User flag (input)
IFC	User flag (input)
IFD	User flag (input)

(continued on next page)

SIGNAL DESCRIPTION (Cont'd)

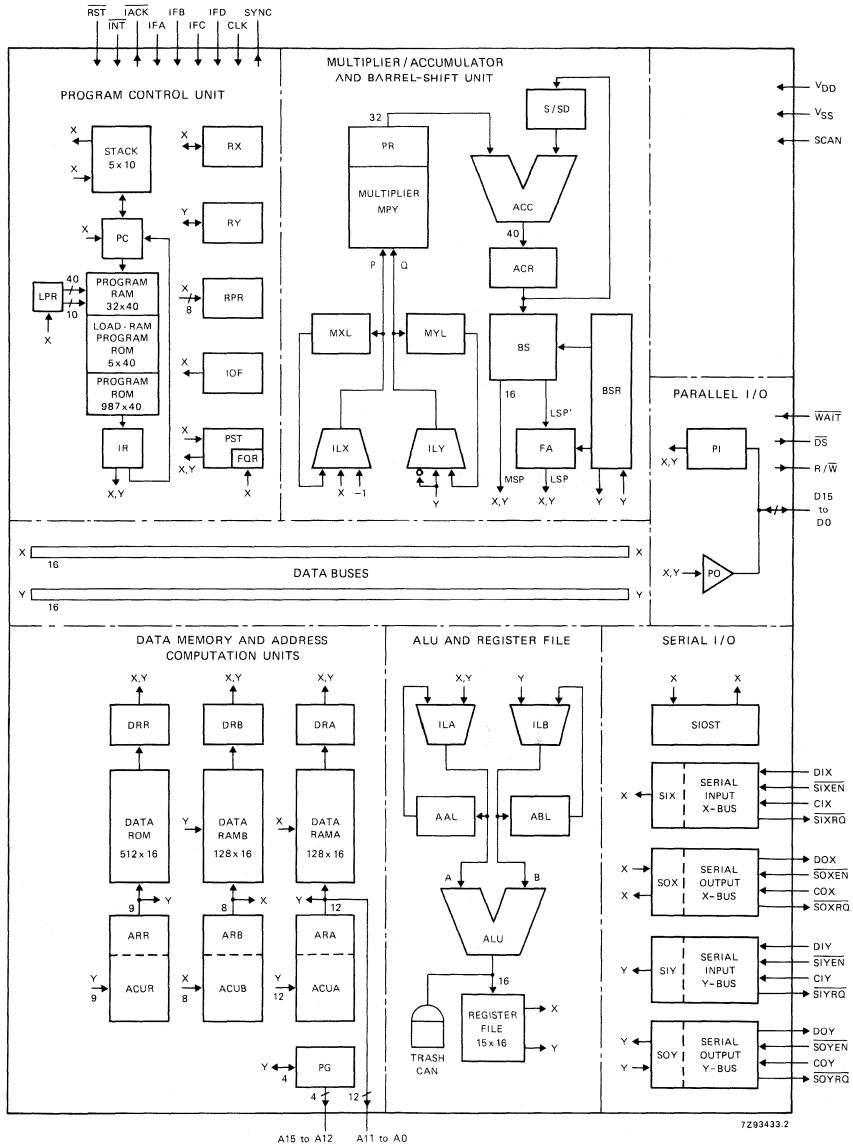
SIGNAL	DESCRIPTION
PCB5011 only:	
PA9 . . . PA0	External program memory address (output)
PD39 . . . PD0	External program word (input)
ARR8 . . . ARR0	9-bit address for 512 words in external data memory (output)
RD15 . . . RD0	Second 16-bit parallel I/O port (input/output)
RR/$\overline{W}$	Read/write signal for second 16-bit parallel I/O port (output)
$\overline{RDS}$	Data strobe for second 16-bit parallel I/O port (output)

The main blocks of PCB5010/11 are:

- Program control unit with:
  - Program ROM (only for PCB5010)
  - IR (instruction register)
  - Sync pin
  - PC (program counter)
  - RPR (instruction repeat register)
  - Stack
  - PST (processor status register)
  - IOF (input/output status and user flag register)
  - User flag pins
  - $\overline{INT}$ pin and $\overline{IACK}$ pin
  - Bus-save registers RX and RY
  - $\overline{RST}$ pin
  - External program memory port (only PCB5011)
  - External program memory address port (only PCB5011)
- Data memory and address computation units with:
  - RAMA, ACUA (address computation unit A), DRA (data register A)
  - PG (page register)
  - RAMB, ACUB (address computation unit B), DRB (data register B)
  - ROM (only PCB5010), ACUR (address computation unit R), DRR (data register R)
  - External data word pins (only PCB5011)
  - External data memory address pins (only PCB5011)
- Multiplier/accumulator and barrel-shift unit with:
  - Input selectors ILX and ILY
  - Latches MXL and MYL
  - MPY (multiplier)
  - Accumulator with ACC (adder), ACR (multiplication/accumulation register) and S/SD (sign/scale-down block)
  - BS (barrelshifter)
  - FA (format adjuster)
  - BSR (barrelshift and format adjust control register)
- ALU and register file with:
  - Input selectors ILA and ILB
  - Latches AAL and ABL
  - ALU (arithmetic logic unit)
  - R1-R15 (register file)
  - Trash can
- Parallel I/O with:
  - PI (parallel data input latch)
  - PO (parallel data output buffer)
  - Parallel I/O data and control pins
- Serial I/O with:
  - SIX (serial input latch connected to X bus)
  - SOX (serial output latch connected to X bus)
  - SIY (serial input latch connected to Y bus)
  - SOY (serial output latch connected to Y bus)
  - SIOST (serial I/O control register)
  - Serial I/O data and control pins
- Data buses with:
  - 16 bits X-bus
  - 16 bits Y-bus

The working of the main blocks is described in the following sections. The instruction set is described in the section 3.0.

DEVELOPMENT DATA

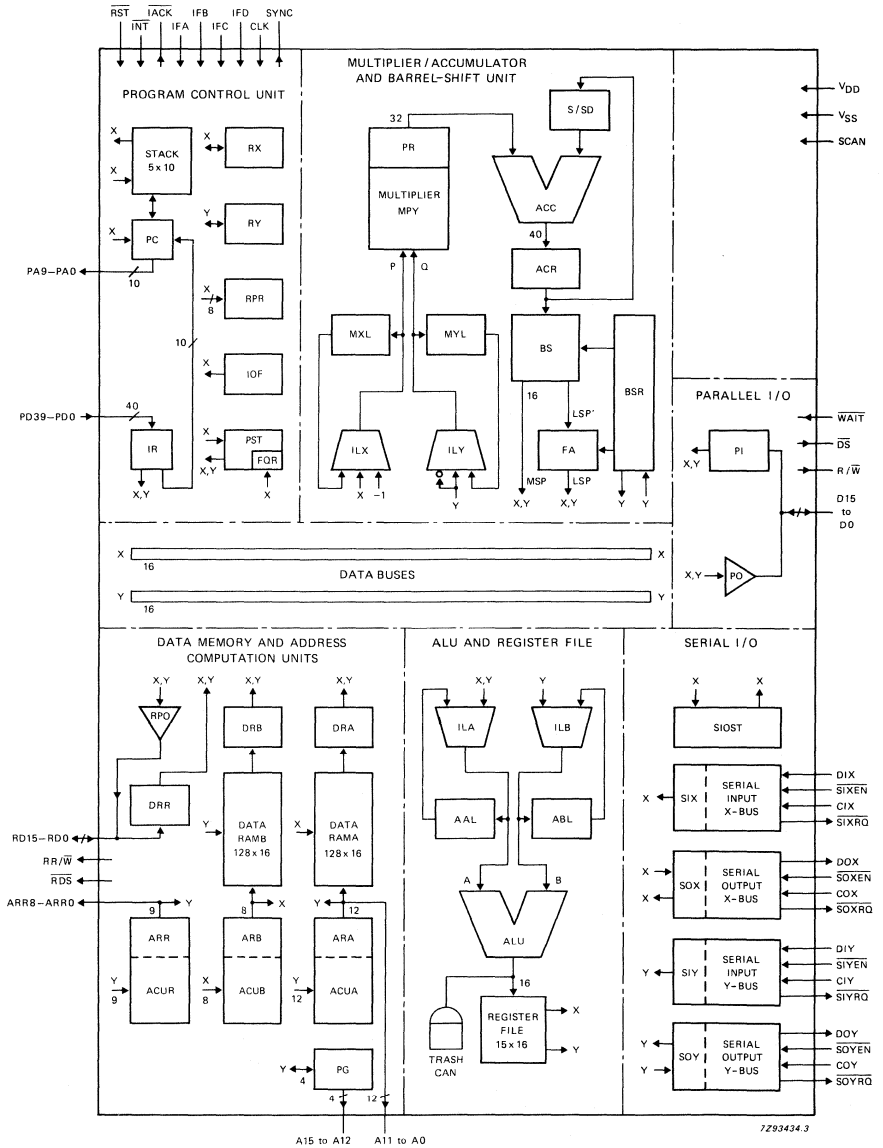


X,Y

Note 1: ↑ means connected to the X-bus/Y-bus.

Note 2: all data paths are 16 bits wide unless indicated otherwise.

Fig. 2.1-1 Detailed block diagram of the PCB5010.



X,Y

Note 1: ↑ means connected to the X-bus/Y-bus.

Note 2: all data paths are 16 bits wide unless indicated otherwise.

Fig. 2.1-2 Detailed block diagram of the PCB5011.

DEVELOPMENT DATA

2.2 PROGRAM CONTROL UNIT

2.2.1 Program memory

The PCB5011 has no on-chip program memory but you can connect external program memory. To access the external program memory, there are 40 program data pins (PD39-PD0) and 10 program address pins (PA9-PA0). The on-chip 10-bit program counter contents are available via these address pins.

The PCB5010 has 1K x 40-bit on-chip program memory:

- 987 x 40-bit mask programmable ROM (address 0-986)
- 32 x 40-bit static RAM (address 992-1023)
- 5 x 40-bit "load RAM" program in ROM (address 987-991)

The memory is addressed by the 10-bit on-chip program counter. The static RAM of the program memory can be loaded via the X-bus by MOVE or LOAD IMMEDIATE operations, following the procedure described in the section 2.2.2 "Load program RAM circuitry".

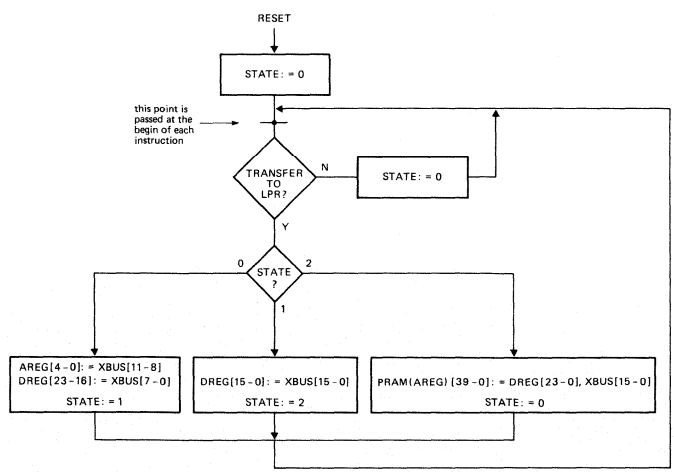
The "load RAM" program stored in the ROM can be used to load the RAM instructions from an external data memory. This program's code, and its equivalent expressed in the PCB5010/11 assembly language is given in Fig. 2.2.1-1.

IN PROGRESS

Fig. 2.2.1-1 "Load RAM" program.

2.2.2 Load program RAM circuitry (LPR)

LPR enables the programmer to load the 32 x 40-bit program RAM. To load each 40-bit instruction, three 16-bit words must be transferred to LPR via the X-bus, using MOVE or LOAD IMMEDIATE operations. LPR contains a 5-bit address register (AREG) in which the load address (range 0-31) is loaded and it contains a 24-bit data register (DREG) in which the instruction word is assembled. The loading procedure is shown in Fig. 2.2.2-1.



AREG = address register
DREG = data register
PRAM(i) = position i in the program RAM
[m – n]= bit m to bit n

Fig. 2.2.2-1 Data RAM loading procedure.

2.2.3 Program counter (PC) and mode circuitry (P and NP-mode)

The 10-bit program counter in the PCB5010 and PCB5011 is automatically incremented every instruction cycle during sequential program flow.

There are however, certain situations when the program counter is not incremented but updated differently:
- reset (see section 2.2.10)
- interrupt (see section 2.2.9)
- instruction repetition (see section 2.2.6)
- branch operations (see section 2.2.7)
- loading PC via the X-bus by means of a MOVE or LOAD IMMEDIATE operation (the 10 least significant bits of the X-bus data are loaded)

The processors work in two different modes:
1. Pipelined mode (P):
   instruction cycle = one clock cycle
2. Non-pipelined mode (NP):
   instruction cycle = two clock cycles

When the processor works in the P-mode, the result of a basic operation (see section 3.1 on instruction set) is not always available after the first instruction cycle but sometimes one clock cycle later. Since the processor uses pipelining, a new operation can start before the result of the previous operation is available. When the processor operates in NP-mode, the result of a basic operation is always available at the end of an instruction cycle. On reset, the processor is placed in the P-mode. Mode switching is possible under program control by setting and resetting the FQR bit in the PST register using a MOVE or LOAD IMMEDIATE operation. The instruction after the one that caused the change in the FQR bit is executed in the new mode.

2.2.4 Instruction register (IR)

In every instruction cycle (i.e. every clock cycle when working in the P-mode and every two clock cycles when working in the NP-mode) an instruction word is fetched from the program memory. The program memory access and storing of the result in the instruction register (IR) takes one clock cycle. During the next clock cycle, the new contents of the IR is decoded and the processor controlled accordingly.

2.2.5 Stack

When an interrupt or subroutine call occurs, the value of PC (in the P-mode) or the value of PC+1 (in the NP-mode) is placed on the stack. The stack is a 5 x 10-bit LIFO register file that allows automatic nesting up to five levels of subroutines and/or interrupts. The top of the stack containing the most recent PC value can be accessed via the data buses. This enables the programmer to extend the stack in the data memory.

2.2.6 Instruction repeat circuitry (RPR)

RPR is an 8-bit register that can be loaded via the X-bus by a MOVE or LOAD IMMEDIATE operation. Only the 8 LSBs on the bus (representing a number N between 0 and 255) are loaded. The instruction after the one in which these 8 bits are loaded is then executed N times as long as $2 \leqslant N \leqslant 255$. The execution count is undefined when $0 \leqslant N < 2$. The execution count is also undefined when the instruction to be repeated is to load RPR or to load the PC.

2.2.7 Branch circuitry

The PCB5010 and PCB5011 make it possible to depart from the sequential program flow under software control. There are 4 branch types, and each branch can depend on any one of 50 different conditions.

The 4 branch types are:
- go to
- subroutine call
- return from subroutine
- return from interrupt.

The 50 branch conditions are the true and false status of the following flags or combination of flags:
- ALU flags: Z, N, C, $\overline{\text{C.OR.Z}}$, V, VL, $\{\text{N.XOR.V}\}$.OR.Z, $\overline{\text{N.XOR.V}}$
- Accumulator flags: SGNM, OVFL
- Barrel-shifter flags: OOR, OORL
- ALU and barrel-shifter flags: OORL.OR.VL
- ACU flags: ACA, ACB, ACR
- User flags: IFA, IFB, IFC, IFD, IFA.AND.IFB.AND.IFC.AND.IFD
- Serial I/O flags: SIXACK, SIYACK, SOXACK, SOYACK

2.2.8 The PST and IOF registers (PST, IOF, IFA – IFD pins)

PST and IOF are 16-bit registers that contain all the flags. Furthermore, PST also contains a bit (EI) that indicates whether the interrupt is enable or not (enabled = 1; disabled = 0), a bit (FQR) indicating which mode (P = 0 or NP = 1) the processor is working in, and two bits (PIO1 and PIO2) determining the input criteria for the parallel input. The meaning of each bit of PST and IOF is given below:

bit	PST register		IOF register	
00	OVFL	(accumulator flag)	SIXACK	(serial I/O flag)
01	OORL	(barrel-shifter flag)	SOXACK	(serial I/O flag)
02	VL	(ALU flag)	SIYACK	(serial I/O flag)
03	V	(ALU flag)	SOYACK	(serial I/O flag)
04	C	(ALU flag)	IFA	(user flag)
05	Z	(ALU flag)	IFB	(user flag)
06	N	(ALU flag)	IFC	(user flag)
07	OOR	(barrel-shifter flag)	IFD	(user flag)
08	SGNM	(accumulator flag)	reserved	
09	ACA	(ACU flag)	reserved	
10	ACB	(ACU flag)	reserved	
11	ACR	(ACU flag)	reserved	
12	PIO2	(parallel I/O flag)	reserved	
13	PIO1	(parallel I/O flag)	reserved	
14	EI	(interrupt enable/disable)	reserved	
15	FQR	(operation mode)	reserved	

The flags in PST and IOF reflect the status of the functional units to which they belong and they are updated during each relevant instruction. The flags IFA, IFB, IFC and IFD reflect the signal level on their respective input pins. "Lock" type flags (OVFL, OORL and VL) can only be changed from 0 to 1 by the functional units to which they belong. The programmer can overrule the functional units updating the flags in the PST register: the PST register can be overwritten using a MOVE or LOAD IMMEDIATE operation. These operations are also used for loading the EI, FQR, PIO1 and PIO2 bits. Furthermore, the programmer can load the FQR bit using a MOVE or LOAD IMMEDIATE operation without changing the other bits in the PST register.

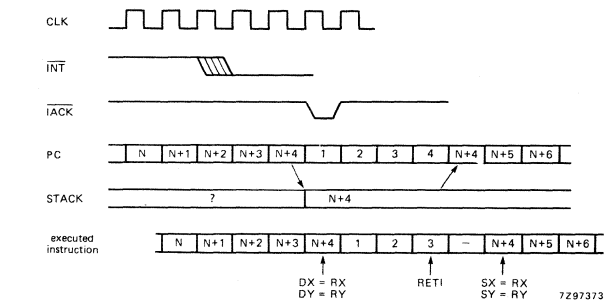
The PST and IOF register can be read using a MOVE operation. The flags can also be used as conditions in BRANCH operations (see section 2.2.7 on BRANCH circuitry). "Lock" type flags will automatically be reset to 0 when they are tested in a BRANCH operation.

2.2.9 Interrupt circuitry ($\overline{INT}$ and $\overline{IACK}$ pins, RX and RY)

The processor has an interrupt facility that the user can access via the $\overline{INT}$ and $\overline{IACK}$ pins. When an interrupt is accepted by the processor the program counter is loaded with address 1. The interrupt procedure is described below and the logic timing diagrams are given in Figures 2.2.9-1 and 2.2.9-2.

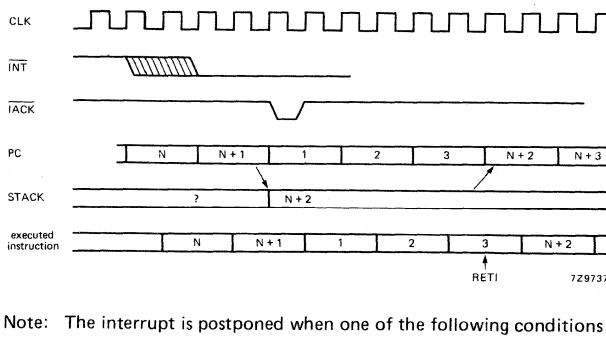
An interrupt is initiated by a LOW on the $\overline{INT}$ pin. The first positive going edge of CLK after a HIGH to LOW transition on $\overline{INT}$, the interrupt is clocked in by the processor. Two or three clock pulses later (see timing diagrams), the processor decides to accept the interrupt or postpone it. An interrupt is postponed:

- while the PC is being loaded (as part of a BRANCH, MOVE or LOAD IMMEDIATE operation)
- while the RPR register is loaded
- during instruction repetition
- when the stack is the source or destination to the X-bus
- when the interrupt is disabled by software (EI bit in PST register is 0).



Note: The interrupt is postponed when one of the following conditions is true during instruction N + 3:
- loading of PC (as part of BRANCH or MOVE operation)
- loading of RPR
- stack source or destination to X-bus
- repetition mechanism on
- interrupt disabled by EI = 0

Fig. 2.2.9-1 Interrupt logic timing in P-mode.



Note: The interrupt is postponed when one of the following conditions is true during instruction N + 1:
- loading of PC (as part of BRANCH or MOVE operation)
- loading of RPR
- stack source or destination to X-bus
- repetition mechanism on
- interrupt disabled by EI = 0

Fig. 2.2.9-2 Interrupt logic timing in NP-mode.

As soon as the above situations are completed, the postponed interrupt is accepted and thereafter handled in the same way as an interrupt that was accepted directly.

Accepting the interrupt means pushing the value of the PC (in P-mode) or PC+1 (NP-mode) on to the stack and loading the PC with address 1. When the PC contains address 1, the $\overline{IACK}$ pin goes LOW for one clock cycle to indicate to the outside world that the interrupt has been acknowledged.

In the P-mode, the instruction in the pipeline that should have been executed is not executed while the PC contains address 1 even though the expected X and Y-bus sources put their data on the buses. This data is stored in the bus-save register, RX and RY, that are automatically assigned as the destinations for the buses.

An interrupt routine is completed under program control using an RETI conditional BRANCH operation (see instruction set section 3.0). When the condition is true, PC is loaded with the address that was pushed onto the stack, and then the instruction at that address is executed. In the P-mode, however, RX and RY are used instead of the indicated X and Y-bus sources. Nested interrupts are permitted, but in the P-mode they are latched so long as the programmer has not stored the data in RX and RY elsewhere (i.e. so long as RX and RY have not been used as sources for the X and Y-buses). After returning from a nested interrupt in the P-mode, the old contents of RX and RY must be restored by the programmer.

Note: A new interrupt can onle be generated after the $\overline{INT}$ signal has been HIGH for at least one positive going edge on CLK.

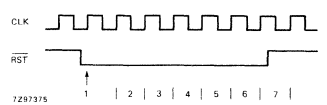
2.2.10 Reset circuitry ($\overline{RST}$ pin)

The processor is reset to an initial state when RST is LOW over at least 7 rising edges of CLK. A shorter reset may lead to an undefined situation.

The initial state is characterized by:

- PC : all zeros
- PST : all zeros
- IOF : SIXACK = SIYACK = 0 and SOXACK = SOYACK = 1
- RX, RY : all zeros
- STACK : all zeros (5 x 10)
- RPR : instruction repeat mechanism off
- LPR : undefined STATE; STATE = 0 in an instruction in which there is no MOVE or LOAD IMMEDIATE operation with PRAM as the destination
- MXL, MYL : all zeros
- PR : all zeros
- ACR : all zeros
- BSR : all zeros
- AAL, ABL : all zeros
- ARA, AA, SA : all zeros
- ARB, AB, SB : all zeros
- ARR, AR, SR : all zeros
- MA, MB, MR : undefined
- PG : all zeros
- SIOST : all zeros
- SOX, SOY : all zeros
- $\overline{SIXRQ}$ = $\overline{SIYRQ}$ = 0
- $\overline{SOXRQ}$ = $\overline{SOYRQ}$ = 1

The logic timing of the reset is shown in Fig. 2.2.10-1.



1 – reset detection
2 – reset of all registers and latches mentioned above (excluding ACR), and "top of stack", serial I/O request signals, and PR
3 – reset of ACR and 2nd stack position
4 – reset of 3rd stack position
5 – reset of 4th stack position
6 – reset of 5th stack position
7 – reset of 6th stack position (note: this stack position must not be used, otherwise it may lead to an undefined state).

Fig. 2.2.10-1
The complete reset action.

2.2.11 Synchronization circuitry (SYNC pin)

The processor indicates where execution of a new instruction starts with a HIGH at the SYNC pin for half a clock cycle. This occurs once every two clock cycles in the NP-mode, and once every clock cycle in the P-mode. However, in the P-mode it will not occur when the program counter is loaded with a new address during the execution of a BRANCH operation, a MOVE operation, an interrupt, or a reset.

2.3 DATA MEMORIES AND ACU'S

2.3.1 Data memories

The processor contains 3 on-chip data memories:

- RAMA: 128 x 16 bits, static
- RAMB: 128 x 16 bits, static
- ROM : 512 x 16 bits (only on PCB5010, external for PCB5011)

It is also possible to connect up to 64K of external data memory via the parallel I/O.

Memory outputs are connected to the data registers (DRA, DRB and DRR) and the parallel data input register PI. DRA, DRB and DRR are updated every instruction cycle, but in the P-mode, DRA and DRB are not updated when in that instruction cycle, data is moved (from one of the buses) into RAMA or RAMB. Updating PI is described in the section 2.6 on the parallel I/O. Data written into the data registers and PI can be transferred via the X or Y-bus during a subsequent instruction.

RAMA can be written-to via the X-bus, RAMB can be written-to via the Y-bus and, external RAM can be written-to via either bus.

With the PCB5011, it is not only possible to read from an external ROM, but in place of the read, it is also possible to write-to an external 512 x 16-bit memory because the port is bidirectional, and therefore can be used as a second parallel I/O port.

DEVELOPMENT DATA

2.3.2 ACUs and PG register

The 3 address computation units, ACUA, ACUB and ACUR, function identically but, their address widths differ: 12, 8 and 9 bits respectively. ACUs calculate the addresses for the on-chip RAMs (only the 7 least significant address bits are used for this addressing) and data ROM. ACUA not only generates the address for RAMA but also the part of the address for external data memory that defines the position within a page of 4096 x 16 words. It is possible to have 16 pages of external data memory. Pages are selected using the address pins A12 to A15 that reflect the contents of the page register PG that is loaded using a MOVE or LOAD IMMEDIATE operation.

Fig. 2.3.2-1 illustrates the memory addressing.

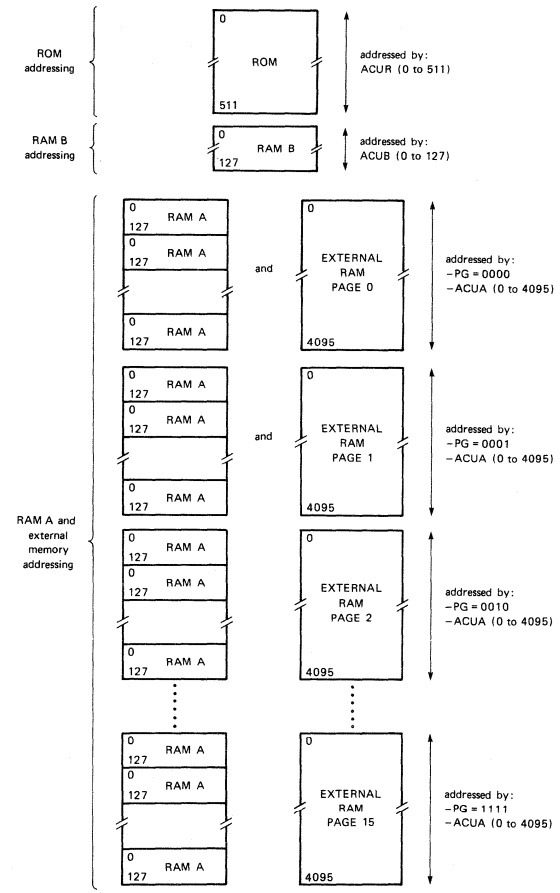


Fig. 2.3.2-1 Memory addressing.

An ACU contains:
- an address register AR (which addresses the ROM respectively RAM's)
- a base address register A
- an offset register S
- an address masking register M
- a dedicated arithmetic unit

Only the AR registers (ARR, ARB and ARA) are shown in Figs 2.1-1 and 2.1-2. The A, S and M registers of RAMA, RAMB and the ROM (AA/BA/RA, AS/BS/RS and AM/BM/RM) are not shown.
The A, S, M and AR registers can be loaded directly (for initialization) via one of the data buses (X-bus for ACUB registers and Y-bus for ACUA and ACUR registers) using a MOVE or LOAD IMMEDIATE operation or they can be modified during address computation operations. Direct loading and address computation cannot take place simultaneously (see the section 3.0 on the instruction set).

Direct loading offers the following options:
- Load AR with value on the bus
- Load AR and A with value on the bus
- Load AR and S with value on the bus
- Load A with value on the bus
- Load S with value on the bus
- Load M with value on the bus
- Load AR, A and S with value on the bus
- Load AR with (value on the bus) !M

The notation !M means that, depending on the contents of the M register, all bits are not necessarily updated as the expression indicates:
- the bits whose corresponding bits in the M register are 1 are updated as specified in the expression
- the bits whose corresponding bits in the M register are 0 will retain their value (for AR) or will receive the previous value of their AR bit (for A and S).

The notation br(. . .) means that the bits are reversed in order.

Three flags in the PST register indicate the status of the 3 ACU's:
- ACA – ACUA flag
  is 1 when AR contains 0000 0000 0000;
  is 0 when AR does not contain 0000 0000 0000
- ACB – ACUB flag
  is 1 when AR contains 0000 0000;
  is 0 when AR does not contain 0000 0000
- ACR – ACUR flag
  is 1 when AR contains 0 0000 0000;
  is 0 when AR does not contain 0 0000 0000

Address computation offers the following options:

• AR:=AR	A:=A	S:=S	M:=M
• AR:=(A+1)!M	A:=(A+1)!M	S:=S	M:=M
• AR:=(A−1)!M	A:=(A−1)!M	S:=S	M:=M
• AR:=(A+S)!M	A:=(A+S)!M	S:=S	M:=M
• AR:=(S+1)!M	A:=A	S:=(S+1)!M	M:=M
• AR:=(A)!M	A:=A	S:=S	M:=M
• AR:=(S)!M	A:=A	S:=S	M:=M
• AR:=br(A+S)	A:=A+S	S:=S	M:=M

2.4 MULTIPLIER, ACCUMULATOR, BARREL-SHIFTER AND FORMAT-ADJUSTER UNIT

2.4.1 Multiplier

The multiplier performs a multiplication of two signed 16 bits operands P and Q presented in 2's complement notation. The result is presented by 32 bits in 2's complement notation and stored in the product result latch PR.

One of the following values can be chosen as P-operand:
- The value present on the X-bus
- The previous value which was automatically latched in the MXL latch
- The number −1

One of the following values can be chosen as Q-operand:
- The value present on the Y-bus
- The previous value which was automatically latched in the MYL latch
- The negated Y-bus value
  Note: when the Y-bus contains the highest negative value, -2^{15} (1000 0000 0000 0000 in binary) then the operand will be the highest positive value plus one, $+2^{15}$ (1 0000 0000 0000 0000 in binary). This number is stored in MYL which has a width of 17 bits for this particular situation.

The contents of MXL, MYL and PR are not changed when no MULTIPLY operation or a multiply HOLD operation is executed.

2.4.2 Accumulator

The accumulator unit consists of a 40 bit adder ACC, a 40 bit multiplication/accumulation register ACR and a sign and scale down block S/SD. The adder adds the in PR stored result of the multiplication to a second operand (provided by the S/SD block) which can be chosen from the following set:

- + or − the contents of ACR
- + or − the contents of ACR divided by 2^{15} (which allows multiprecision multiplication and addition)
- the number 0

The result of the addition is stored in ACR and is fed simultaneously to the barrel-shifter. The contents of the ACR register are not changed when no MULTIPLY operation is performed or a multiply HOLD operation is executed.

The 40-bit width of the accumulator allows the programmer to accumulate a number of multiplier results (at least 256) without the risk of overflow. Two flags in the PST register indicate the status of the accumulator:

- OVFL − overflow lock flag; this flag is set when overflow occurs in the adder (result outside the range -2^{39} to $+2^{39}-1$). For reset conditions see description of PST register (section 2.2.8).
- SGNM − sign flag; indicates the sign of the result of the addition (is identical to bit ACR(39)).

2.4.3 Barrel-shifter

From the 40-bit ACR contents, the barrel-shifter extracts 32 contiguous bits. The programmer determines which group of 32 bits is extracted by a value placed in the BSR register (bits BSR3 to BSR0).
Sixteen different sets are possible:

BSR contents BSR3 - BSR0	32-bit word, E31 - E0, extracted by the barrel-shifter
0000	ACR30, ... ,ACR0,0
0001	ACR31, ... ,ACR0
0010	ACR32, ... ,ACR1
0011	ACR33, ... ,ACR2
0100	ACR34, ... ,ACR3
0101	ACR35, ... ,ACR4
0110	ACR36, ... ,ACR5
0111	ACR37, ... ,ACR6
1000	ACR38, ... ,ACR7
1001	ACR39, ... ,ACR8
1010	ACR39,ACR39, ... ,ACR9
1011	ACR39,ACR39,ACR39, ... ,ACR10
1100	ACR39,ACR39,ACR39,ACR39, ... ,ACR11
1111	ACR39,ACR39,ACR39,ACR39,ACR39, ... ,ACR12
1111	ACR39,ACR39,ACR39,ACR39,ACR39,ACR39, ... ,ACR13
1111	ACR39,ACR39,ACR39,ACR39,ACR39,ACR39,ACR39,. . . . ,ACR14

Two flags in the PST register indicate the status of the barrel-shifter:

- OOR − Out of range flag. It is set when the sign bit of the extracted word E31-E0 has no significance. This occurs when one or more bits of ACR to the left of the extracted word differs from E31.
- OORL − Out of range lock flag. The conditions for setting are identical to those of OOR. The conditions for resetting are given in the section describing the PST register.

2.4.4 Format adjuster

The output E31-E0 of the barrel-shifter is split into a 16-bit most significant part MSP and a 16-bit least significant part LSP'. MSP can be connected directly to the X and/or Y bus. LSP' passes through a format adjuster (FA) before it reaches the X or Y bus. The output of FA is called LSP. Under software control, three FA options can be selected by placing a value in the BSR register (bits BSR5 to BSR4). The options are:

BSR contents BSR5 - BSR4	output LSP of format adjuster
00	E15-E0 (no change)
01	E0-E15 (bits reversed in order, used to speed-up certain serial outputs)
10	0, E15-E1 (bits shifted right over 1 position, left adjusted with zero; used for multi-precision multiplications)
11	reversed/undefined

2.4.5 Barrel-shifter register (BSR)

BSR is a 6-bit register. Its contents control the barrel-shifter (bit 0-3) and the format adjuster (bit 4-5) as explained in the previous sections. BSR can be loaded by means of a MOVE or LOAD IMMEDIATE operation. Loading has to be done at least one instruction before LSP or MSP is read to the X or Y-bus.

2.5 ALU AND REGISTER FILE

2.5.1 ALU

The PCB5010/11 has an ALU totally independent from the multiplier/accumulator unit. It is a 16-bit, 2-operand unit capable of executing 31 distinct operations. There are arithmetic, logic and some special purpose operations. The arithmetic operations defined as "extended" (mnemonic starts with X) are included to facilitate multi-precision operations. Extended operands are represented by 16-bit multiples.

An ALU operation produces a result R that may be stored in the register file or may be ignored (dumped in the trash can).

Several flags in the PST register give the status of the ALU. The flags are:
Z – Zero flag
N – Negative flag
C – Carry flag
V – Overflow flag
VL – Overflow lock flag (same as V but locked; see PST register description, section 2.2.8).

One of the following values can be chosen as A-operand:
– the value on the X-bus
– the value on the Y-bus
– the previous value which was automatically latched in the AAL-latch (this is not the case with the "byte swap" instruction).

One of the following values can be chosen as B-operand:
– the value on the Y-bus
– the previous value which was automatically latched in the ABL-latch

Dyadic operations require an A and B-operand, while monadic operations require only an A-operand. Some operations do not require an operand at all.

The ALU operations and their result R and flag settings are summed up in the following three tables:
A: Arithmetic operations
B: Logic operations
C: Other operations

The following notation is used:

- ZERO(R) – 0 (when not all 16 bits of R are 0)
  – 1 (when all 16 bits of R are 0)
- CARRY(F) – 0 (when a function F does not lead to a carry)
  – 1 (when a function F leads to a carry)
  Note: For this calculation, the operands are unsigned 16-bit numbers from 0 to 65535. The carry is 1 when the result of the addition is greater than 0. In all other cases the carry is 0.
- BORROW(F) – 0 (when a function F does not lead to a borrow)
  – 1 (when a function F leads to a borrow)
  Note: For this calculation, the operands are unsigned 16-bit numbers from 0 to 65535. The borrow is 1 when the result of the subtraction is below 0. In all other cases the borrow is 0.
- OVERFLOW(F) – 0 (when a function F foes not lead to an overflow)
  – 1 (when a function F leads to an overflow)
  Note: For this calculation, the operands are signed 16-bit numbers between -2^{15} and $2^{15}-1$. The overflow is 1 when the result of the calculation is outside this range. In all other cases the overflow is 0.
- R(i) – Bit *i* of the 16-bit word R.

A: Arithmetic operations (see table on next page)

For the calculation of result R, the operands A and B are considered to be binary numbers in 2's complement notation (between -2^{15} and $+2^{15}-1$). R is also a binary number in 2's complement notation. However, with the DIV operation, the operands and the result are unsigned numbers (between 0 and 65535).

DEVELOPMENT DATA

A: Arithmetic operations

function	mnemonic	result (R)	flags				condition
			Z	N	C	V	
addition	ADD	A+B A+B−2^{16} A+B+2^{16}	ZERO(R) ZERO(R) ZERO(R)	R(15) R(15) R(15)	CARRY(A+B) CARRY(A+B) CARRY(A+B)	OVERFLOW(A+B) OVERFLOW(A+B) OVERFLOW(A+B)	no overflow positive overflow negative overflow
extended addition	XADD	A+B+C A+B+C−2(16) A+B+C+2(16)	ZERO(R).AND.Z ZERO(R).AND.Z ZERO(R).AND.Z	R(15) R(15) R(15)	CARRY(A+B+C) CARRY(A+B+C) CARRY(A+B+C)	OVERFLOW(A+B+C) OVERFLOW(A+B+C) OVERFLOW(A+B+C)	no overflow positive overflow negative overflow
subtraction	SUB	A−B A−B−2^{16} A−B+2^{16}	ZERO(R) ZERO(R) ZERO(R)	R(15) R(15) R(15)	BORROW(A−B) BORROW(A−B) BORROW(A−B)	OVERFLOW(A−B) OVERFLOW(A−B) OVERFLOW(A−B)	no overflow positive overflow negative overflow
extended subtraction	XSUB	A−B−C A−B−C−2^{16} A−B−C+2^{16}	ZERO(R).AND.Z ZERO(R).AND.Z ZERO(R).AND.Z	R(15) R(15) R(15)	BORROW(A−B−C) BORROW(A−B−C) BORROW(A−B−C)	OVERFLOW(A−B−C) OVERFLOW(A−B−C) OVERFLOW(A−B−C)	no overflow positive overflow negative overflow
conditional subtraction	CSUB	A	ZERO(R)	R(15)	0	0	If N=1, same as SUB
negate	NEG	0−A A	ZERO(R) ZERO(R)	R(15) R(15)	BORROW(0−A) BORROW(0−A)	OVERFLOW(0−A) OVERFLOW(0−A)	A ≠ -2^{15} A = -2^{15}
extended negate	XNEG	0−A−C A	ZERO(R).AND.Z ZERO(R).AND.Z	R(15) R(15)	BORROW(0−A−C) BORROW(0−A−C)	OVERFLOW(0−A−C) OVERFLOW(0−A−C)	A ≠ -2^{15}.OR.C = 1 A = -2^{15}.AND.C = 0
conditional negate	CNEG	A	ZERO(R)	R(15)	0	0	
decrement	DEC	A−1 $2^{15}-1$	ZERO(R) ZERO(R)	R(15) R(15)	BORROW(A−1) BORROW(A−1)	OVERFLOW(A−1) OVERFLOW(A−1)	A ≠ -2^{15} A = -2^{15}
extended decrement	XDEC	A−C $2^{15}-1$	ZERO(R).AND.Z ZERO(R).AND.Z	R(15) R(15)	BORROW(A−C) BORROW(A−C)	OVERFLOW(A−C) OVERFLOW(A−C)	A ≠ -2^{15}.OR.C = 0 A = -2^{15}.AND.C = 1
increment	INC	A+1 -2^{15}	ZERO(R) ZERO(R)	R(15) R(15)	CARRY(A+1) CARRY(A+1)	OVERFLOW(A+1) OVERFLOW(A+1)	A ≠ $2^{15}-1$ A = $2^{15}-1$
extended increment	XINC	A+C -2^{15}	ZERO(R).AND.Z ZERO(R).AND.Z	R(15) R(15)	CARRY(A+C) CARRY(A+C)	OVERFLOW(A+C) OVERFLOW(A+C)	A ≠ $2^{15}-1$.OR.C = 0 A = $2^{15}-1$.AND.C = 1
arithmetic shift left	ASL	2*A 2*A−2^{16} 2*A+2^{16}	ZERO(R) ZERO(R) ZERO(R)	R(15) R(15) R(15)	CARRY(A+A) CARRY(A+A) CARRY(A+A)	OVERFLOW(A+A) OVERFLOW(A+A) OVERFLOW(A+A)	$-2^{14} < A < 2^{14}$ A ⩾ 2^{14} A < -2^{14}
extended arithmetic shift left	XASL	2*A+C 2*A+C−2^{16} 2*A+C+2^{16}	ZERO(R).AND.Z ZERO(R).AND.Z ZERO(R).AND.Z	R(15) R(15) R(15)	CARRY(A+A) CARRY(A+A) CARRY(A+A)	OVERFLOW(A+A) OVERFLOW(A+A) OVERFLOW(A+A)	$-2^{14} < A < 2^{14}$ A ⩾ 2^{14} A < -2^{14}
arithmetic shift right	ASR	A/2-fraction	ZERO(R)	R(15)	1(0)	0	
extended arithmetic shift right	XASR	A/2-fraction A/2-fraction−2^{15} A/2-fraction+2^{15}	ZERO(R).AND.Z ZERO(R).AND.Z ZERO(R).AND.Z	N N N	A(0) A(0) A(0)	0 0 0	MSB of A = C MSB of A = 0.AND.C = 1 MSB of A = 1.AND.C = 0

A: Arithmetic operations (Cont'd)

function	mnemonic	result (R)	flags				condition
			Z	N	C	V	
add MSB of B to A	ADDM	A+B(15) A+B(15)−2^{16}	ZERO(R) ZERO(R)	R(15) R(15)	CARRY(A+B(15)) CARRY(A+B(15))	OVERFLOW(A+B(15)) OVERFLOW(A+B(15))	A+B(15) < 2^{15} A+B(15) = 2^{15}
unsigned division	DIV	2*(A−B) 2*A	ZERO(R) ZERO(R)	R(15) R(15)	$\overline{\text{BORROW(A−B)}}$ $\overline{\text{BORROW(A−B)}}$	0 0	0 ⩽ A−B < 2^{15} A−B < 0 or A−B > 2^{15}
sign extension	XSGN	NN . . . . N	Z.AND.NOT.N	N	N	V	

B: Logic operations

For the calculation of result R, the operands A and B are considered to be 16-bit binary words. R is also a 16-bit binary word.

function	mnemonic	result (R)	flags				condition
			Z	N	C	V	
compare	COM	.NOT.A(i)		R(15)	0	0	for i=0 . . . 15
logic AND	AND	A(i).AND.B(i)	ZERO(R)	R(15)	0	0	for i=0 . . . 15
logic OR	OR	A(i).OR.B(i)	ZERO(R)	R(15)	0	0	for i=0 . . . 15
exclusive OR	EXOR	A(i).EXOR.B(i)	ZERO(R)	R(15)	0	0	for i=0 . . . 15
byte swap	SWAP	A(i+8) A(i−8)	ZERO(R) ZERO(R)	R(15) R(15)	0 0	0 0	for i=0 . . . 7 for i=8 . . . 15
logic shift left	LSL	A(i−1) 0	ZERO(R) ZERO(R)	A(14) A(14)	A(15) A(15)	0 0	for i=1 . . . 15 for i=0
logic rotate left	LROL	A(i−1) C	ZERO(R) ZERO(R)	A(14) A(14)	A(15) A(15)	0 0	for i=1 . . . 15 for i=0
logic shift right	LSR	A(i+1) 0	ZERO(R) ZERO(R)	0 0	A(0) A(0)	0 0	for i=0 . . . 14 for i=5
logic rotate right	LROR	A(i+1) C	ZERO(R) ZERO(R)	0 0	A(0) A(0)	0 0	for i=0 . . . 14 for i=5

C: Other operations

function	mnemonic	result (R)	flags				condition
			Z	N	C	V	
pass and flag update	PASS	A	ZERO(A)	A(15)	0	0	
generate 0	NULL	0	1	0	0	0	
no operation	–	–	Z	N	C	V	

2.5.2 Register file

The output of the ALU is connected to the register file. This register file contains fifteen 16-bit registers. The programmer can choose to which register the ALU result is written and also has the option of discarding the result by writing it to the trash can.

The contents of any register may be read to either or both of the buses. Moreover, the register file is implemented as a 3-port memory, so that in the same instruction cycle three registers can be accessed: two accesses to read the present contents of register(s) and one to write in a new value.

The register file can be filled directly from the buses by a MOVE or LOAD IMMEDIATE operation. This does not affect the ALU flags. ALU operations cannot be specified simultaneously with the aforementioned MOVE operation.

2.6 PARALLEL I/O

Via the D15 – D0 pins, the PCB5010/11 permits parallel communication between the X or Y-bus and the outside world. An input ot output can take place during each instruction cycle. Output occurs when data is transferred via the X or Y-bus with destination PO. The output is direct, there is no latching. Inputs are loaded into the parallel input latch PI, whose contents can be transferred via the X or Y-bus during a subsequent instruction. The programmer can select one of three input criteria by writing specified values into PST bits, PIO1 and PIO2;

The following control signals are associated with the parallel I/O (see timing in section 4):

- R/$\overline{W}$: indicates a read or write action (output)
- $\overline{DS}$: data strobe (output)
- $\overline{WAIT}$: signals for synchronization of the parallel I/O (input). This signal delays the internal clock, so that "slow" peripheral devices may be connected. Note: that $\overline{WAIT}$ signal can also be used with the second parallel I/O port in the PCB5011 (see section 2.3.1).

A block diagram of the parallel I/O circuitry is shown in Fig. 2.6-1.

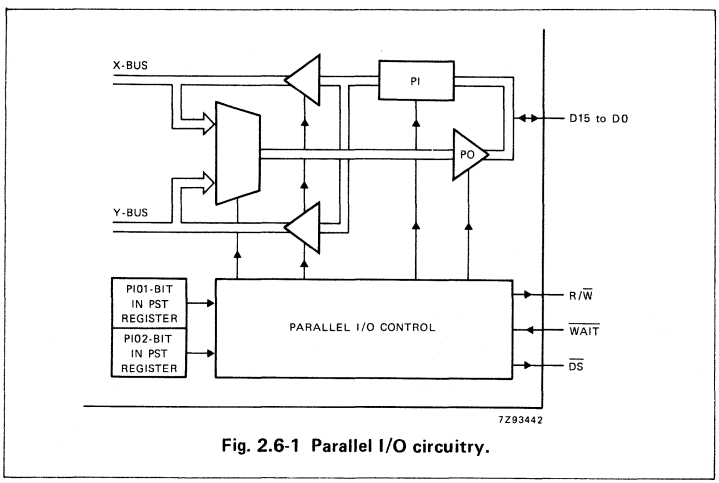


Fig. 2.6-1 Parallel I/O circuitry.

PIO1	PIO2	
0	0	A read action is performed each instruction cycle
0	1	A read action is performed only when: – ACUA is not executing a "no operation", or when – PI latch is source to X or Y-bus
1	0	A read action is performed only when: – PI latch is source to X or Y-bus
1	1	Reserved

Note: a read is only performed when there is no write (MOVE or LOAD IMMEDIATE operation to PO).

2.7 SERIAL I/O

The PCB5010/11 has 2 independent serial inputs DIX and DIY and 2 independent serial outputs DOX and DOY (destinations and sources for X and Y-bus respectively). Actual transfer occurs, with a maximum speed of 4 million bits/s, under the control of clocks CIX, COX, CIY, and COY from external devices.

The following handshake signals, described in more detail in the following sections, are associated with the serial I/O:
- input/output enable $\overline{\text{SIXEN}}$, $\overline{\text{SOXEN}}$, $\overline{\text{SIYEN}}$ and $\overline{\text{SOYEN}}$ (input signals)
- input/output request $\overline{\text{SIXRQ}}$, $\overline{\text{SOXRQ}}$, $\overline{\text{SIYRQ}}$ and $\overline{\text{SOYRQ}}$ (output signals)

The following flags, whose functions are also described in the following sections, are associated with the serial I/O:
- serial I/O acknowledge flags SIXACK, SOXACK, SIYACK and SOYACK.

The programmer may control the length of the words to be transferred (between 1 and 16 bits) by writing the applicable values in the SIOST register fields:

bit: 15 14 13 12	11 10 09 08	07 06 05 04	03 02 01 00
SOLY	SILY	SILX	SOLX

SILX = length of serial input word with destination X-bus
SOLX = length of serial output word with source X-bus
SILY = length of serial input word with destination Y-bus
SOLY = length of serial output word with source Y-bus

The binary number in each field specifies the length; the code 0000 indicates a length of 16 bits.

2.7.1 Serial input procedure

The serial input procedure, as illustrated in the flow diagram in Fig. 2.7.1-1, is described below. A detailed block diagram of the serial input circuitry (Fig. 2.7.1-2) is needed to understand the procedure.

Note: Only the serial input via the DIX pin is described since the serial input via the DIY pin operates in the same way.

1. At reset, the SIXACK flag is reset to '0' indicating that no word has been received in the SIX latch awaiting transfer via the X-bus.

2. The $\overline{\text{SIXRQ}}$ signal is reset to '0', indicating to external devices that they are allowed to send new data.

3. Nothing happens until the external device enables the serial input by setting the input signal $\overline{\text{SIXEN}}$ to '0'. This is checked only on the rising edge of the incoming clock signal CIX.

   $\overline{\text{SIXEN}}$ equal to '0', results in the shifting in of the first bit (LSB) available via the DIX pin. Then an internal counter COUNT is set to one less than the word-length. The word-length information is taken from the SILX field of the SIOST register.

4. At the next rising edge of the incoming clock signal CIX, the contents of COUNT is tested. When the contents of the counter is zero, data input is finished (continue at step 5). If it does not contain zero the data input continues as long as the $\overline{\text{SIXEN}}$ signal is '0': COUNT is decremented and a bit is shifted in via the DIX pin. Step 4 is then repeated. When the $\overline{\text{SIXEN}}$ signal is one, the data input is finished (aborted) and the procedure continues with step 5.

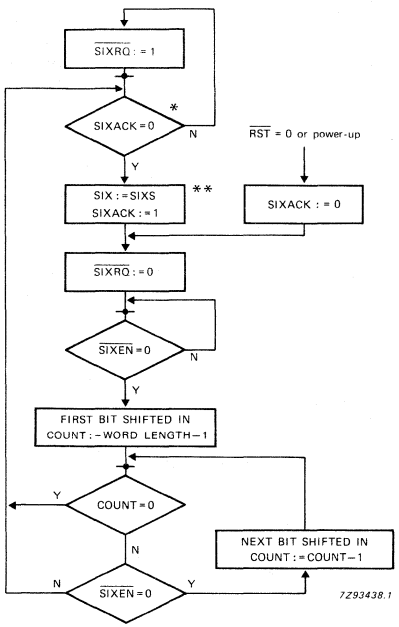
5. After finishing data input, the content of the shift register SIXS is copied to latch SIX. When less than 16 bits have been received the word is placed in the most significant part of SIX and the remaining bits are set to '0'. Also, the SIXACK flag is set to 1 indicating that a word has been received and is awaiting transfer via the X-bus.

   This copying and flag setting however does not occur when the SIXACK flag is not '0' (indicating that the previous input has not been transferred via the X-bus; note that the first time after reset or power up SIXACK is always 0). The $\overline{\text{SIXRQ}}$ signal is set to one, indicating to external devices that they are not allowed to send new data.

   After SIXACK has become '0' (this is tested only at rising edges of the CIX clock signal) the copying and flag setting finally takes place. The procedure continues with step 2.

Note: After changing the contents of the SILX and SILY fields of the SIOST register (which means: changing the word-length) the contents of SIX and SIY are undefined.

DEVELOPMENT DATA



* SIXACK goes to 0 when SIX is the source on the X-bus.
** When less than 16-bits have been received, the word is placed in the most significant part of SIX and the remaining bits are set to 0.
◆ This point is only passed when a rising edge on CIX occurs.

Fig. 2.7.1-1 Serial input procedure.

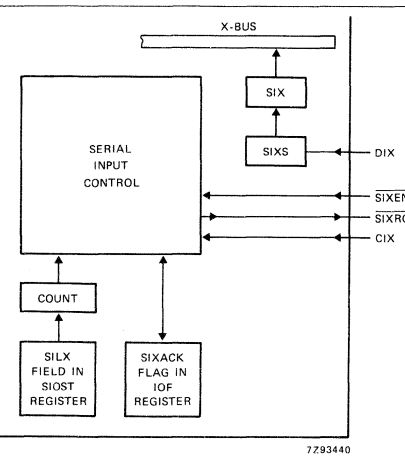


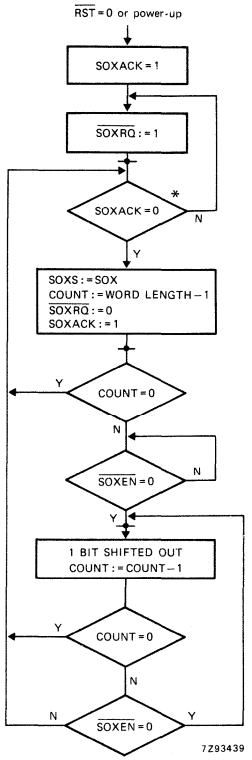
Fig. 2.7.1-2 Serial input circuitry.

2.7.2 Serial output procedures

The serial output procedure, as illustrated in the flow diagram in Fig. 2.7.2-1 is described below. A detailed block diagram of the serial output circuitry (Fig. 2.7.2-2) is needed to understand the procedure.

Note: Only the serial output via the DOX pin is described; the serial output via the DOY pin operates in the same way.

1. At reset, the SOXACK flag is set to '1', indicating that a word may be transferred via the X-bus to the SOX latch.
2. The $\overline{\text{SOXRQ}}$ output signal is set to '1' indicating to the external devices that there is no data available to be clocked out. The procedure is suspended until the first falling edge of incoming clock signal COX.
3. When the SOXACK flag is not '0' (then no data has been transferred via the X-bus to the SOX latch), step 2 is repeated. When SOXACK is '0' (data has been transferred via the X-bus to the SOX latch), the data in the SOX latch is copied to the SOXS output shift register. The first bit (LSB) of the word that must be shifted out is placed in front of the output 3-state driver.
   Then the $\overline{\text{SOXRQ}}$ output signal is set to '0' indicating to the external devices that data is available to be clocked out.
   Also, the SOXACK flag is set to one to indicate that a new data word may be transferred via the X-bus to the SOX latch. Finally, an internal counter is loaded with a value one less than the word-length. The word-length information is taken from the SOXL field of the SIOST register.
   The procedure is now suspended until the next falling edge of the incoming clock signal COX.
4. When the word-length is 1 (then the count is 0), the output procedure is finished. One bit words can be read by external devices that set the $\overline{\text{SOXEN}}$ signal to '0' which enables the 3-state output buffer. In this case, the procedure continues with step 3. It is important to note that SOXS can be overwritten before the bit has been read by the external devices.
   When the word-length is greater than 1 (then count is not equal to zero) the output procedure is not finished.
   The procedure is suspended until one of the external devices enables the serial output by setting the input signal $\overline{\text{SOXEN}}$ to '0'. The first bit is available on the DOX pin immediately after the serial output is enabled.
5. Nothing happens until a falling edge on the clock input COX is generated. At that moment another bit in the SOXS shift-register is shifted out via DOX. In addition, the contents of the counter is decremented. The output is finished when the counter contents has been decremented to zero (continue at step 3).
   If after decrementing, the counter value is not zero, the output continues by repeating step 5 unless $\overline{\text{SOXEN}}$ is now '1'. When $\overline{\text{SOXEN}}$ = 1, the output is finished (aborted) and the procedure continues at step 3.



* SOXACK goes to 0 when SOX is the destination on the X-bus.

• This point is only passed when a falling edge on COX occurs.

Note: DOX is not enabled (in the 3-state mode) while $\overline{SOXEN}$ = 1.

Fig. 2.7.2-1 Serial output procedure.

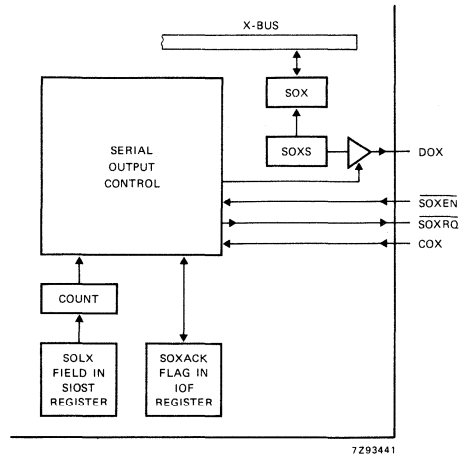


Fig. 2.7.2-2 Serial output circuitry.

2.8 DATA BUSES X AND Y

The data buses have a width of 16-bits and are used for transferring data between the functional units connected to them. Bus transfers are always part of MOVE and LOAD IMMEDIATE operations and can be part op MPY and ALU operations. The possible sources and destinations of the bus are given in the table below.

source on X	destination on X	source on Y	destination on Y
STACK	STACK		
RX	RX	RY	RY
	PC		
	RPR		
IOF			
IR(DATA field)		IR(DATA field)	
PST	PST	PST	
	FQR		
DRA DRB DRR		DRA DRB DRR	
	RAMA		RAMB
ARB		ARA ARR	
	ACUB		ACUA ACUR
		PG	PG
	ILX		ILY
LSP MSP		LSP MSP	
		BSR	BSR
RFILE	RFILE	RFILE	RFILE
	ILA		ILA ILB
PI	PO	PI	PO
SIOST	SIOST		
SIX SOX	SOX	SIY SOY	SOY

Note 1: FQR is loaded by putting the required value (0 or 1) in bit 13 of the 16-bit word that is sent. The other bits are 'don't cares'.

Note 2: When the source or destination has a width of less than 16 bits, only the least significant bits on the bus (same width) contain information.

Note 3: ILX and ILY select the operands for the multiplier; ILA and ILB select the operands for the ALU.

3.0 INSTRUCTION SET

The behaviour of the processor is controlled by the instructions stored in on-chip ROM (PCB5010) or external program memory (PCB5011). Each instruction leads to the execution of one or more (up to 6) basic operations. The basic operations, the instruction format and the instruction fields are described in the following sections.

3.1 BASIC OPERATIONS

The execution of a basic operation can take either 1 or 2 clock cycles. This means that:

- for 2 cycle basic operations pipelining is used in the P-mode
- a 1 cycle operation is extended by a second "no action" cycle when the processor is operating in the NP-mode.

Basic operation sequencing is shown in Fig. 3.1-1 for the P-mode, and Fig. 3.1-2 for the NP-mode. There are 6 different basic operations which are described below.

- **ALU operation** (1 clock cycle)

  One of 31 different ALU operations is executed. The operations need 2, 1 or 0 operands. The operation's result is stored in the register file or thrown away (into the trash can). In addition, a number of flags are updated.

- **MOVE operation** (1 clock cycle)

  Data from an X-bus (Y-bus) source is transferred via the X-bus (Y-bus) to an X-bus (Y-bus) destination. In the case that both buses have the same destination no transfer takes place.

- **ACU (address computation and memory access) operation** (2 clock cycles)

  Clock cycle 1:
  An address is calculated by the ACU and written into an address register (ARA, ARB, or ARR).

  Clock cycle 2:
  RAMA, RAMB, ROM or external memory is accessed at the location given by the address register (ARA, ARB, or ARR). This access is normally a read which updates one of the data registers (DRA, DRB, DRR, or PI). If in the P-mode, however, the relevant RAM (RAMA, RAMB or external RAM) is assigned as a destination during a MOVE executed in parallel, the access will be a write. The result in this case is that the RAM is updated.

- **MPY (multiply/accumulate) operation** (2 clock cycles)

  Clock cycle 1:
  Two operands are multiplied together and the result stored in the PR latch.

  Clock cycle 2:
  The content of PR is added to the output of S/SD and stored in the ACR.

  Note 1: The content of the ACR is accessed via the barrel-shifter and format-adjuster so that it can be transferred on the X or Y-bus during the clock cycle following ACR loading. The buses are selected by specifying LSP and/or MSP as the source for the X or Y-bus.

  Note 2: The explanation of the MPY operation is a simplified presentation of what really happens. In fact, clock cycle 1 is not just used for multiplication, but part of the accumulation as well: i.e. accumulation of the lower part of the multiplier result and the contents of the accumulator. Therefore, in the P-mode, an MPY with +ACRS or −ACRS in the MPY field (see section 3.3 on description of MPY field) that directly follows another MPY operation leads to an undefined situation where the higher part of the first multiplication/accumulation result is already required at the end on clock cycle 1, but is only ready after clock cycle 2. If it is necessary to have an MPY operation with +ACRS or −ACRS directly after a previous MPY operation (for multiprecision multiplication), then there has to be a no operation cycle (or at least no MPY operation) between them.

- **BRANCH operation** (1 or 2 clock cycles)

  Clock cycle 1:
  The branch condition is checked, the result of which can be true or false. When the result is false, the BRANCH operation is terminated. When the result is true, the action specified under "Clock cycle 2" is executed.

  Clock cycle 2:
  The program counter is loaded with the branch address.

  Note: no new operation is started during loading.

- **LOAD IMMEDIATE operation** (1 clock cycle)

  The 16-bit data word specified in the instruction is put on the X-bus and the Y-bus and transferred to one or two specified destinations. In the case that two identical destinations are specified no transfer takes place.

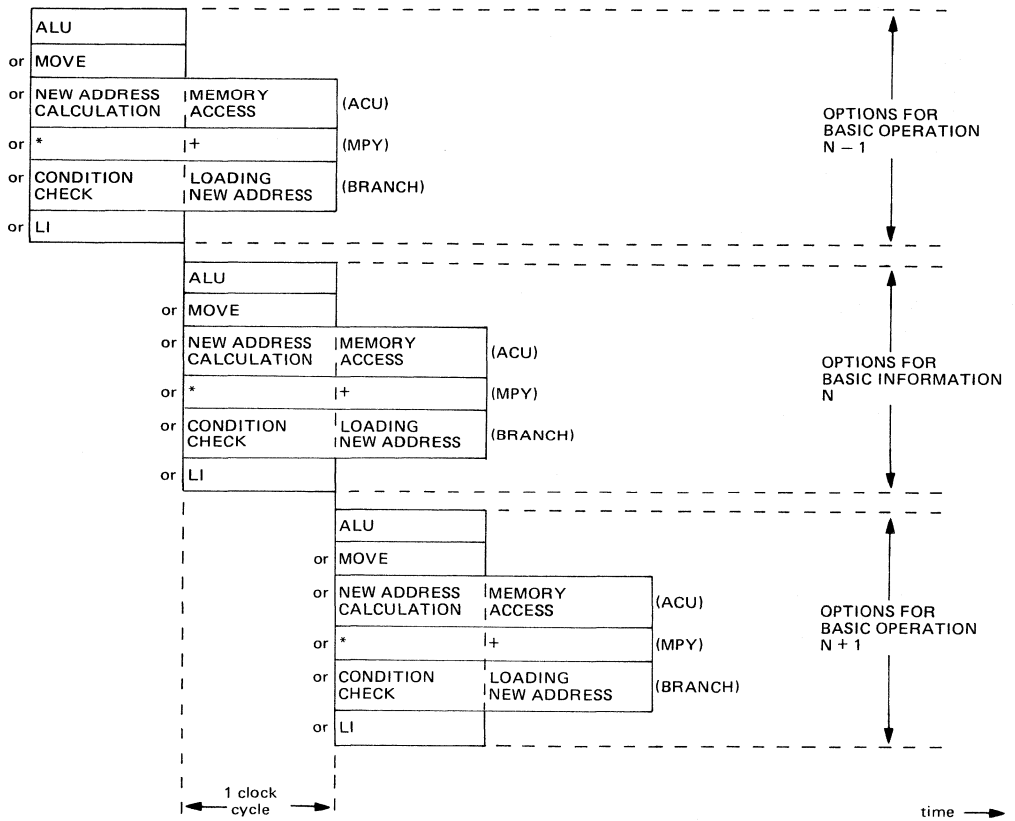


Fig. 3.1-1 Basic operation sequencing in P-mode.

Notes to Fig. 3.1-1

NOTE 1: Each instruction may contain several simultaneous basic operations (see section 3.2 on instruction formats).

NOTE 2: "Loading new address" (BRANCH) takes place only when the condition is true.

NOTE 3: During "loading new address" (BRANCH) no new operation is started.

NOTE 4: Memory access is normally a data register (DRA, DRB, DRR, or PI) read. This read doesn't take place when there is a simultaneous MOVE to a RAM (during a RAM access).

NOTE 5: An MPY operation with +ACRS or –ACRS in the MPY field directly following another MPY operation leads to an undefined situation (see description of MPY operation).

NOTE 6: After instruction N which specifies a MOVE or LOAD IMMEDIATE operation to PC (with DATA m) first instruction N + 1 is exceeded before instruction m is executed.

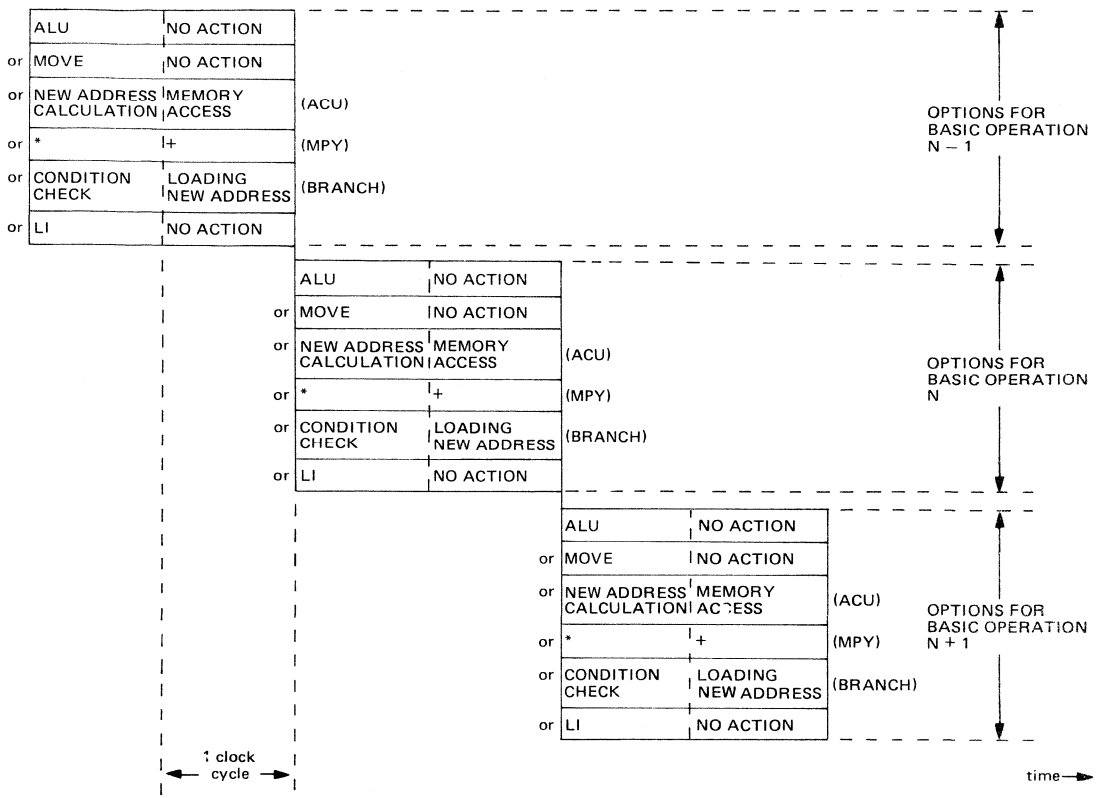


Fig. 3.1-2 Basic operation sequencing in NP-mode.

Notes to Fig. 3.1-2

NOTE 1: Each instruction may contain several simultaneous basic operations (see section 3.2 on instruction format).

NOTE 2: "Loading new address" (BRANCH) takes place only when the condition is true. If the condition is false, there is no action during the second clock cycle.

NOTE 3: Memory access is a data register read.

DEVELOPMENT DATA

3.2 INSTRUCTION FORMAT

All instruction words are 40-bits wide.

4 types of instructions are defined, each with their own set of basic operations:

Type 0: ALU operation + 2 MOVE operations + 3 ACU operations

Type 1: MULTIPLY operation + 2 MOVE operations + 3 ACU operations

Type 2: BRANCH operation + 3 ACU operations

Type 3: LOAD IMMEDIATE operation + 3 ACU operations.

Each basic operation is specified by the contents of one or more instruction fields (see Fig. 3.2-1).

The field at the far left of each instruction indicates the instruction type. The other fields are defined in the following table.

ALU	= Type of ALU operation	dedicated for ALU operations
AOPS	= ALU operands	dedicated for ALU operations
SX	= Source on X-bus	These fields are for ALU and/or MOVE operations,
SY	= Source on Y-bus	or for MPY and/or MOVE operations
DX	= Destination on X-bus	for MOVE or LI operations
DY	= Destination on Y-bus	for MOVE or LI operations
RFILE	= Destination in Register file	for ALU, MOVE or LI operations
ACUA	= Type of ACUA operation	dedicated for ACU operations
ACUB	= Type of ACUB operation	dedicated for ACU operations
ACUR	= Type of ACUR operation	dedicated for ACU operations
MPY	= Type of accumulator operation	dedicated for MPY operations
MOPS	= Multiply operands	dedicated for MPY operations
NAP	= Address of next instruction when condition is true	dedicated for BRANCH operations
BR	= Type of branch operation	dedicated for BRANCH operations
COND	= Branch condition	dedicated for BRANCH operations
DATA	= 16 bits data word that is transmitted on X and Y-bus	dedicated for LI operations

3.3 INSTRUCTION FIELDS

For each instruction field, the valid codes and their function are specified. For most fields, not only the function is specified but also a mnemonic. The PCB5010/11 assembly language, using the mnemonics in this data sheet, is described in a separate document.

Instruction type 0:

39	38	37 36 35 34 33	32 31	30 29 28 27 26	25 24 23 22 21	20 19 18 17	16 15 14 13	12 11 10 9	8 7 6	5 4 3	2 1 0
0	0	ALU	AOPS	SX	SY	DX	DY	RFILE	ACUA	ACUR	ACUB

Instruction type 1:

39	38	37 36 35	34 33 32 31	30 29 28 27 26	25 24 23 22 21	20 19 18 17	16 15 14 13	12 11 10 9	8 7 6	5 4 3	2 1 0
0	1	MPY	MOPS	SX	SY	DX	DY	RFILE	ACUA	ACUR	ACUB

Instruction type 2:

39	38	37 36 35 34 33 32 31 30 29 28 27 26 25 24 23 22	21 20 19	18 17 16 15 14 13	12 11 10 9	8 7 6	5 4 3	2 1 0
1	0	NAP	BR	CONDITION	– – – –	ACUA	ACUR	ACUB

Instruction type 3:

39	38	37	36 35 34 33 32 31 30 29 28 27 26 25 24 23 22 21	20 19 18 17	16 15 14 13	12 11 10 9	8 7 6	5 4 3	2 1 0
1	1	–	DATA	DX	DY	RFILE	ACUA	ACUR	ACUB

Fig. 3.2-1 The four instruction types and their fields.

ALU

mnemonic	code	type	function
Arithmetic operations:			
ADD	10100	dyadic	addition
XADD	10101	dyadic	extended addition
SUB	10110	dyadic	subtraction
XSUB	10111	dyadic	extended subtraction
CSUB	11000	dyadic	conditional subtraction
NEG	01100	monadic	negate
XNEG	01101	monadic	extended negate
CNEG	01110	monadic	conditional negate
DEC	01010	monadic	decrement
XDEC	01011	monadic	extended decrement
INC	01000	monadic	increment
XINC	01001	monadic	extended increment
ASL	00111	monadic	arithmetic shift left
XASL	00110	monadic	extended arithmetic shift left
ASR	00011	monadic	arithmetic shift right
XASR	00000	monadic	extended arithmetic shift right
ADDM	11010	dyadic	add MSB of B to A
DIV	11001	dyadic	unsigned division step
XSGN	11110	no operand	extended N flag
Logic operations:			
COM	10000	monadic	logic complement
AND	10001	dyadic	logic AND
OR	10010	dyadic	logic OR
EXOR	10011	dyadic	logic exclusive OR
SWAP	11100	monadic	byte swap
LSL	00101	monadic	logic shift left
LROL	00100	monadic	logic rotate left
LSR	00001	monadic	logic shift right
LROR	00010	monadic	logic rotate right
Other operations:			
PASS	01111	monadic	pass with flag update
NULL	11101	no operand	generate 0
–	11111	–	no operation
–	11011	–	reserved

AOPS

mnemonic	code	AINS-type	A-operand	B-operand
*, *	00	dyadic	source on X-bus	source on Y-bus
AAL, *	01	dyadic	AAL	source on Y-bus
*, ABL	10	dyadic	source on X-bus	ABL
AAL, ABL	11	dyadic	AAL	ABL
AAL	00	monadic	AAL	–
(the operation is undefined when this operand is selected with a SWAP operation)				
*	01	monadic	source on X-bus	–
*	10	monadic	source on Y-bus	–
–	11	–	reserved	–

Note: the AOPS field has no meaning when a no operand operations is specified in the ALU field and as there is no operation, AAL and ABL retain their values.

* The mnemonic of the particular source is used (see SX and SY fields).

SX

mnemonic	code	full name of source
–	00000	no source
ROM	00001	data register DRR
–	00010	reserved
TOS	00011	top of stack (Note 1)
RX	00100	bussave register X
PI	00101	parallel data input register
IOF	00110	input/output status and user flag register
SOX	00111	serial output register connected to X-bus
SIX	01000	serial input register connected to X-bus
SIOST	01001	serial I/O control register
–	01010	reserved
RAMB	01011	data register DRB
PST	01100	processor status register
RAMA	01101	data register DRA
ACU(RAMB)	01110	address register ARB
LSP	01111	least significant 16 bits of multiply/shift/adjust result
MSP	10000	most significant 16 bits of multiply/shift/adjust result
R1	10001	register 1
R2	10010	register 2
R3	10011	register 3
R4	10100	register 4
R5	10101	register 5
R6	10110	register 6
R7	10111	register 7
R8	11000	register 8
R9	11001	register 9
R10	11010	register 10
R11	11011	register 11
R12	11100	register 12
R13	11101	register 13
R14	11110	register 14
R15	11111	register 15

Note 1: When TOS is used as a source, the stack is popped one level.

SY

mnemonic	code	full name of source
–	00000	no source
RAMB	00001	data register DRB
–	00010	reserved
RAMA	00011	data register DRA
RY	00100	bussave register Y
ROM	00101	data register DRR
SIY	00110	serial input register connected to Y-bus
–	00111	reserved
PG	01000	page register
BSR	01001	barrel-shifter/format adjuster control register
ACU(ROM)	01010	address register ARR
ACU(RAMA)	01011	address register ARA
PST	01100	processor status register
PI	01101	parallel data input register
SOY	01110	serial output register connected to Y-bus
LSP	01111	least significant 16 bits of multiply/shift/adjust result
MSP	10000	most significant 16 bits of multiply/shift/adjsut result
R1	10001	register 1
R2	10010	register 2
R3	10011	register 3
R4	10100	register 4
R5	10101	register 5
R6	10110	register 6
R7	10111	register 7
R8	11000	register 8
R9	11001	register 9
R10	11010	register 10
R11	11011	register 11
R12	11110	register 12
R13	11101	register 13
R14	11110	register 14
R15	11111	register 15

DX

mnemonic	code	full name of destination
–	0000	no destination
FQR	0001	FQR bit in PST; the required value for FQR must be present in bit 15 on the bus; the other bits are don't cares
SIOST	0010	serial I/O control register
RPO	0011	second parallel data output buffer (PCB5011 only)
RX	0100	bussave register X
PC	0101	program counter
SOX	0110	serial output register connected to X-bus
PRAM	0111	program RAM
–	1000	reserved
RPR	1001	instruction repeat register
PO	1010	parallel data output buffer
RAMA	1011	RAMA
PST	1100	processor status register
TOS	1101	top of stack (Note 1)
ACU(RAMB,*)	1110	ACUB
**	1111	register file

* fill in a mnemonic of the ACU-initialization field.
** fill in a mnemonic of the RFILE field.

Note 1: When TOS is used as a destination, the stack is pushed one level.

DY

mnemonic	code	full name of destination
–	0000	no destination
BSR	0001	barrel-shifter/format adjuster control register
–	0010	reserved
RAMB	0011	RAMB
–	0100	reserved
RY	0101	bussave register Y
–	0110	reserved
–	0111	reserved
–	1000	reserved
PG	1001	page register
ACU(RAMA,*)	1010	ACUA
PO	1011	parallel data output buffer
RPO	1100	second parallel data output buffer (PCB5011 only)
SOY	1101	serial output register connected to Y-bus
ACU(ROM,*)	1110	ACUR
**	1111	register file

* fill in a mnemonic of the ACU-initialization field.
** fill in a mnemonic of the RFILE field.

RFILE

mnemonic	code	full name of destination
–	0000	no destination
R1	0001	register 1
R2	0010	register 2
R3	0011	register 3
R4	0100	register 4
R5	0101	register 5
R6	0110	register 6
R7	0111	register 7
R8	1000	register 8
R9	1001	register 9
R10	1010	register 10
R11	1011	register 11
R12	1100	register 12
R13	1101	register 13
R14	1110	register 14
R15	1111	register 15

ACU

The ACU fields for RAMA, RAMB and ROM are identical. The content of the field has a different meaning if it is for ACU initialization or address computation.

ACU INITIALIZATION

mnemonic	code	New values for:			
		AR	A	S	M
AR	000	source	A	S	M
AAR	001	source	source	S	M
SAR	010	source	A	source	M
A	011	AR	source	S	M
S	100	AR	A	source	M
M	101	AR	A	S	source
ASAR	110	source	source	source	M
AR1M	111	(source)!M	A	S	M

ADDRESS COMPUTATION

mnemonic	code	New values for:			
		AR	A	S	M
	000	AR	A	S	M
INCA	001	(A+1)!M	(A+1)!M	S	M
DECA	010	(A–1)!M	(A–1)!M	S	M
STEP	011	(A+S)!M	(A+S)!M	S	M
INCS	100	(S+1)!M	A	(S+1)!M	M
A	101	(A)!M	A	S	M
S	110	(S)!M	A	S	M
REV	111	br(A+S)	A+S	S	M

Note: see section 2.3.2 on ACUs and PG register for explanation of !M and br(. . . .).

MPY

mnemonic	code	new ACR-value
–	000	ACR (HOLD)
0	001	P*Q
+ACR	010	P*Q + ACR
–ACR	011	P*Q – ACR
+ACRS	100	P*Q + ACR x 2^{-15}
–ACRS	101	P*Q – ACR x 2^{-15}

MOPS

mnemonic	code	P-input of multiplier	Q-input of multiplier
*, *	0000	source on X-bus	source on Y-bus
, –	0001	source on X-bus	–(source on Y-bus)
MXL, *	0010	MXL	source on Y-bus
MXL, –*	0011	MXL	–(source on Y-bus)
*, MYL	0100	source on X-bus	MYL
MXL, MYL	0101	MXL	MYL
–1, –*	0110	–1	–(source on Y-bus)
–1, *	0111	–1	source on Y-bus
–1, MYL	1000	–1	MYL

Note: When the MPY-field contains 000, then independent of the MOPS-code the following is true:

mnemonic	code	P-input of multiplier	Q-input of multiplier
–	XXXX	MXL	MYL

* Fill in the mnemonic of the particular source (see SX and SY fields).

NAP

The NAP field contains the address of the next instruction to be executed if a branch condition is true.

BR

mnemonic	code	function
GOTO	000	goto
CALL	001	subroutine call
RET	010	return from subroutine
RETI	100	return from $\overline{\text{INT}}$ interrupt

DEVELOPMENT DATA

COND

mnemonic	code	condition	mnemonic explanation
–	00000X	always true	
AN	00001X	SGNM	accumulator negative
XO	00010X	OOR	
–	00011X	reserved	
GE or NOT LT	00100X	$\overline{\text{N.EXOR.V}}$	greater or equal to/ not less than
NOT GT or LE	00101X	(N.EXOR.V).OR.Z	not greater than/less than or equal to
IFX	00110X	IFA.AND.IFB.AND.IFC.AND.IFD.	
HI or NOT LS	00111X	$\overline{\text{C.OR.Z}}$	higher/not less than or equal to
OFL	01000X	OORL.OR.VL	system overflow
XOL	01001X	OORL	extractor overflow
AOL	01010X	OVFL	accumulator overflow
VL	01011X	VL	
Z or EQ	01100X	Z	
N	01101X	N	
C	01110X	C	
V	01111X	V	
SIX	10000X	SIXACK	
SOX	10001X	SOXACK	
SIY	10010X	SIYACK	
SOY	10011X	SOYACK	
ACU(RAMA)	10100X	ACA	
ACU(RAMB)	10101X	ACB	
ACU(ROM)	10110X	ACR	
–	10111X	reserved	
IFA	11000X	IFA	
IFB	11001X	IFB	
IFC	11010X	IFC	
IFD	11011X	IFD	
–	11100X	reserved	
–	11101X	reserved	
–	11110X	reserved	
–	11111X	reserved	

X = 0: branch takes place when condition = true
X = 1: branch takes place when condition = false

Note: the mnemonic names GE, LT, GT, LE, HI, and LS refer to situations where the flag setting is a result of a subtraction (SUB) of operands A and B. For example, GE means that A is greater than or equal to B.

DATA

Data is a 16 bit data word which is transmitted on the X and Y-bus.

4.0 ELECTRICAL SPECIFICATION

4.1 ABSOLUTE MAXMIMUM RATINGS

symbol	parameter	condition	min.	max.	unit
V_{DD}	supply voltage		−0.3	+7	V
V_I	voltage at any input		−0.3	+7	V
T_{amb}	operating ambient temperature range	PCB5010/11 PCF5010/11	0 −40	+70 +85	°C °C
T_{stg}	storage temperature range		−55	+150	°C

4.2 DC CHARACTERISTICS

symbol	parameter	condition	min.	typ.	max.	unit
V_{DD}	supply voltage		4.75	5.0	5.25	V
V_{IH}	HIGH level input voltage		2	–	V_{DD}+0.3	V
V_{IL}	LOW level input voltage		−0.3	–	+0.8	V
V_{OH}	HIGH level output voltage	V_{DD} = 4.75 V I_{OH} = 100 μA	2.4	–	–	V
V_{OL}	LOW level output voltage	V_{DD} = 4.75 V I_{OL} = 2 mA	–	–	0.4	V
I_{OH}	HIGH level output current		–	–	100	μA
I_{OL}	LOW level output current		–	–	2	mA
C_I	input pin capacitance		–	–	tbf	pF
C_O	output pin capacitance		–	–	tbf	pF
P	power dissipation		–	–	tbf	mW

DEVELOPMENT DATA

4.3 AC CHARACTERISTICS

no.	parameter	min.	max.	unit	note
1	CLK width HIGH	60	1000	ns	
2	CLK width LOW	65	1000	ns	
3	CLK HIGH to SYNC HIGH	10	50	ns	
4	CLK HIGH to Ax valid	10	60	ns	x = 0 . . . 15
5	CLK HIGH to R/$\overline{W}$ LOW	10	60	ns	
6	R/$\overline{W}$ width LOW	120		ns	
7	CLK LOW to $\overline{DS}$ LOW	5	35	ns	
8	CLK HIGH to $\overline{DS}$ HIGH	10	50	ns	
9	Dx set-up time	15		ns	x = 0 . . . 15
10	Dx hold time	0		ns	x = 0 . . . 15
11	Dx valid to $\overline{DS}$ LOW	0		ns	x = 0 . . . 15
12	$\overline{DS}$ HIGH to Dx invalid	5		ns	x = 0 . . . 15
13	$\overline{WAIT}$ set-up time	10		ns	
14	$\overline{WAIT}$ hold time	20		ns	
15	$\overline{INT}$ set-up time	5		ns	
16	$\overline{INT}$ hold time	20		ns	
17	CLK HIGH to $\overline{IACK}$ LOW		60	ns	
18	CLK HIGH to $\overline{IACK}$ HIGH		60	ns	
19	IFA, IFB, IFC, IFD set-up time	5		ns	
20	IFA, IFB, IFC, IFD hold time	20		ns	
21	$\overline{RST}$ set-up time	10		ns	
22	$\overline{RST}$ hold time	20		ns	
24	CLK HIGH to PAx or ARRx valid	5	50	ns	x = 0 . . 9 or . . 8
25	CLK HIGH to RR/$\overline{W}$ LOW		50	ns	
26	RR/$\overline{W}$ width LOW	120		ns	
27	CLK LOW to $\overline{RDS}$ LOW		20	ns	
28	CLK HIGH to $\overline{RDS}$ HIGH	10	40	ns	
29	RDx set-up time	15		ns	x = 0 . . . 15
30	RDx hold time	0		ns	x = 0 . . . 15
31	RDx valid to $\overline{RDS}$ LOW	0		ns	x = 0 . . . 15
32	$\overline{RDS}$ HIGH to RDx invalid	5		ns	x = 0 . . . 15
41	COX, CIX, COY, CIY width HIGH	125		ns	
42	COX, CIX, COY, CIY width LOW	125		ns	
43	COX(Y) LOW to $\overline{SOX(Y)RQ}$ LOW		40	ns	
44	COX(Y) LOW to $\overline{SOX(Y)RQ}$ HIGH		40	ns	
45	$\overline{SOX(Y)EN}$ set-up time	40		ns	
46	$\overline{SOX(Y)EN}$ hold time	40		ns	
47	COX(Y) LOW to next DOX(Y) valid		40	ns	
48	$\overline{SOX(Y)EN}$ LOW to DOX(Y) valid	10		ns	
49	$\overline{SOX(Y)EN}$ HIGH to DOX(Y) 3-state		40	ns	
53	CIX(Y) LOW to $\overline{SIX(Y)RQ}$ LOW		40	ns	
54	CIX(Y) LOW to $\overline{SIX(Y)RQ}$ HIGH		40	ns	
55	$\overline{SIX(Y)EN}$ set-up time	30		ns	
56	$\overline{SIX(Y)EN}$ hold time	10		ns	
57	DIX, DIY set-up time	10		ns	
58	DIX, DIY hold time	40		ns	
59	PDx set-up time	5		ns	x = 0 . . . 39
60	PDx hold time	15		ns	x = 0 . . . 39

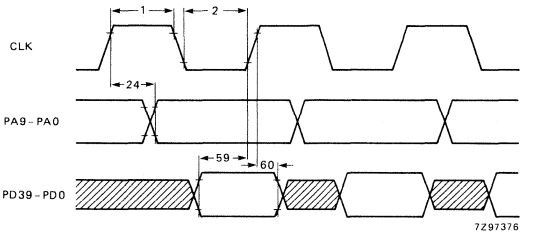


Fig. 4.3-1 Program memory access timing (for PCB5011 only).

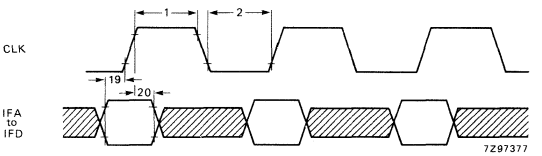


Fig. 4.3-2 User flag timing.

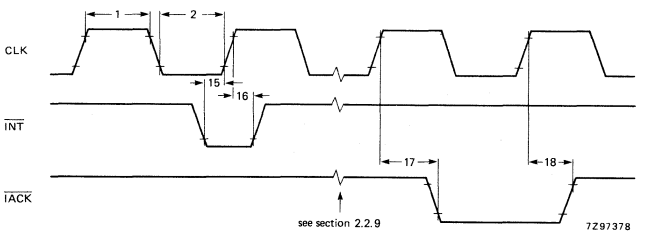


Fig. 4.3-3 Interrupt timing.

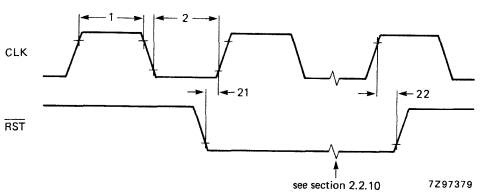


Fig. 4.3-4 Reset timing.

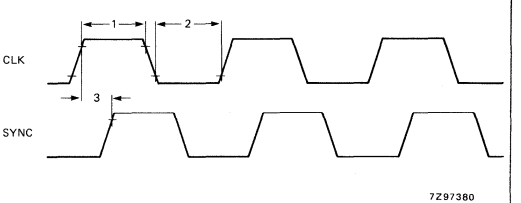


Fig. 4.3-5 Synchronization timing.

DEVELOPMENT DATA

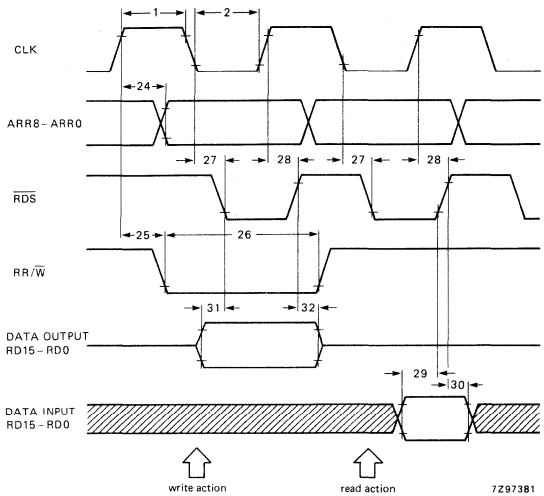
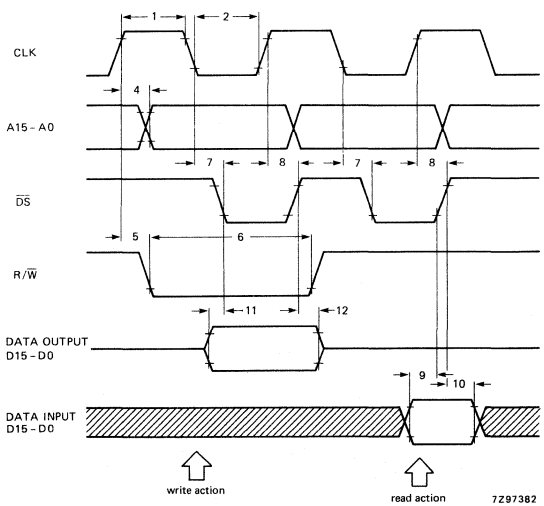
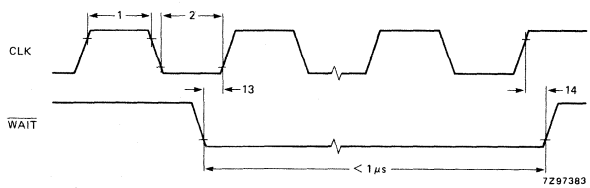


Fig. 4.3-6 Second parallel I/O timing (for PCB5011 only).



Note: (part of) the signals A15 – A0 can be used just as hardware controlling outputs if they are not needed for addressing external RAM.

Fig. 4.3-7 Parallel I/O timing ($\overline{\text{WAIT}}$ inactive).



Note: as long as $\overline{WAIT}$ = 0 all clocked (by CLK) processor actions are suspended i.e. the internal status and all signals remain unchanged.

Fig. 4.3-8 $\overline{WAIT}$ signal timing.

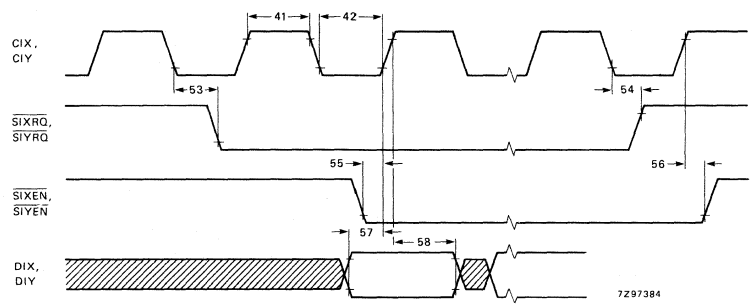


Fig. 4.3-9 Serial input timing.

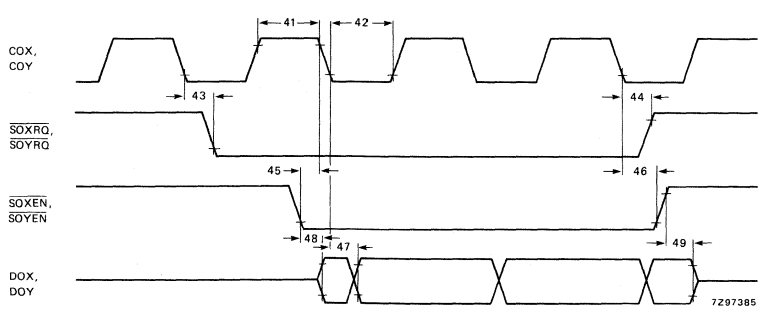


Fig. 4.3-10 Serial output timing.

DEVELOPMENT DATA

5.0 PACKAGE OUTLINES

5.1 PCB5010 PACKAGE OUTLINE (dimensions in mm)

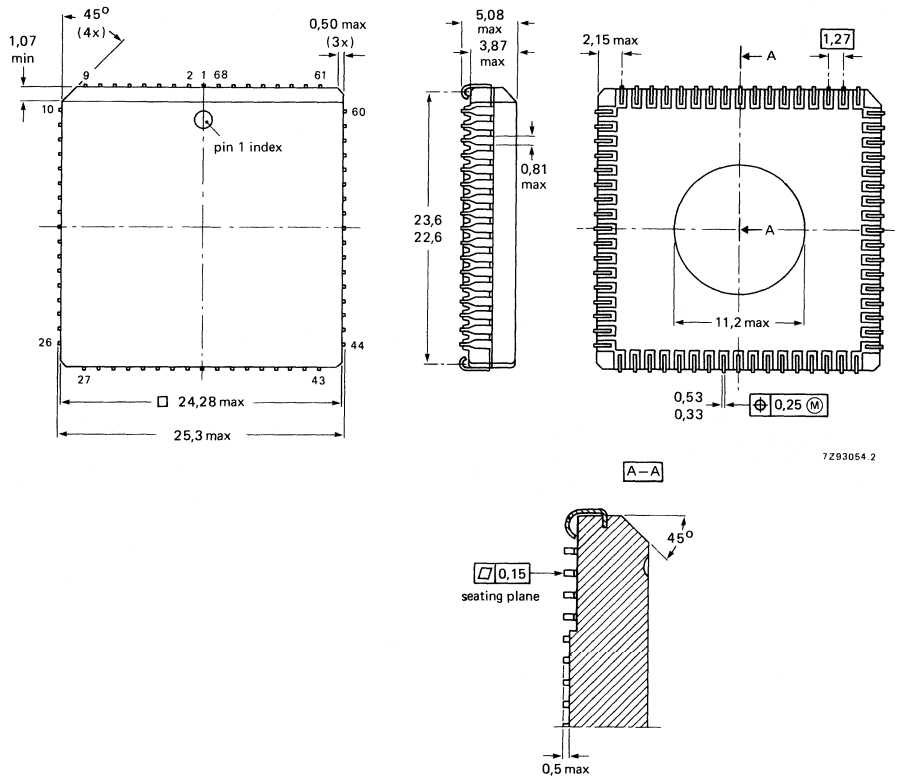


Fig. 5.1-1 68-lead plastic leaded-chip-carrier (PLCC) (SOT-188A) package for the PCB5010.

DEVELOPMENT DATA

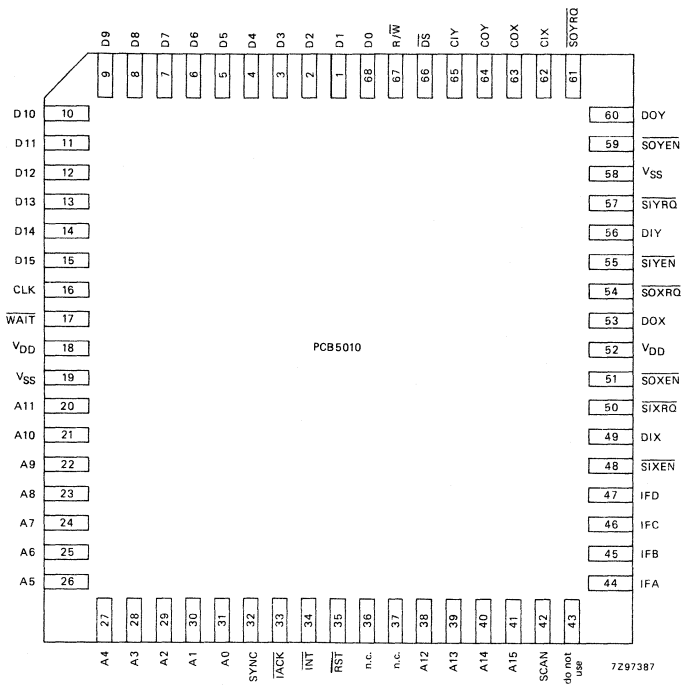


Fig. 5.1-2 Pinning for the PCB5010 (PLCC).

5.2 PCB5011 PACKAGE OUTLINE (dimensions in mm)

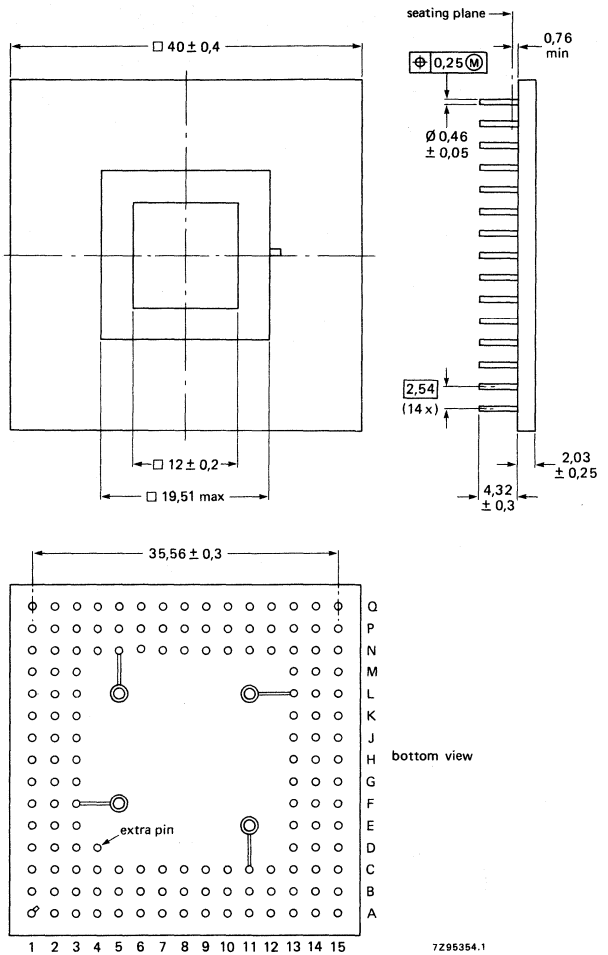


Fig. 5.2-1 144-pin grid array (PGA) package for the PCB5011.

DEVELOPMENT DATA

5.3 PIN ASSIGNMENT FOR Fig. 5.2-1

PCB5011 signal name	pin number
V_{DD}	N8, C8
V_{SS}	N9, A2
CLK	Q2
SCAN	B3
$\overline{RST}$	L14
D15	P3
D14	N4
D13	Q1
D12	P2
D11	N3
D10	M3
D9	P1
D8	N2
D7	L3
D6	M2
D5	N1
D4	M1
D3	L2
D2	L1
D1	K3
D0	K2
A15	A1
A14	B2
A13	C3
A12	C4
A11	N5
A10	Q3
A9	P5
A8	Q4
A7	N6
A6	P6
A5	Q5
A4	P7
A3	N7
A2	Q6
A1	Q7
A0	P8
R/$\overline{W}$	J2
$\overline{DS}$	K1
$\overline{WAIT}$	P4
DIX	B1

PCB5011 signal name	pin number
$\overline{SIXEN}$	C2
$\overline{SIXRQ}$	D2
CIX	D3
DIY	H3
$\overline{SIYEN}$	J3
$\overline{SIYRQ}$	H1
CIY	J1
DOX	C1
$\overline{SOXEN}$	E2
$\overline{SOXRQ}$	E3
COX	D1
DOY	G1
$\overline{SOYEN}$	F1
$\overline{SOYRQ}$	H2
COY	G3
$\overline{INT}$	M15
$\overline{IACK}$	M14
SYNC	N14
IFA	F3
IFB	F2
IFC	E1
IFD	G2
PA9	Q13
PA8	P12
PA7	N11
PA6	P13
PA5	Q14
PA4	N12
PA3	N13
PA2	P14
PA1	Q15
PA0	M13

(continued on next page)

PCB5011 signal name	pin number
PD39	C5
PD38	B4
PD37	A3
PD36	A4
PD35	B5
PD34	A5
PD33	C6
PD32	B6
PD31	B7
PD30	A6
PD29	A7
PD28	C7
PD27	A8
PD26	B8
PD25	A9
PD24	A10
PD23	C9
PD22	B9
PD21	A11
PD20	B10
PD19	C10
PD18	A12
PD17	B11
PD16	A13
PD15	C11
PD14	B12
PD13	A14
PD12	B13
PD11	C12
PD10	G15
PD9	G13
PD8	K14
PD7	L15
PD6	J14
PD5	J13
PD4	K15
PD3	J15
PD2	H14
PD1	H15
PD0	H13
ARR8	Q8
ARR7	Q9
ARR6	Q10
ARR5	P9
ARR4	P10
ARR3	N10
ARR2	Q11
ARR1	P11
ARR0	Q12

PCB5011 signal name	pin number
RD15	F15
RD14	G14
RD13	F14
RD12	F13
RD11	E15
RD10	E14
RD9	D15
RD8	C15
RD7	D14
RD6	E13
RD5	C14
RD4	B15
RD3	D13
RD2	C13
RD1	B14
RD0	A15
RR/$\overline{W}$	L13
$\overline{RDS}$	N15

VIDEO DISPLAY

SINGLE-CHIP COLOUR CRT CONTROLLER (EUROM)

GENERAL DESCRIPTION

The SAA5350 EUROM is a single-chip VLSI NMOS crt controller capable of handling all display functions required by the CEPT videotex terminal, model A4. Only minimal hardware is required to produce a videotex terminal using EUROM – the simplest configuration needs just a microcontroller and 4 Kbytes of display memory.

Features

- Minimal additional hardware required
- Screen formats of 40/80 character by 1-to-25 row display
- 512 alphanumeric or graphical characters on-chip or extendable off-chip
- Serial attribute storage (STACK) and parallel attribute storage
- Dynamically redefinable character (DRCS) capability over full field
- Interfaces with 8/16-bit microprocessors with optional direct memory access
- On-chip scroll map minimizes data to be transferred when scrolling
- On-chip colour map RAM (4096 locations) and three on-chip digital-to-analogue converters allow 32 colours on-screen
- On-chip digital-to-analogue converters are non-linear to compensate for crt non-linearity
- Memory interface capable of supporting multi-page terminals. EUROM can access up to 128 Kbytes of display memory
- Programmable cursor
- Programmable local status row
- Three synchronization modes:

stand-alone	built-in oscillator operating with an external 6 MHz crystal
simple slave	directly synchronized from the source of text composite sync
phase-locked slave	indirect synchronization allows picture-in-text displays (e.g. VCR/VLP video with text overlay)

- On-chip timing composite sync output
- Zoom feature which allows the height of any group of rows to be increased to enhance legibility

PACKAGE OUTLINE

40-lead DIL; plastic (SOT-129).

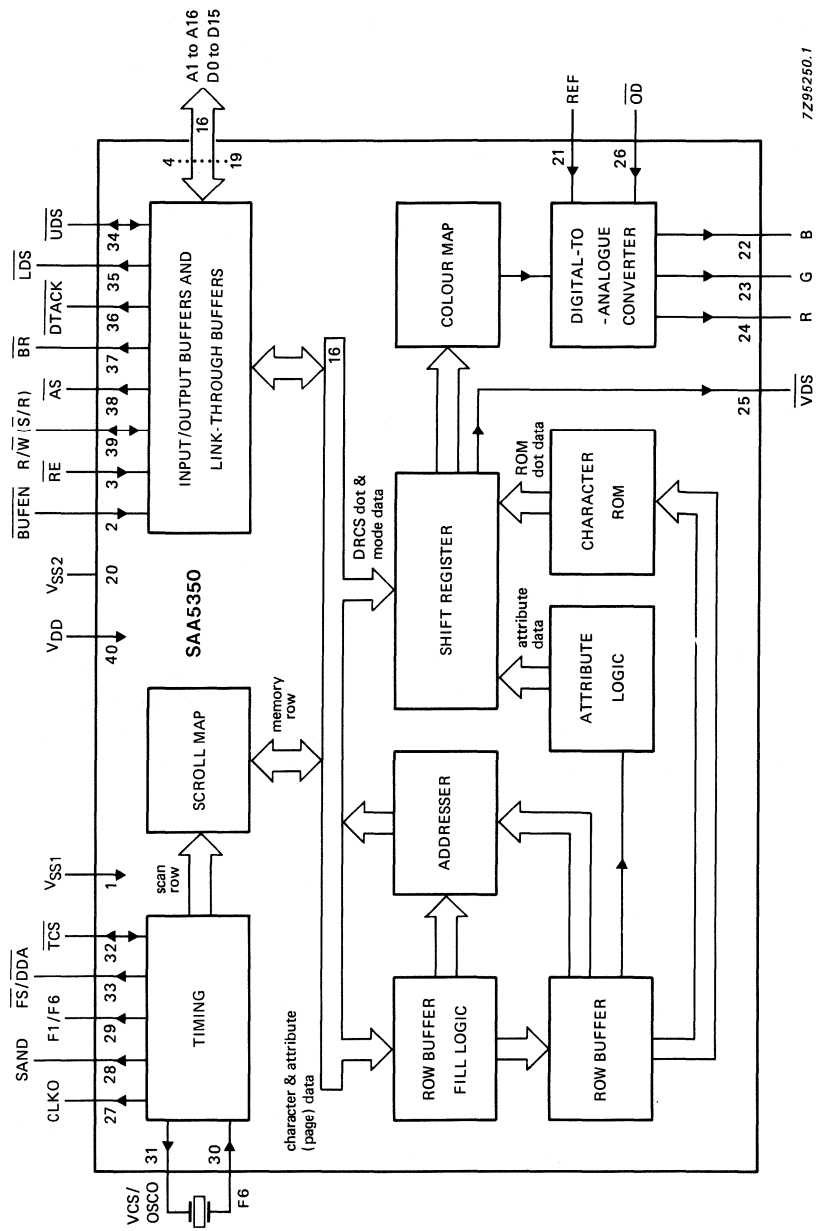


Fig. 1 Block diagram.

DEVELOPMENT DATA

PINNING

1	V_{SS1}	Ground 0 V.
2	$\overline{BUFEN}$	Buffer enable input to the 8-bit link-through buffer.
3	$\overline{RE}$	Register enable input. This enables A1 to A6 and $\overline{UDS}$ as inputs, and D8 to D15 as input/outputs.
4 to 19	A16 to A1/ D15 to D0	Multiplexed address and data bus input/outputs. These pins also function as the 8-bit link-through buffer.
20	V_{SS2}	Ground (0 V).
21	REF	Analogue reference input.
22	B	Analogue outputs (signals are gamma-corrected).
23	G	
24	R	
25	$\overline{VDS}$	Switching output for dot, screen (row), box and window video data; for use when video signal is present (e.g. from tv, VLP, alpha + photographic layer). This output is LOW for tv display and HIGH for text and will interface directly with a number of colour decoder ICs (e.g. TDA3560, TDA3505).
26	$\overline{OD}$	Output disable causing R, G, B and $\overline{VDS}$ outputs to go to high-impedance state. Can be used at dot-rate.
27	CLKO	12 MHz clock output for hard-copy dot synchronization (referenced to output dots).
28	SAND	Sandcastle feedback output for SAA5230 teletext video processor or other circuit. Used when the display must be locked to the video source (e.g. VLP). The phase-lock part of the sandcastle waveform can be disabled to allow free-running of the SAA5230 phase-locked loop.
29	F1/F6	1 MHz or 6 MHz output.
30	F6	6 MHz clock input (e.g. from SAA5230). Internal a.c. coupling is provided.
31	VCS/OSCO	Video composite sync input (e.g. from SAA5230) for phase reference of vertical display timing when locking to a video source (e.g. VLP) or, in stand-alone sync mode, output from internal oscillator circuit (fixed frequency).
32	$\overline{TCS}$	Text composite sync input/output depending on master/slave status.
33	$\overline{FS}/\overline{DDA}$	Field sync pulse output or defined-display-area flag output (both referenced to output dots).
34	$\overline{UDS}$	Upper data strobe input/output.
35	$\overline{LDS}$	Lower data strobe output.
36	$\overline{DTACK}$	Data transfer acknowledge (open drain output).
37	$\overline{BR}$	Bus request to microprocessor (open drain output).
38	$\overline{AS}$	Address strobe output to external address latches.
39	R/$\overline{W}$ ($\overline{S}$/R)	Read/write input/output. Also serves as send/receive for the link-through buffer.
40	V_{DD}	Positive supply voltage (+ 5 V).

PINNING (continued)

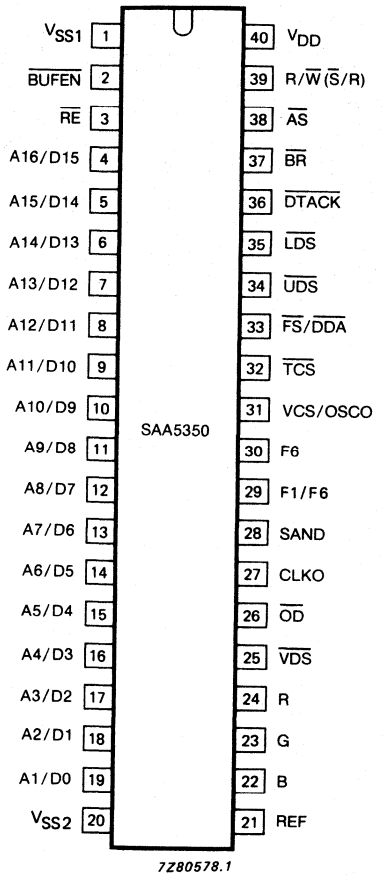


Fig. 2 Pinning diagram.

RATINGS

Limiting values in accordance with the Absolute Maximum System (IEC 134)

Supply voltage range (pin 40)	V_{DD}	−0,3 to + 7,5 V
Maximum input voltage (except F6, $\overline{TCS}$, REF)	V_{Imax}	−0,3 to + 7,5 V
Maximum input voltage (F6, $\overline{TCS}$)	V_{Imax}	−0,3 to + 10,0 V
Maximum input voltage (REF)	V_{REF}	−0,3 to + 3,0 V
Maximum output voltage	V_{Omax}	−0,3 to + 7,5 V
Maximum output current	I_{Omax}	10 mA
Operating ambient temperature range	T_{amb}	−20 to + 70 °C
Storage temperature range	T_{stg}	−55 to + 125 °C

Outputs other than CLKO, OSCO, R, G, B, and $\overline{VDS}$ are short-circuit protected.

DEVELOPMENT DATA

CHARACTERISTICS

V_{DD} = 5 V ± 5%; V_{SS} = 0 V; T_{amb} = −20 to + 70 °C; unless otherwise specified

parameter	symbol	min.	typ.	max.	unit
SUPPLY					
Supply voltage (pin 40)	V_{DD}	4,75	5,0	5,25	V
Supply current (pin 40)	I_{DD}	–	–	350	mA
INPUTS					
F6 (note 1)					
Slave modes (Fig. 3)					
Input voltage (peak-to-peak value)	$V_{I(p-p)}$	1,0	–	7,0	V
Input peaks relative to 50% duty factor	$\pm V_P$	0,2	–	3,5	V
Input leakage current at V_I = 0 to 10 V; T_{amb} = 25 °C	I_{LI}	–	–	20	μA
Input capacitance	C_I	–	–	12	pF
Stand-alone mode (Fig. 4)					
Series capacitance of crystal	C_1	–	28	–	fF
Parallel capacitance of crystal	C_0	–	7,1	–	pF
Resonance resistance of crystal	R_r	–	–	60	Ω
Gain of circuit	G	–	–	tbf	V/V
$\overline{\text{BUFEN}}$, $\overline{\text{RE}}$, $\overline{\text{OD}}$					
Input voltage LOW	V_{IL}	0	–	0,8	V
Input voltage HIGH	V_{IH}	2,0	–	6,5	V
Input current at V_I = 0 to V_{DD}+0,3 V; T_{amb} = 25 °C	I_I	−10	–	+ 10	μA
Input capacitance	C_I	–	–	7	pF
REF (Fig. 5)					
Input voltage	V_{REF}	0	1 to 2	2,7	V
Resistance (pin 21 to pin 20) with REF supply and R, G, B outputs OFF	R_{REF}	–	125	–	Ω

CHARACTERISTICS (continued)

parameter	symbol	min.	typ.	max.	unit
OUTPUTS					
SAND					
Output voltage high level at I_O = 0 to −10 μA	V_{OH}	4,2	–	V_{DD}	V
Output voltage intermediate level at I_O = −10 to + 10 μA	V_{OI}	1,3	2,0	2,7	V
Output voltage low level at I_O = 0,2 mA	V_{OL}	0	–	0,2	V
Load capacitance	C_L	–	–	130	pF
F1/F6, CLKO, $\overline{\text{DDA}}$/$\overline{\text{FS}}$					
Output voltage HIGH at I_{OH} = −200 μA	V_{OH}	2,4	–	V_{DD}	V
Output voltage LOW at I_{OL} = 3,2 mA	V_{OL}	0	–	0,4	V
Load capacitance	C_L	–	–	50	pF
$\overline{\text{LDS}}$, $\overline{\text{AS}}$					
Output voltage HIGH at I_{OH} = −200 μA	V_{OH}	2,4	–	V_{DD}	V
Output voltage LOW at I_{OL} = 3,2 mA	V_{OL}	0	–	0,4	V
Load capacitance	C_L	–	–	200	pF
$\overline{\text{DTACK}}$, $\overline{\text{BR}}$ (open drain outputs)					
Output voltage LOW at I_{OL} = 3,2 mA	V_{OL}	0	–	0,4	V
Load capacitance	C_L	–	–	150	pF
Capacitance (OFF state)	C_{OFF}	–	–	7	pF
R, G, B (note 2)					
Output voltage HIGH (note 3) at I_{OH} = −100 μA; V_{REF} = 2,7 V	V_{OH}	2,4	–	–	V
Output voltage LOW at I_{OL} = 2 mA	V_{OL}	–	–	0,4	V
Output resistance during line blanking	R_{OBL}	–	–	150	Ω
Output capacitance (OFF state)	C_{OFF}	–	–	12	pF
Output leakage current (OFF state) at V_I = 0 to V_{DD}+ 0,3 V; T_{amb} = 25 °C	I_{OFF}	−10	–	+ 10	μA

parameter	symbol	min.	typ.	max.	unit
$\overline{VDS}$					
Output voltage HIGH at $I_{OH} = -250\ \mu A$	V_{OH}	2,4	–	V_{DD}	V
Output voltage LOW at $I_{OL} = 2$ mA	V_{OL}	0	–	0,4	V
Output voltage LOW at $I_{OL} = 1$ mA	V_{OL}	0	–	0,2	V
Output leakage current (OFF state) at $V_I = 0$ to $V_{DD}+0,3$ V; $T_{amb} = 25\ ^oC$	I_{OFF}	−10	–	+ 10	μA
INPUT/OUTPUTS					
VCS/OSCO					
Input voltage HIGH	V_{IH}	2,0	–	6,0	V
Input voltage LOW	V_{IL}	0	–	0,8	V
Input current (output OFF) at $V_I = 0$ to $V_{DD}+0,3$ V; $T_{amb} = 25\ ^oC$	I_I	−10	–	+ 10	μA
Input capacitance	C_I	–	–	10	pF
Load capacitance	C_L	–	–	50	pF
$\overline{TCS}$					
Input voltage HIGH	V_{IH}	3,5	–	10,0	V
Input voltage LOW	V_{IL}	0	–	1,5	V
Input current at $V_I = 0$ to $V_{DD}+0,3$ V; $T_{amb} = 25\ ^oC$	I_I	−10	–	+ 10	μA
Input capacitance	C_I	–	–	10	pF
Output voltage HIGH at $I_{OH} = -200$ to $100\ \mu A$	V_{OH}	2,4	–	6,0	V
Output voltage LOW at $V_{OL} = 3,2$ mA	V_{OL}	0	–	0,4	V
Load capacitance	C_L	–	–	50	pF
A1/D0 to A16/D15, $\overline{UDS}$, R/$\overline{W}$					
Input voltage LOW	V_{IL}	0	–	0,8	V
Input voltage HIGH	V_{IH}	2,0	–	6,0	V
Input current at $V_I = 0$ to $V_{DD}+0,3$ V; $T_{amb} = 25\ ^oC$	I_I	−10	–	+ 10	μA
Input capacitance	C_I	–	–	10	pF
Output voltage HIGH at $I_{OH} = -200\ \mu A$	V_{OH}	2,4	–	V_{DD}	V
Output voltage LOW at $I_{OL} = 3,2$ mA	V_{OL}	0	–	0,4	V
Load capacitance	C_L	–	–	200	pF

DEVELOPMENT DATA

CHARACTERISTICS (continued)

parameter	symbol	min.	typ.	max.	unit
TIMING					
F6 (Fig. 3)					
Rise and fall times	t_r, t_f	10	–	80	ns
Frequency	f_{F6}	5,9	–	6,1	MHz
CLKO, F1/F6, R, G, B, $\overline{VDS}$, $\overline{FS}$/$\overline{DDA}$, $\overline{OD}$ (notes 4, 5 and Fig. 6)					
CLKO HIGH time	t_{CLKH}	25	–	–	ns
CLKO LOW time	t_{CLKL}	15	–	–	ns
CLKO rise and fall times	t_{CLKr} t_{CLKf}	–	–	10	ns
CLKO HIGH to R, G, B, $\overline{VDS}$ change	t_{VCH}	10	–	–	ns
R, G, B, $\overline{VDS}$ valid to CLKO rise	t_{VOC}	10	–	–	ns
CLKO HIGH to R, G, B, $\overline{VDS}$ valid	t_{COV}	–	–	60	ns
CLKO HIGH to R, G, B, $\overline{VDS}$ floating after $\overline{OD}$ fall	t_{FOD}	0	–	30	ns
Skew between outputs R, G, B, $\overline{VDS}$	t_{VS}	–	–	20	ns
R, G, B, $\overline{VDS}$ rise and fall times	t_{Vr}, t_{Vf}	–	–	30	ns
CLKO HIGH to R, G, B, $\overline{VDS}$ active after $\overline{OD}$ rise	t_{AOD}	0	–	60	ns
CLKO HIGH to $\overline{FS}$/$\overline{DDA}$ change	t_{COD}	10	–	55	ns
$\overline{FS}$/$\overline{DDA}$ valid to CLKO rise	t_{DOC}	5	–	–	ns
F1 HIGH time (note 6)	t_{F1H}	400	500	580	ns
F1 LOW time (note 6)	t_{F1L}	400	500	580	ns
F6 HIGH time	t_{F6H}	40	83	120	ns
F6 LOW time	t_{F6L}	40	83	120	ns
$\overline{OD}$ to CLKO rise set-up	t_{ODS}	–	–	45	ns
$\overline{OD}$ to CLKO HIGH hold	t_{ODH}	–	–	0	ns
MEMORY ACCESS TIMING (notes 7, 8, 9 and Fig. 7)					
$\overline{UDS}$, $\overline{LDS}$, $\overline{AS}$					
Cycle time	t_{cyc}	480	500	520	ns
$\overline{UDS}$ HIGH to bus-active for address output	t_{SAA}	75	–	–	ns
Address valid set-up to $\overline{AS}$ fall	t_{ASU}	20	–	–	ns
Address valid hold from $\overline{AS}$ LOW	t_{ASH}	20	–	–	ns
Address float to $\overline{UDS}$ fall	t_{AFS}	0	–	–	ns

parameter	symbol	min.	typ.	max.	unit	
$\overline{AS}$ LOW to $\overline{UDS}$ fall delay	t_{ATD}	50	–	–	ns	
$\overline{UDS}$, $\overline{LDS}$ HIGH time	t_{HDS}	220	–	–	ns	
$\overline{UDS}$, $\overline{LDS}$ LOW time	t_{LDS}	200	–	–	ns	
$\overline{AS}$ HIGH time	t_{HAS}	125	–	–	ns	
$\overline{AS}$ LOW time	t_{LAS}	320	–	–	ns	
$\overline{AS}$ LOW to $\overline{UDS}$ HIGH	t_{AUH}	305	–	–	ns	
Data valid set-up to $\overline{UDS}$ rise	t_{DSU}	30	–	–	ns	
Data valid hold from $\overline{UDS}$ HIGH	t_{DSH}	10	–	–	ns	
$\overline{UDS}$ HIGH to $\overline{AS}$ rise delay	t_{UAS}	0	–	15	ns	←
$\overline{AS}$ LOW to data valid	t_{AFA}	–	–	270	ns	
Link-through buffers (notes 7, 8 and Fig. 8)						
$\overline{BUFEN}$ LOW to output valid	t_{BEA}	–	–	100	ns	
Link-through delay time	t_{LTD}	–	–	85	ns	
Input data float prior to direction change	t_{IFR}	0	–	–	ns	
Output float after direction change	t_{OFR}	–	–	60	ns	
Output float after $\overline{BUFEN}$ HIGH	t_{BED}	–	–	60	ns	
Microprocessor READ from EUROM (Fig. 9)						
R/$\overline{W}$ HIGH set-up to $\overline{UDS}$ fall	t_{RUD}	0	–	–	ns	
$\overline{UDS}$ LOW to returned-data access time	t_{UDA}	–	–	210	ns	
$\overline{RE}$ LOW to returned data access time	t_{REA}	–	–	210	ns	
Data valid to $\overline{DTACK}$ LOW delay	t_{DTL}	40	–	–	ns	
$\overline{DTACK}$ LOW to $\overline{UDS}$ rise	t_{DLU}	10	–	–	ns	←
$\overline{UDS}$ HIGH to $\overline{DTACK}$ rise	t_{DTR}	0	–	75	ns	
$\overline{UDS}$ HIGH to address hold	t_{DSA}	10	–	–	ns	
$\overline{UDS}$ HIGH to data hold	t_{DSH}	10	–	–	ns	
$\overline{UDS}$ HIGH to $\overline{RE}$ rise	t_{SRE}	10	–	–	ns	
$\overline{UDS}$ HIGH to R/$\overline{W}$ fall	t_{UDR}	0	–	–	ns	
$\overline{UDS}$ LOW to $\overline{DTACK}$ LOW	t_{DSD}	250	–	350	ns	
Address valid to $\overline{UDS}$ fall	t_{AUL}	0	–	–	ns	

DEVELOPMENT DATA

CHARACTERISTICS (continued)

parameter	symbol	min.	typ.	max.	unit
MEMORY ACCESS TIMING (continued)					
Microprocessor WRITE to EUROM (Fig. 10)					
Write cycle time (note 10)	t_{WCY}	500	–	–	ns
R/$\overline{W}$ LOW set-up to $\overline{UDS}$ fall	t_{WUD}	0	–	–	ns
$\overline{RE}$ LOW to $\overline{UDS}$ fall	t_{RES}	30	–	–	ns
Address valid to $\overline{UDS}$ fall	t_{ASS}	30	–	–	ns
$\overline{UDS}$ LOW time	t_{LUS}	100	–	–	ns
Data valid to $\overline{UDS}$ rise	t_{DSS}	80	–	–	ns
$\overline{UDS}$ LOW to $\overline{DTACK}$ LOW	t_{DTA}	0	–	60	ns
$\overline{UDS}$ HIGH to $\overline{DTACK}$ rise	t_{DTR}	0	–	75	ns
$\overline{UDS}$ HIGH to data hold	t_{DSH}	10	–	–	ns
$\overline{UDS}$ HIGH to address hold	t_{DSA}	10	–	–	ns
$\overline{UDS}$ HIGH to $\overline{RE}$ rise	t_{SRE}	10	–	–	ns
$\overline{UDS}$ HIGH to R/$\overline{W}$ rise	t_{UDW}	0	–	–	ns
F1/F6 to memory access cycle (Fig. 11)					
$\overline{UDS}$ HIGH to F6 (component of F1/F6) rise	t_{UF6}	20	–	–	ns
F6 (component of F1/F6) HIGH to $\overline{UDS}$ rise	t_{F6U}	40	–	–	ns
SYNCHRONIZATION and BLANKING					
$\overline{TCS}$, SAND, $\overline{FS}$/$\overline{DDA}$					
See Fig. 12 for timing relationships and Fig. 13 for vertical sync and blanking waveforms.					

Notes to the characteristics

1. Pin 30 must be biased externally as it is internally a.c. coupled.
2. 16-level analogue voltage outputs.
3. Output voltage guaranteed when programmed for top level.
4. CLKO, R, G, B, F1/F6, $\overline{VDS}$: C_L = 25 pF
   $\overline{FS}$/$\overline{DDA}$: C_L = 50 pF
5. CLKO, F1/F6, $\overline{VDS}$, $\overline{FS}$/$\overline{DDA}$: reference levels = 0,8 to 2,0 V
   R, G, B: reference levels = 0,8 to 2,0 V with V_{REF} = 2,7 V
6. These times may momentarily be reduced to a nominal 83 ns in slave-sync mode at the moment of re-synchronization.
7. C_L = 150 pF.
8. Reference levels = 0,8 to 2,0 V.
9. F6 input at 6 MHz.
10. Microprocessor write cycle times of less than 500 ns are permitted but often result in Wait States being generated, the precise timing of $\overline{DTACK}$ will then depend on the internal synchronization time.

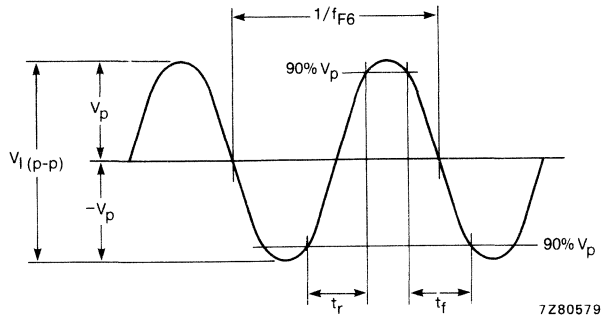
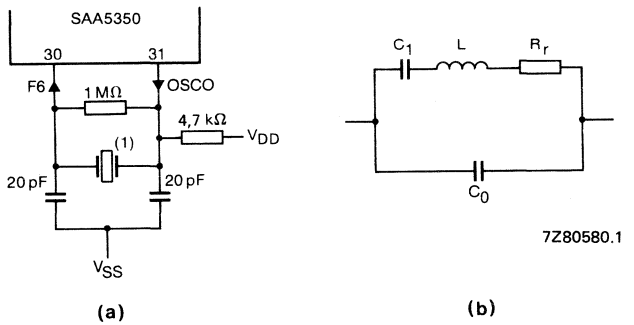


Fig. 3 F6 input waveform.



(1) Catalogue number of crystal: 4322 143 04101

Fig. 4(a) Oscillator circuit for SAA5350 stand-alone sync mode and (b) equivalent circuit of crystal at resonance (see **characteristics** for values).

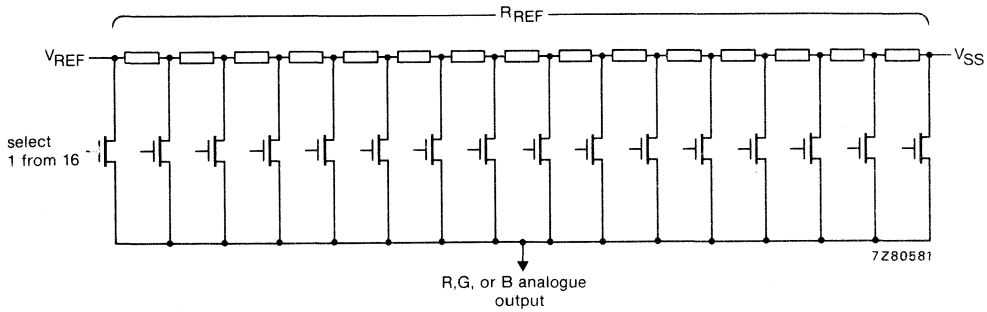


Fig. 5 Circuit arrangement giving one-of-sixteen reference voltage levels for the R, G or B analogue outputs.

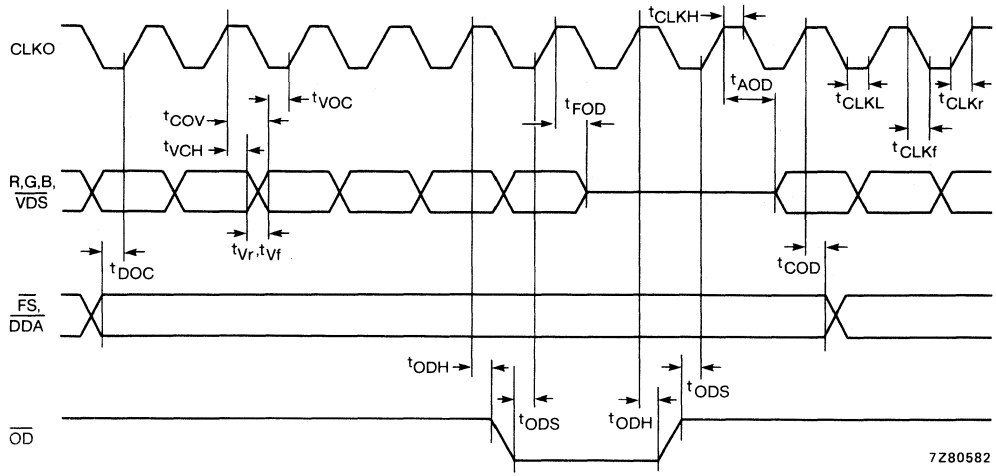


Fig. 6 Video timing.

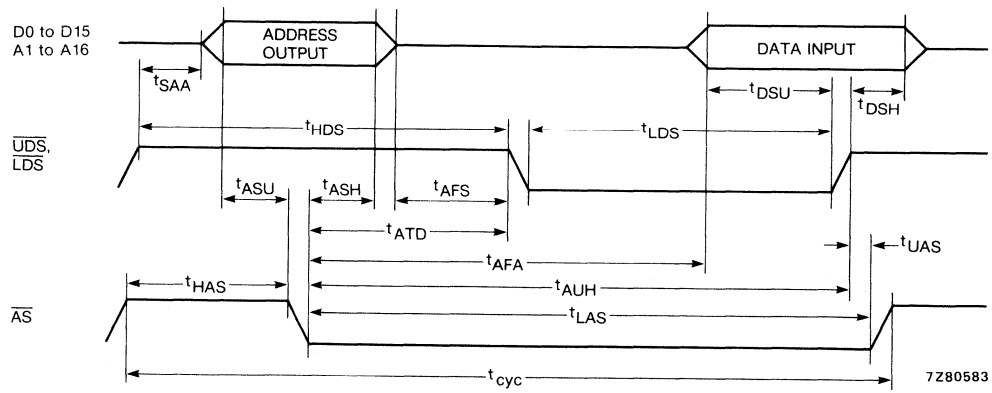


Fig. 7 Memory access timing.

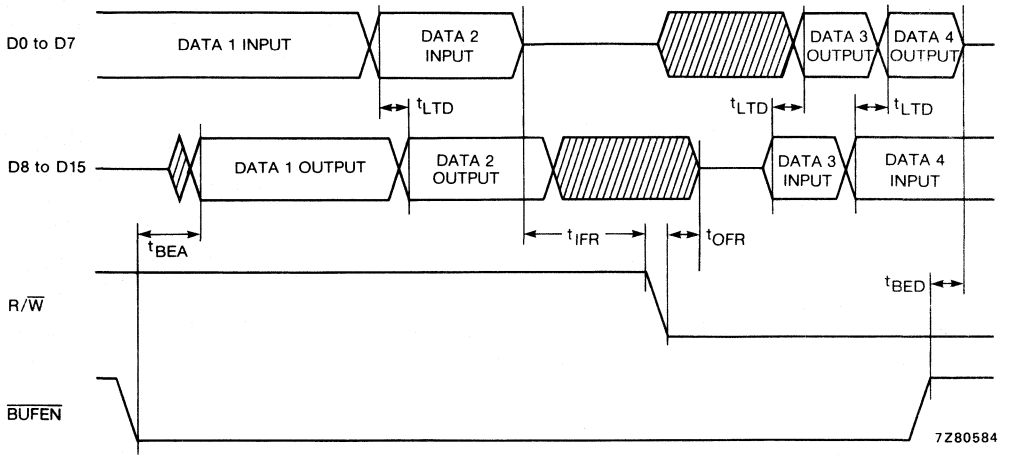


Fig. 8 Timing of link-through buffers.

DEVELOPMENT DATA

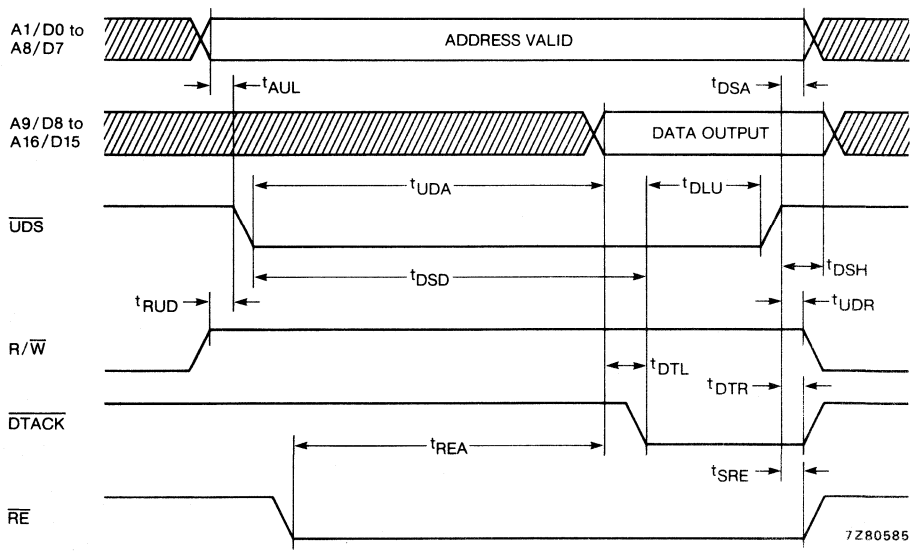


Fig. 9 Timing of microprocessor read from EUROM.

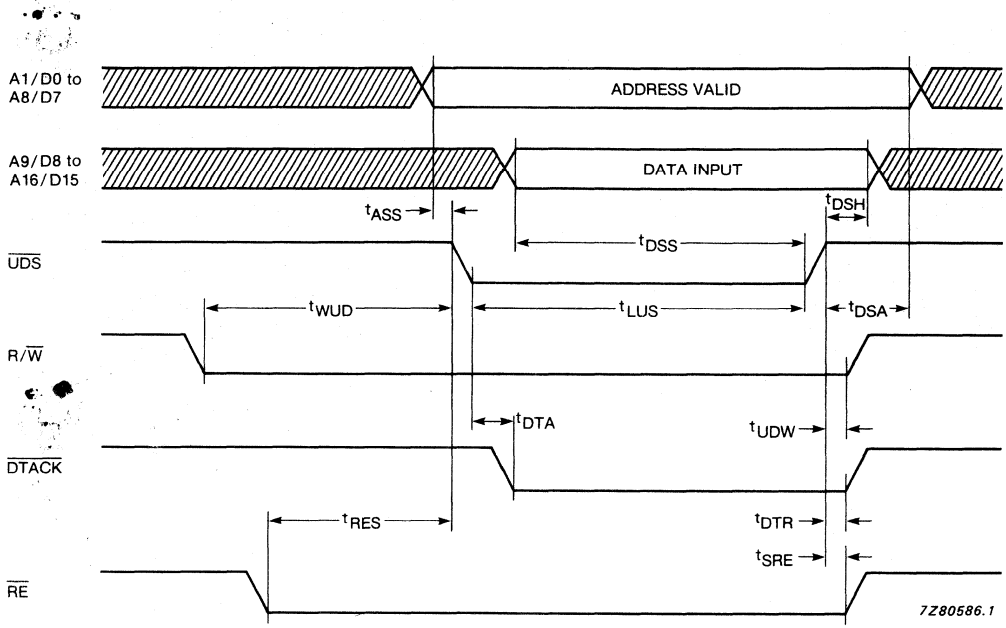


Fig. 10 Timing of microprocessor write to EUROM.

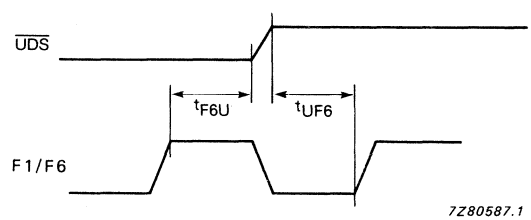


Fig. 11 Timing of F1/F6 to memory access cycle.

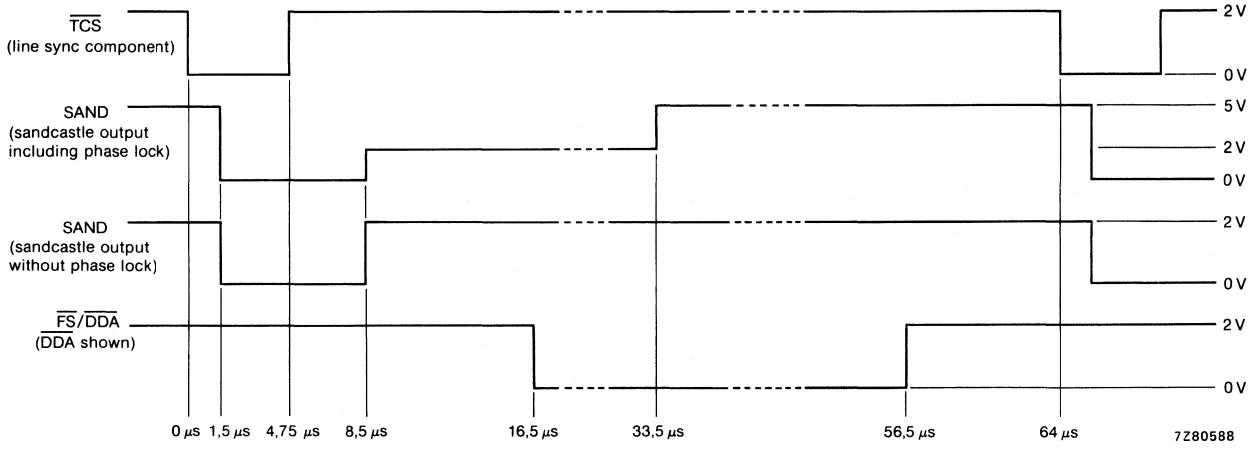


Fig. 12 Timing of synchronization and blanking outputs; all timings are nominal and assume f_{F6} = 6 MHz.

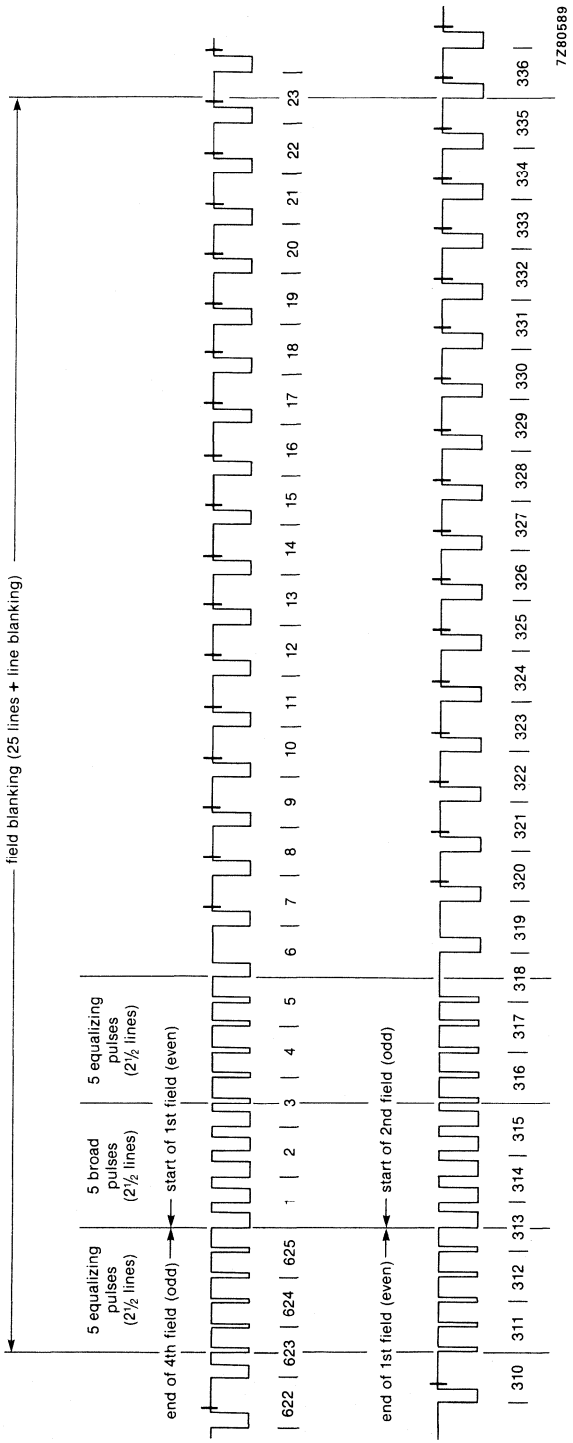


Fig. 13 Vertical synchronization and blanking waveforms; separation of broad pulses = 4,75 μs; equalizing pulse widths = 2,25 μs.

APPLICATION INFORMATION

More detailed application information is available on request

BASIC VIDEOTEX DECODER CONFIGURATION

A basic, practical decoder configuration is shown in Fig. 14, reference should also be made to the block diagram Fig. 1.

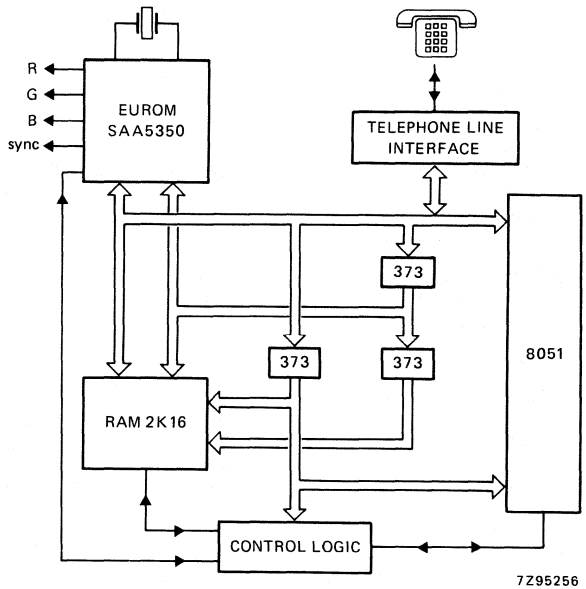


Fig. 14 Basic videotex decoder configuration.

Character and attribute data is fetched from the external memory, processed by the row buffer fill logic according to the stack coding scheme (in stack mode) and then fed into one half of the dual display row buffer. The data fetch process takes place during one line-flyback period (per row) and, since time is required to complete the fill, the other half of the dual row buffer is used for display. The row buffers exchange functions on alternate rows – each holds the 40 columns of 32 bits required to define explicitly every character in a row.

The addresser is used for row buffer filling and for fetching screen colours, and during the display time it is also used for addressing DRCS characters.

Timing

The timing chain operates from an external 6 MHz clock or an on-chip fixed-frequency crystal oscillator. The basic video format is 40 characters per row, 24/25 rows per page and 10 video lines per row. EUROM will also operate with 20/21 rows per page and 12 video lines per row. The two extra lines per row are added symmetrically and contain background colour only for ROM-based alpha-numeric characters. DRCS characters, block and smooth mosaics and line drawing characters occupy all 12 lines.

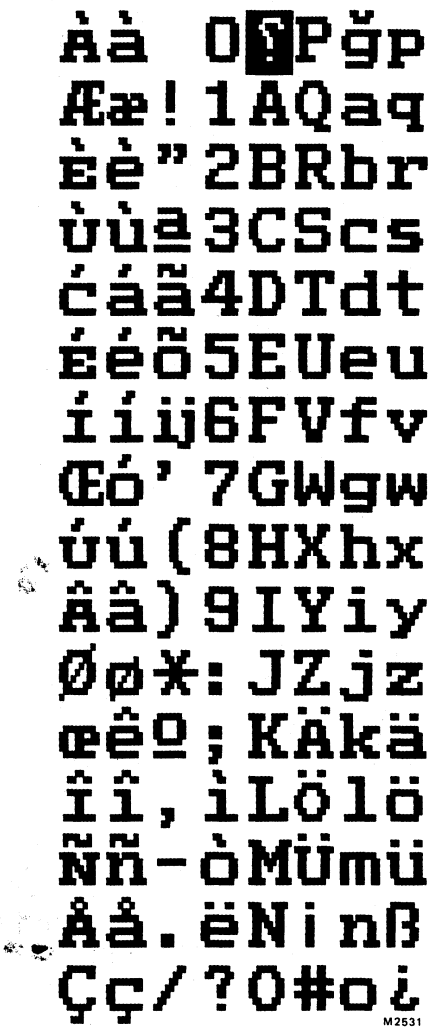
The display is generated to the normal 625-line/50 Hz scanning standard (interlaced or non-interlaced). In addition to composite sync (pin 32) for conventional timebases, a clock output at 1 MHz or 6 MHz (pin 29) is available for driving other videotex devices, and a 12 MHz clock (pin 27) is available for hard-copy dot synchronization. A defined-display-area timing signal (pin 33) simplifies the application of external peripherals such as a light pen; this signal is nominally coincident with the character dot information.

DEVELOPMENT DATA

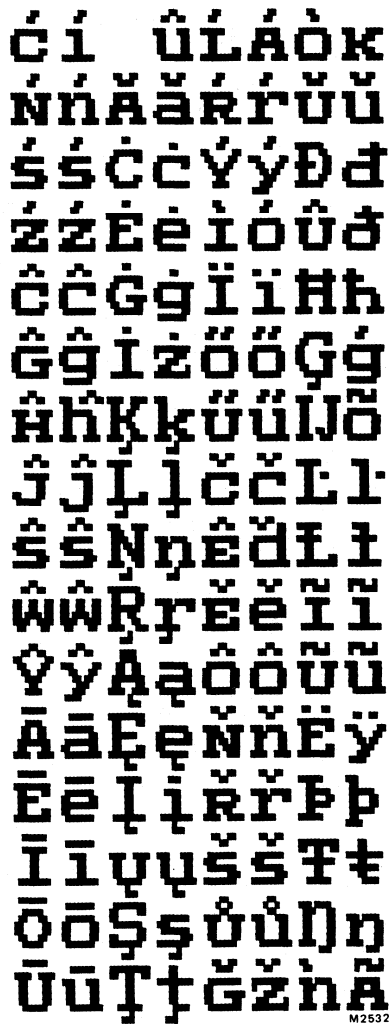
APPLICATION INFORMATION (continued)

Character generation

EUROM supports eight character tables, each of (nominally) 128 characters. Four tables are in on-chip ROM and contain fixed characters and four are stored in an external RAM. The contents of the fixed character tables (Tables 0 to 3) are shown in Figs 15 and 16.

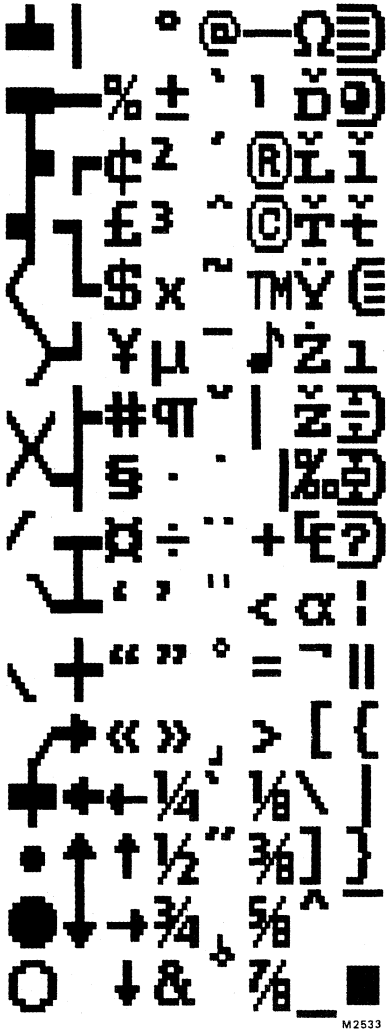


(a)

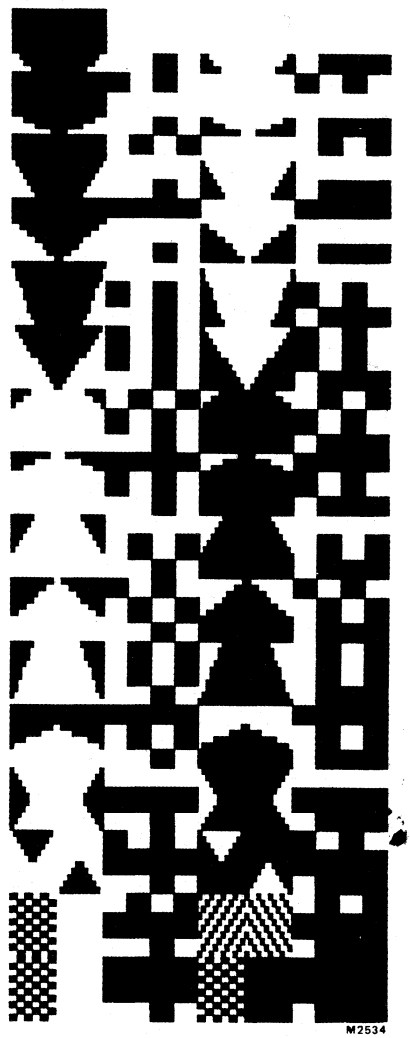


(b)

Fig. 15 On-chip characters: (a) Table 0; (b) Table 1.



(a)



(b)

Fig. 16 On-chip characters: (a) Table 2; (b) Table 3.

APPLICATION INFORMATION (continued)

Character generation (continued)

The 128 most commonly used characters are contained in Table 0, these are the standard upper and lower-case letters of the Roman alphabet, numerals, punctuation and the more common accented characters. In normal text transmission, Table 0 is used most of the time. Table 1 contains other accented characters. Miscellaneous characters, mathematical symbols, the line drawing character set and accents without associated symbols are contained in Table 2. Table 3 contains the block mosaics for the basic alpha-mosaic service and also the new smooth mosaics.

The four tables stored in the external RAM (Tables 4 to 7) are used for DRCS.

Scroll map

The scroll map uses a 26-byte area of on-chip RAM and functions in association with the timing chain. It maps the scan row on to the fetched memory row so allowing the stored page to be displayed in any row order. For each row, a 1-byte pointer to the display memory row is stored in the scroll map. This allows scrolling without the need for data transfer to, or from, side storage.
Additional control bits are stored, allowing 1 to 25 rows to be displayed at any location on the screen.

Colour map and digital-to-analogue converters

The colour map RAM contains thirty-two 12-bit words that are loaded by the microprocessor and read out in three 4-bit groups at pixel rate. Each group is fed to a non-linear (gamma-corrected) D-A converter. The resulting R, G and B outputs are low-impedance with peak-to-peak amplitudes controlled by the reference voltage applied at pin 21.

Cursor

The cursor is available in the stack mode. Its position, character code, character table, foreground colour, background colour, lining and flash attributes are all software programmable via internal register bits.

NON-VIDEOTEX APPLICATIONS

For non-Videotex applications, the device will also support the following operating modes:

Explicit fill mode. An alternative 40 character/rows mode which does not use the memory compression technique of stack coding. More display memory is required but there are no limitations on the number of display attribute changes per row.

80 characters/rows mode. When operating with 80 characters per row, the available display attributes are eight foreground colours, eight (potentially different) background colours (including transparent) as well as underline and blink.

Full field DRCS mode. This mode is not mutually exclusive to the explicit fill and 80 characters/rows modes but rather the available DRCS memory is expanded so that the whole screen can be covered, thus enabling a 'bit map'. All ROM-based characters and all display attributes remain available.

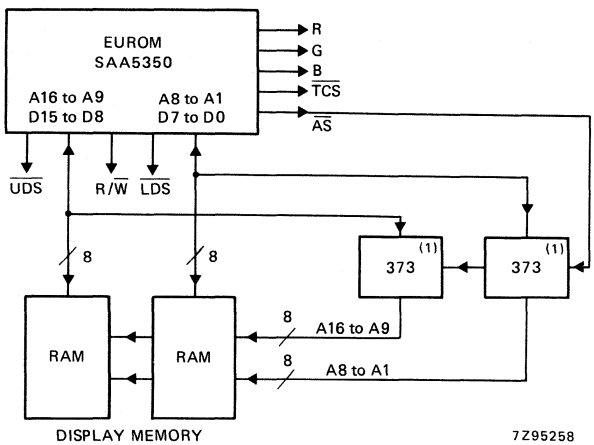
MICROPROCESSOR and RAM BUS INTERFACE

Three types of data transfer take place at the bus interface:

- EUROM fetches data from the display memory
- The microprocessor reads from, or writes to, EUROM's internal register map
- The microprocessor accesses the display memory

EUROM access to display memory (Figs 17 and 18)

EUROM accesses the external display memory via a 16-bit multiplexed address and data bus with a cycle time of 500 ns. The address strobe ($\overline{AS}$) signal from EUROM flags the bus cycle and writes the address into octal latches (74LS373). The display data is stored in bytes of upper (most-significant) and lower (least-significant) display information and is always fetched in pairs of bytes (upper + lower = 16 bits). The upper and lower display RAM sections are enabled simultaneously by the upper and lower data strobes (respectively $\overline{UDS}$ and $\overline{LDS}$) which are always asserted together to fetch a 16-bit word. The read/write control R/$\overline{W}$ is included although EUROM only reads from the display memory.



(1) 74LS373 octal transparent latch (3-state)

Fig. 17 Simple RAM interface circuit for display memory access.

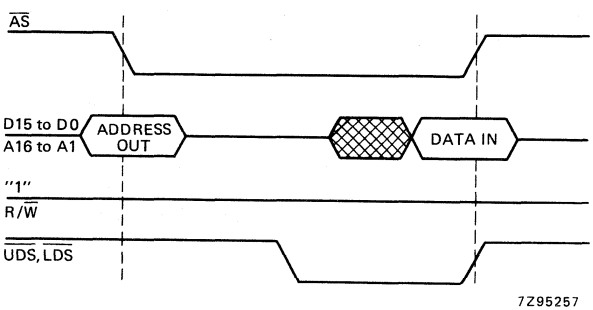


Fig. 18 Bus timing for display memory access.

APPLICATION INFORMATION (continued)

EUROM access to display memory (continued)

The display memory organization uses the word/byte addressing convention adopted for the SCN68000 microprocessor series. Data fetched on the 16-bit bus is considered in terms of bytes where the even numbered bytes use the upper (most-significant) part of the bus as shown in Fig. 19. The word addresses are numerically the same as the upper byte that they contain – there are no odd-numbered word addresses.

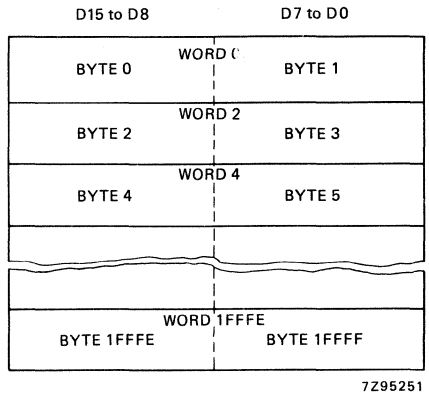


Fig. 19 Display memory word/byte organization.

Warning time

As EUROM is a real-time display device, it must have direct access to the display memory with priority over the microprocessor and other peripheral devices. This is achieved by EUROM issuing a bus request ($\overline{BR}$) signal for the duration of the memory access plus a programmable advance warning time which allows the microprocessor to complete its current bus cycle.

In systems where the buses of the microprocessor and EUROM are intimately connected (connected systems), $\overline{BR}$ may be used to suspend all microprocessor activity so that EUROM can act as a dedicated DMA controller. In systems where the two buses are separated by buffers (disconnected systems), $\overline{BR}$ may be used either to generate an interrupt or as a direct signal. To these ends, the warning time between the assertion of $\overline{BR}$ and the beginning of EUROM's bus activity is programmable to be between 0 and 23 μs.

Microprocessor access to register map

EUROM has a set of internal registers which, when memory-mapped, behave as an 8-bit wide RAM connected to the upper part of the data bus (Fig. 20). The control signals $\overline{UDS}$ and R/$\overline{W}$ are reversed to become inputs and the register map is enabled by the signal $\overline{RE}$. Addresses are input via the lower part of the bus. A data transfer acknowledge signal ($\overline{DTACK}$) indicates to the microprocessor that the data transfer is complete.

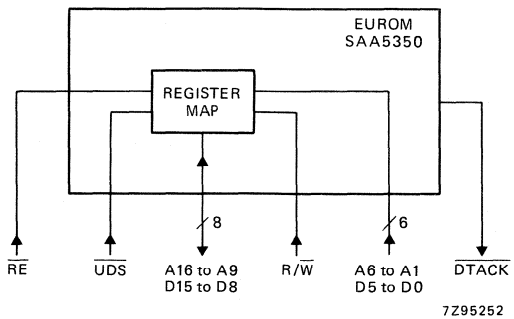


Fig. 20 Microprocessor access to register map.

The main data and address paths used in a connected 68000 interface are shown in Fig. 21. The outputs from the octal latches (74LS373) are enabled only when the 68000 has made the bus available in response to a bus request ($\overline{BR}$). When the register map is accessed data is transferred via the upper part of the bus and the microprocessor's low-order address is passed to EUROM via the octal buffers (74LS244). At the same time the bidirectional buffers (74LS245) disable the signals from the low order data bus of the 68000.

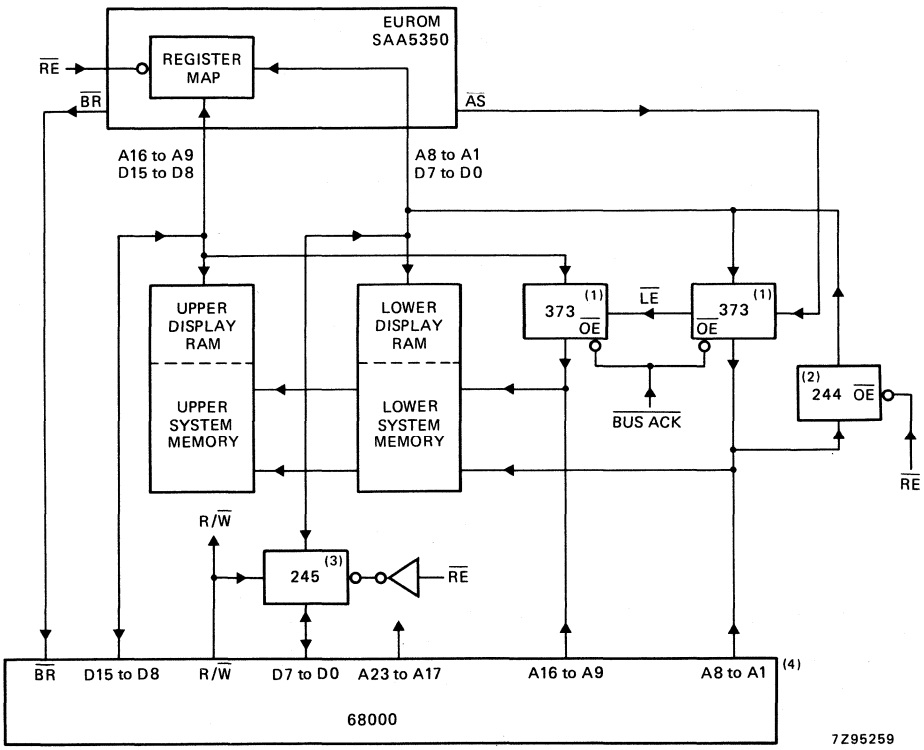
The buffers '244 and '245 may be omitted in a 16-bit write-only configuration where the least-significant data byte is interpreted by EUROM as an address. Here it will generally be necessary for the microprocessor to hold a (readable) 'master copy' of EUROM's scroll map contents at a location in its main memory.

8-bit microprocessors

Although the control bus is optimised for the SCN68000 16-bit microprocessor unit, EUROM will operate with a number of widely differing industry-standard 8, 16 or more-bit microprocessors or microcontrollers (e.g. SCN68008, MAB8051). The interfacing of 8-bit microprocessors to the 16-bit wide display memory is made simple by EUROM's on-chip link-through buffer which provides the microprocessor with bidirectional access to the lower (odd) half of the memory. The link-through buffer is enabled by the buffer-enable signal $\overline{BUFEN}$, and the send/receive direction is controlled by the signal $\overline{S}$/R.

The main data and address paths used in a connected 8-bit microprocessor system are shown in Fig. 22. The interface is similar to that of the 16-bit system but here the display memory does not receive A0 as an address, rather A0 is used as the major enabling signal for $\overline{BUFEN}$ (enables when HIGH).

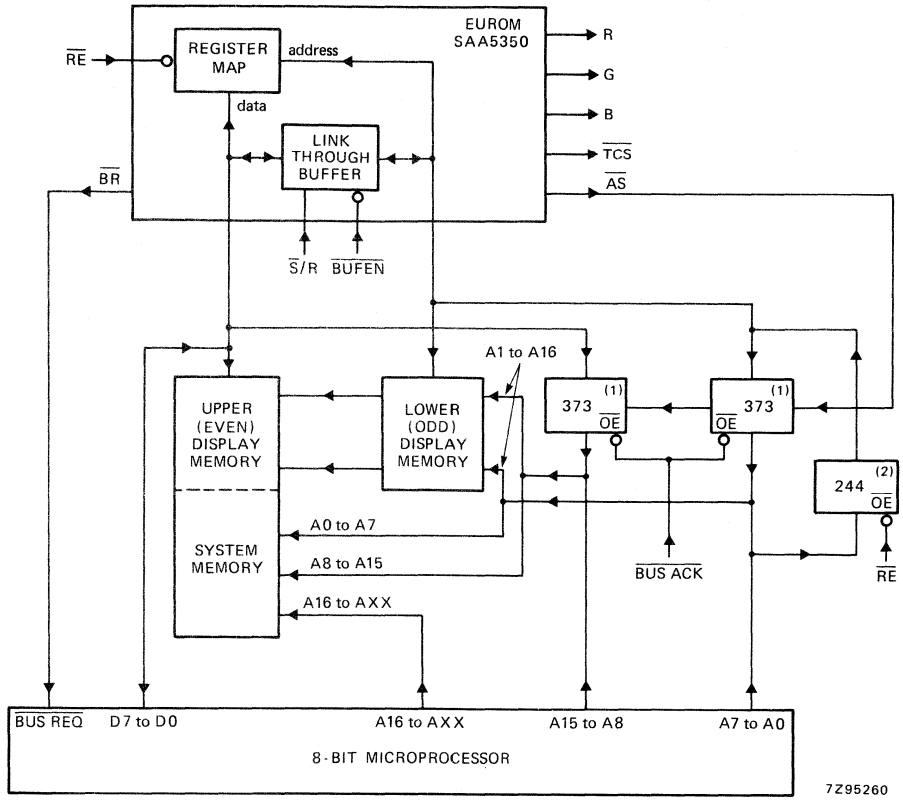
APPLICATION INFORMATION (continued)



(1) 74LS373 octal transparent latch (3-state)
(2) 74LS244 octal buffer (3-state)
(3) 74LS245 octal bus transceiver (3-state)
(4) SCN68000 microprocessor unit

Fig. 21 Connected 16-bit microprocessor system.

DEVELOPMENT DATA



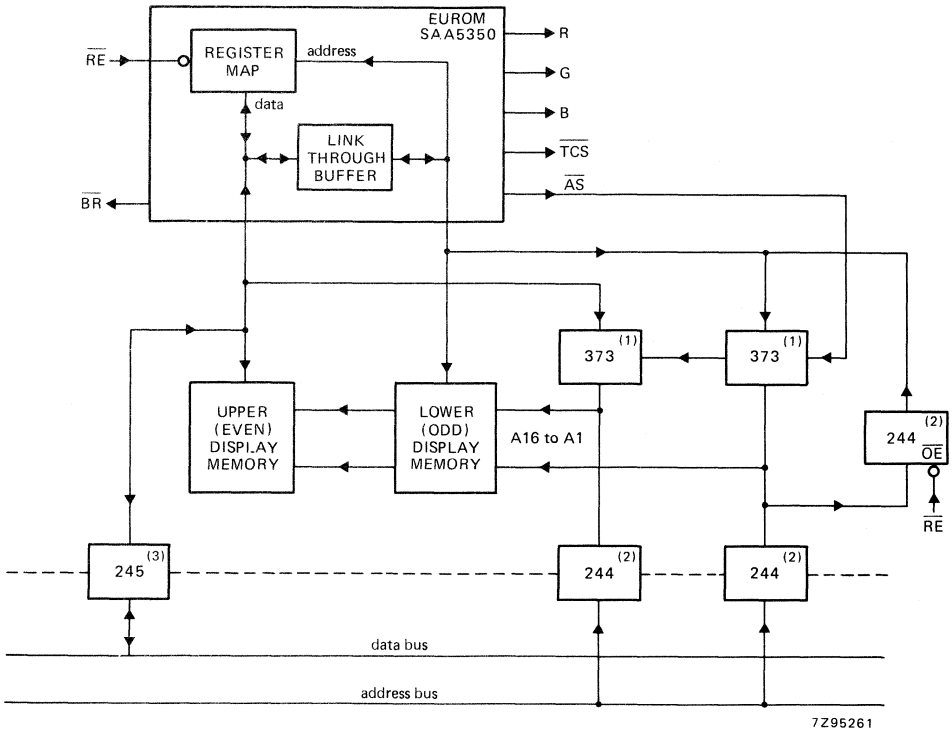
(1) 74LS373 octal transparent latch (3-state)
(2) 74LS244 octal buffer (3-state)

Fig. 22 Connected 8-bit microprocessor system.

APPLICATION INFORMATION (continued)

Disconnected systems

For many applications it may be desirable to disconnect EUROM and the display memory from the microprocessor and its ROM, RAM and other peripherals by using isolating buffers as shown in Fig. 23. The two parts of the system then operate independently and communicate only when the microprocessor accesses EUROM's register map or the display memory.



(1) 74LS373 octal transparent latch (3-state)
(2) 74LS244 octal buffer (3-state)
(3) 75LS245 octal bus transceiver (3-state)

Fig. 23 Disconnected 8-bit system.

Synchronization

Stand-alone mode

As a stand-alone device (e.g. in terminal applications) EUROM can output a composite sync signal ($\overline{TCS}$) to the display timebase IC or to a monitor. Timing is obtained from a 6 MHz on-chip oscillator using an external crystal as shown in Fig. 24.

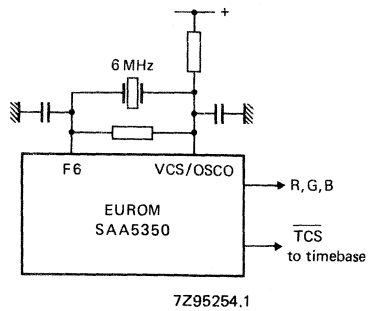


Fig. 24 Stand-alone synchronization mode.

Simple-slave

In the simple-slave mode EUROM synchronizes directly to another device, such as to the $\overline{TCS}$ signal from the SAA5240 European computer-controlled teletext circuit (CCT) or from another EUROM as shown in Fig. 25. EUROM's horizontal counter is reset by the falling edge of $\overline{TCS}$. A dead time of 250 ns is built in to avoid resetting the counter at every tv line and so prevents screen jitter.
Field synchronization is made using EUROM's internal field sync separator.

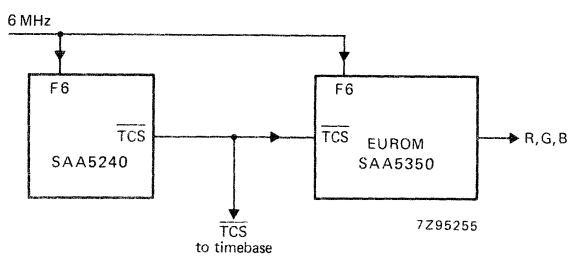


Fig. 25 Simple-slave (direct sync) mode.

APPLICATION INFORMATION (continued)

Synchronization (continued)

Phase-locked slave

The phase-locked slave (indirect sync) mode is shown in Fig. 26. A phase-locked VCO in the SAA5230 teletext video processor provides sync to the timebases. When EUROM is active, its horizontal counter forms part of the phase control loop – a horizontal reference is fed back to the SAA5230 from the SAND output and a vertical reference is generated by feeding separated composite sync to EUROM's field sync separator via the VCS input. In the phase-locked slave mode, the display derived from EUROM can sync with that from a tv source or a local VLP player, thus giving picture-in-text display possiblities.

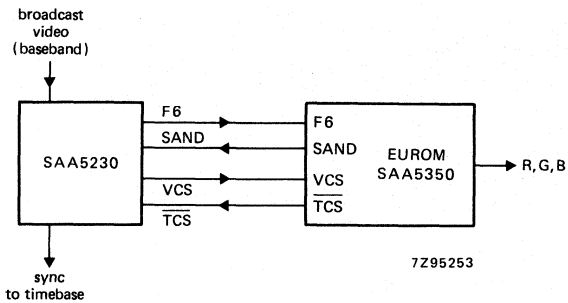


Fig. 26 Phase-locked slave (indirect sync) mode.

Display Character and Graphics Generator (DCGG)

Product Specification

Originally published by Signetics February 1985

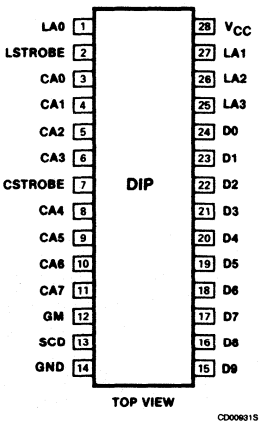
DESCRIPTION

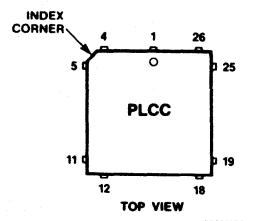
The Signetics SCN2670 Display Character and Graphics Generator (DCGG) is a mask-programmable 11,648-bit line select character generator. It contains 128 10 x 9 characters placed in a 10 x 16 matrix, and has the capability of shifting certain characters, such as j, y, g, p and q, that normally extend below the baseline. Character shifting, previously requiring additional external circuitry, is now accomplished internally by the DCGG; effectively, the 9 active lines are lowered within the matrix to compensate for the character's position.

FEATURES

- **5 x 7 or 7 x 9 character fonts**
- **128 10 x 9 matrix characters**
- **256 graphic characters**
- **Optional thin graphics for forms**
- **Character and line address latches**
- **Internal descend logic**
- **200nsec and 300nsec character select access time versions**
- **Control character output inhibit logic**
- **Static operation — no clocks required**
- **Single 5V power supply**
- **TTL compatible inputs and outputs**

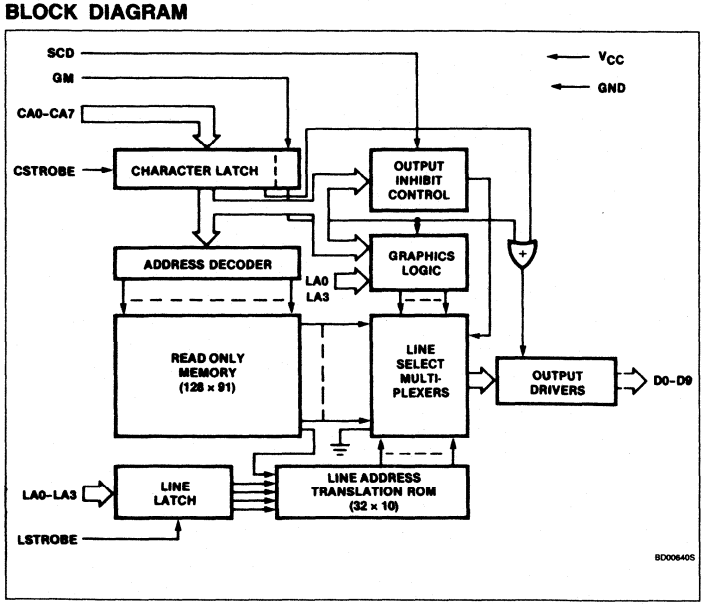
PIN CONFIGURATION





NOTE:
Pin functions same as DIP.

BLOCK DIAGRAM



ORDERING CODE

PACKAGES	CHARACTER FONT SIZE	V_{CC} = 5V ±5%, T_A = 0°C to 70°C	
		t_{CA} = 200ns	t_{CA} = 300ns
Ceramic DIP Plastic DIP Plastic LCC	5 X 7	SCN2670CC2I28 SCN2670CC2N28 SCN2670CC2A28	SCN2670CC3I28 SCN2670CC3N28 SCN2670CC3A28
Ceramic DIP Plastic DIP Plastic LCC	7 X 9	SCN2670BC2I28 SCN2670BC2N28 SCN2670BC2A28	SCN2670BC3I28 SCN2670BC3N28 SCN2670BC3A28

PIN DESCRIPTION

MNEMONIC	PIN NO.	TYPE	NAME AND FUNCTION
CA0 – CA7	3 – 6, 8 – 11	I	**Character Address:** Eight bit code specifies the character or graphic pattern for which matrix data is to be supplied. In character mode (GM = 0), CA0 through CA6 select one of the 128 ROM-defined characters and CA7 is a chip enable. The outputs are active when CA7 = 1 and are three-stated when CA7 = 0. In graphics mode (GM = 1), the outputs are always active and CA0 through CA7 select one of 256 possible graphic patterns to be output.
CSTROBE	7	I	**Character Strobe:** Used to store the character address (CA0 through CA7) and graphics mode (GM) inputs into the character latch. Data is latched on the negative going edge of CSTROBE. Can be connected to the $\overline{CCLK}$ of the Signetics attributes controller chips.
GM	12	I	**Graphics Mode:** GM = 0 (low) selects character mode; GM = 1 (high) selects graphics mode.
LA0 – LA3	1, 25 – 27	I	**Line Address:** In character mode, selects one of the 16 lines of matrix data for the selected character to appear at the 10 outputs. LA0 is the LSB and LA3 is the MSB. The input codes which cause each of the nine lines of character data to be output are specified as part of the programming data for both non-shifted and shifted fonts. Cycling through the nine specified counts at the LA0 through LA3 inputs cause successive lines of data to be output on D0 through D9. The 7 non-specified codes for both non-shifted and shifted characters cause blanks (logic zeros) to be output. In graphics mode, the line address gates the latched graphics data directly to the outputs.
LSTROBE	2	I	**Line Strobe:** Used to store the line address data (LA0 through LA3) in the line address latch. Data is latched on the negative going edge of LSTROBE. Can be connected to the BLANK output of the Signetics video display controller chips.
SCD	13	I	**Selected Character Disable:** In character mode, a high level at this input causes all outputs (regardless of line address) to be blanks (zeros) for characters for which CA6 and CA5 are both 0. A low level input selects normal operation. Inoperative in the graphics mode.
D9 – D0	15 – 24	O	**Data Outputs:** Provide the data for the specified character and line.
V_{CC}	28	I	+ 5V power supply.
GND	14	I	Ground.

Seven bits of an 8-bit address code are used to select 1 of the 128 available characters. The eighth bit functions as a chip enable signal. Each character is defined by a pattern of logic 1's and 0's stored in a 10 x 9 matrix. When a specific 4-bit binary line address code is applied, a word of 10 parallel bits appears at the output. The lines can be sequentially selected, providing a 9-word sequence of 10 parallel bits per word for each character selected by the address inputs. As the line address inputs are sequentially addressed, the device will automatically place the 10 x 9 character in 1 of 2 pre-programmed positions on the 16-line matrix with the positions defined by the 4-line address inputs. One or more of the 10 parallel outputs can be used as control signals to selectively enable functions such as half-dot shift, color selection, etc.

The SCN2670 DCGG includes latches to store the character address and line address data. A control input to inhibit character data output for certain groups of characters is also provided. The SCN2670 also includes a graphics capability, wherein the 8-bit character code is translated directly into 256 possible user programmable graphic patterns. Thus, the DCGG can generate data for 384 distinct patterns, of which 128 are defined by the mask programmable ROM. See figure 1 for a typical applications display.

FUNCTIONAL DESCRIPTION

The DCGG consists of nine major sections. Line and character codes are strobed into the line and character latches. The character latch outputs are presented to the three sources of data; the ROM through an address decoder, the graphics logic, and the output inhibit control. The output inhibit control (together with the SCD input) suppresses the ROM data for selected character codes. The outputs from the line latch drive the line address translation ROM which maps the character ROM data onto 9 of 16 line positions. Finally, the line select multiplexers route the ROM or graphics data to the output drivers on D0 through D9.

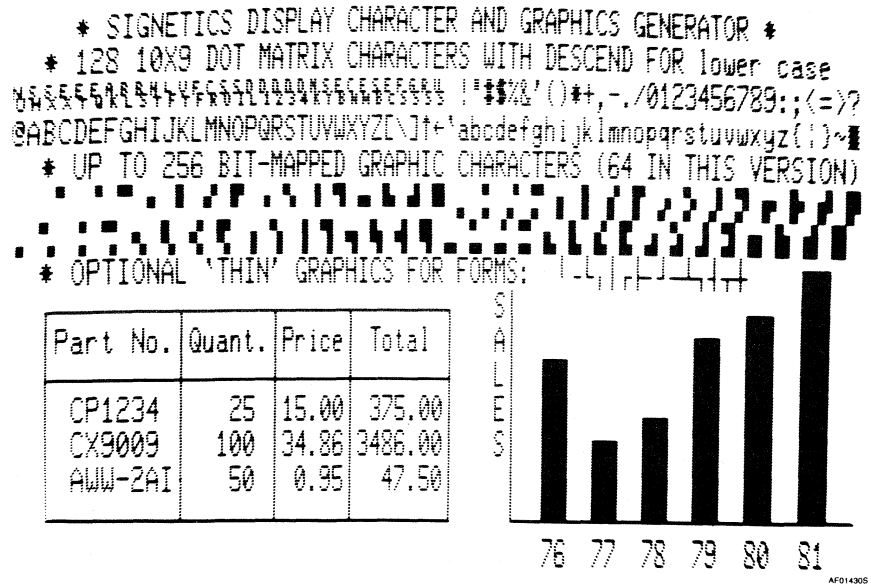


Figure 1. Typical Application Using The 7 x 9 Character Font

Character Latch

The character latch is a 9-bit edge triggered latch used to store the character address (CA0 through CA7) and graphics mode (GM) inputs. The data is stored on the falling edge of CSTROBE. Seven latched addresses (CA0 through CA6) are inputs to the ROM character address decoder. In character mode (GM = 0), CA7 operates as a chip enable. The output drivers are enabled when CA7 = 1 and are three-stated when CA7 = 0. In graphics mode (GM = 1), the output drivers are always enabled and the CA0 through CA7 outputs of the latch are used to generate graphic symbols.

Character Address Decoder

This circuit decodes the 7-bit character address from the character latch to select one of the 128 character fonts stored in the ROM section of the DCGG.

Read Only Memory

The 11,648-bit ROM stores the fonts for the 128 matrix-defined characters. The data for each character consists of 91 bits. Ninety bits represent the 10X9 matrix and one bit specifies whether the character data is output at the normal (unshifted) lines or at the descended (shifted) lines. The 90 data bit outputs are supplied to the line select multiplexers. The descend control bit is an input to the line address translation ROM.

Graphics Logic

When the GM input is zero (low), the DCGG operates in the character mode. When it is one (high), it operates in the graphics mode. In graphics mode, output data is generated by the graphics logic instead of the ROM. The graphics logic maps the latched character address (CA0 through CA7) to the outputs (D0 through D9) as a function of line address (LA0 through LA3). For any particular line address value, two of the CA bits are output: CA0, CA2, CA4 or CA6 is output on D0 through D4 and CA1, CA3, CA5 or CA7 is output on D5 through D9. The outputs are paired: When CA0 is output on D0 through D4, CA1 is output on D5 through D9 and likewise for CA2 – CA3, CA4 – CA5 and CA6 – CA7.

A ROM within the graphics logic allows the specific line numbers for which each pair of bits is output to be specified by the customer. Figure 2 illustrates the general format for graphics symbols and an example where CA7 through CA0 = H'65'. The outputs from the graphics logic go to the line select multiplexers. The multiplexers route the graphic symbol data to the outputs when GM = 1.

Thin Graphics Option

As a customer specified option, 16 of the possible graphic codes (H'80' to H'8F') may be used to generate the special graphic characters illustrated in figure 3. For each of these characters, the vertical component appears on the D4 output. The horizontal component occurs on L_H which is specified by the customer. The vertical components specified by CA0 and CA2 are output for line addresses zero through L_H and L_H through fifteen, respectively.

Line Select Multiplexers

The ten line select multiplexers select ROM data as specified by the line address translation ROM when GM = 0, or graphics data when GM = 1. The inputs to each multiplexer are the nine line outputs from the ROM, an output from the graphics logic and a logic zero (ground).

Output Drivers

Ten output drivers with 3-state capability serve as buffers between the line select multiplexers and external logic. The 3-state control input to these drivers is supplied from the CA7 latch when GM = 0. When GM = 1, the outputs are always active.

Output Inhibit Control

The output inhibit control logic operates only if GM = 0. It causes the output of the line select multiplexers to be logic zero if the SCD input is high and CA6 and CA5 of the latched character address are 00. If the SCD input is low, normal operation occurs. (This feature is useful in ASCII coded applications to selec-

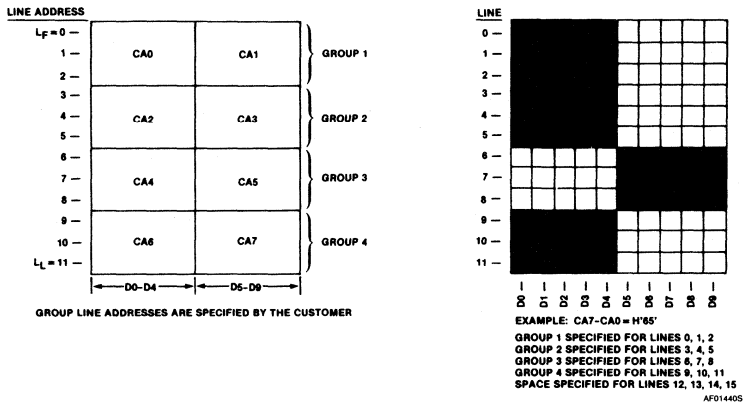


Figure 2. Graphics Symbols — General Format

Table 1. MEMORY MAP FORMAT

2732 Byte Address	Data
0	0 = No thin graphics 1 = Thin graphics
1	Line address for the horizontal segment of the thin line graphics fonts (set to '0' if byte 0 = '0').
2 – 17 (2 – 11H)	Specify group number (1 – 4) to be output at the corresponding line address (0 – F). A '0' specifies no data for that address.
18(12H)	Line address at which the first NONSHIFTED font data appears.
19(13H)	second
20(14H)	third
21(15H)	fourth
22(16H)	fifth
23(17H)	sixth
24(18H)	seventh
25(19H)	eighth
26(1AH)	ninth
27(1BH)	Line address at which the first SHIFTED font data appears.
28(1CH)	second
29(1DH)	third
30(1EH)	fourth
31(1FH)	fifth
32(20H)	sixth
33(21H)	seventh
34(22H)	eighth
35(23H)	ninth
36(24H)	Font truth table printout format 0 = Left to right printing of horizontal data bits D0 – D9 1 = Left to right printing of horizontal data bits D9 – D0
37(25H)	Font truth table printout format 0 = Top to bottom printing of vertical line addresses (0 – FH) 1 = Top to bottom printing of vertical line addresses (F – 0H)
38 – 57 (26 – 39H)	Character 0 data
58 – 77 (3A – 4DH)	Character 1 data
. . .	. . .
2578 – 2597 (A12 – A25H)	Character 128 (7F H) data

tively disable character generation for non-displayable characters such as line feed, carriage return, etc.)

Line Address Latch

The line address latch is a 4-bit latch used to store the line address (LA0 – LA3). The data is stored on the negative edge of the LSTROBE input.

Line Address Translation ROM

This 32 x 10 ROM translates the 5-bit code consisting of the 4 outputs from the line address latch and the descend control bit from the ROM into a 1-of-10 code for the line select multiplexers. Programming information provided by the customer specifies the address which selects each line of ROM data for both shifted and non-shifted characters. Thus, there are nine line addresses which select ROM data for unshifted characters and nine addresses for shifted characters. These combinations are usually specified by the customer in either ascending or descending order. For the remaining 14 codes (7 each for unshifted and shifted characters), the translation ROM forces zeros at the outputs of the line select multiplexers.

This circuitry only operates if GM = 0. When GM = 1, the line select multiplexers are forced to select the outputs from the graphics logic.

Figure 4 shows an example of data outputs where the customer has specified line 1 as the first line for unshifted characters, line 4 as the first line for shifted characters and line address combinations in ascending order.

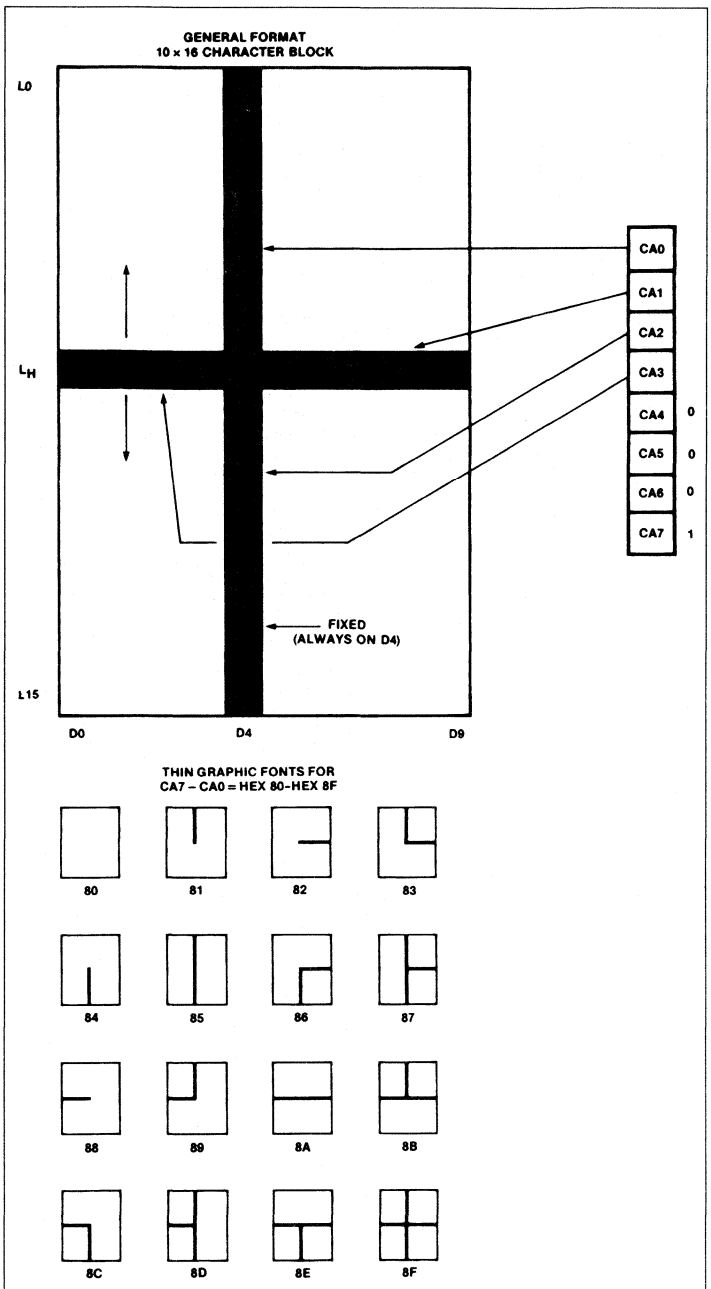


Figure 3. Special Graphic Characters

5 x 7 Design Note

The 5 x 7 character fonts of the 2670C have been centered in the 10 x 9 character fields. Because of this centering, only the D1 through D8 data bus pins (pins 23 through 16) are to be used for the character data. D1 can be connected to the least significant bit of the Signetics Video Attributes Controller data bus (D0), D2 of the 2670C can be connected to D1 of the attributes controller, etc. The D0 and D9 pins will always be read as zero and will not be connected.

CUSTOM PATTERN PROGRAMMING INSTRUCTIONS

Signetics requires that the customer supply them with a 2732 EPROM containing the necessary data for specifying a custom version of the SCN2670. The 2732 will contain the data to be stored in the ROM array, the programmable line address translation ROM, thin graphics option and the graphics line font translation ROM. Signetics will no longer accept card decks.

The memory map format for programming the 2732 is contained in table 1.

Bytes 0 and 1 relate to the thin line graphics option. Thin line graphics consists of two components: a vertical and a horizontal segment. The vertical segment always occurs at the D4 output. The location of the horizontal segment (line addresses 0 – 15) is specified by the customer in byte 1.

Bytes 2 through 17 correspond to how the graphics blocks are created from the 16 horizontal addresses. A '0' specifies no data for that address. Figure 5 shows an example of how the character field can be divided into the graphics blocks. The line addresses can be divided into a maximum of four groups.

Since line addresses A through F are not used, byte addresses 12 through 17 are set to '0'. Line addresses 0 and 1 are part of group 1, so byte addresses 2 and 3 are set equal to '1'. Line addresses 2 through 4 are in group 2, so byte addresses 4 through 6 are set equal to '2'. The same procedure is followed for the remaining line addresses. The group numbering assignments must be sequential (i.e., line address 1 cannot be in group 2 if line addresses 0 and 2 are in group 1).

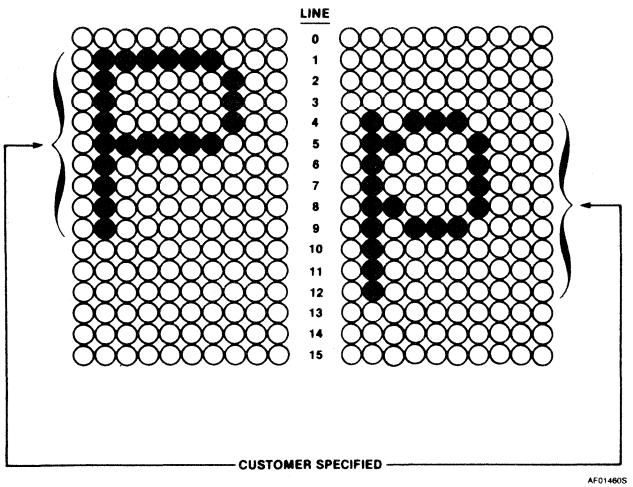


Figure 4. Customer Specified Example

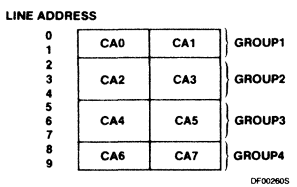


Figure 5. Character Field Divisions

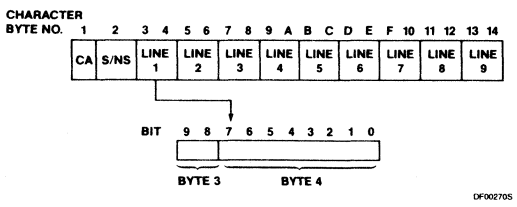


Figure 6. Data Format

Byte Addresses 18 through 26 are used to indicate which line address the nonshifted character fonts begin and end. The first line of the character font must occur from line address 0 to 7. If the character font begins at line address 2, the byte addresses are set up as follows:

BYTE ADDRESS	DATA
18	2
19	3
20	4
21	5
22	6
23	7
24	8
25	9
26	10

Byte Addresses 27 through 35 are used to indicate which line address the shifted character fonts begin and end. Again the first line of the character font must occur from line addresses 0 through 7. The data is entered the same as for the nonshifted character fonts above.

Bytes 36 and 37 are only used for printouts of the ROM codes by Signetics.

The remaining byte addresses (38 – 2597) contain the data for the 128 characters. Each character consists of 20 bytes. The format is contained in figure 6.

Character Byte 1 (CA) contains the character address (CA6 – CA0).

For character byte 2, if S/NS = '1', the character font will begin at the shifted location as specified by byte addresses 27 – 35. If S/NS = '0', then the character font will begin at the nonshifted location as specified by byte addresses 18 – 26.

The remaining character bytes define the nine line character font. The bytes are paired as each font line requires 10 bits of information to be output on lines D9 – D0 (see figure 6). Remember, when coding each line, D9 is the rightmost bit of the character line, and D0 is the leftmost bit.

Printouts

Signetics will submit the following printouts to the customer for approval:

- A repeat of all customer information.
- A separate font drawing for each of the 128 ROM characters and 256 graphics fonts. The font drawings are positioned on a 10 x 16 matrix as specified by the customer's translation data.

Customer Example

A question mark has a character address of 3F hex. The character code for it is as follows:

```
L0    . . X X X X X . . .
L1    . X . . . . . X . .
L2    . X . . . . . X . .
L3    . . . . . . . X . .
L4    . . . . . X X . . .
L5    . . . . X . . . . .
L6    . . . . X . . . . .
L7    . . . . . . . . . .
L8    . . . . X . . . . .
bit   0 1 2 3 4 5 6 7 8 9
```

Character

Byte 1:	3F			
2:	00			(nonshifted)
3:	00	4:	C7	(Line 1)
5:	00	6:	28	(Line 2)
7:	00	8:	28	(Line 3)
9:	00	A:	08	(Line 4)
B:	00	C:	06	(Line 5)
D:	00	E:	01	(Line 6)
F:	00	10:	01	(Line 7)
11:	00	12:	00	(Line 8)
13:	00	14:	01	(Line 9)

We recommend that the character fonts be created with respect to the center of the 10 x 16 character field (data bit D4) for optimum symmetry, especially if thin line graphics is implemented as the vertical segment is always located at bit D4.

ABSOLUTE MAXIMUM RATINGS [1]

PARAMETER	RATING	UNIT
Supply voltage	6.0	V
Operating ambient temperature[2]	0 to +70	°C
Storage temperature	−65 to +150	°C
All voltages with respect to ground[3]	−0.3 to +6.0	V

NOTES:

1. Stresses above those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other condition above those indicated in the operation section of this specification is not implied.
2. For operating at elevated temperatures, the device must be derated based on + 150°C maximum junction temperature and thermal resistance of 60°C/W junction to ambient (ceramic package).
3. This product includes circuitry specifically designed for the protection of its internal devices from the damaging effects of excessive static charge. Nonetheless, it is suggested that conventional precautions be taken to avoid applying any voltages larger than the maxima.

DC ELECTRICAL CHARACTERISTICS $T_A = 0°C$ to $70°C$, $V_{CC} = 5.0V \pm 5\%$ [1,2,3]

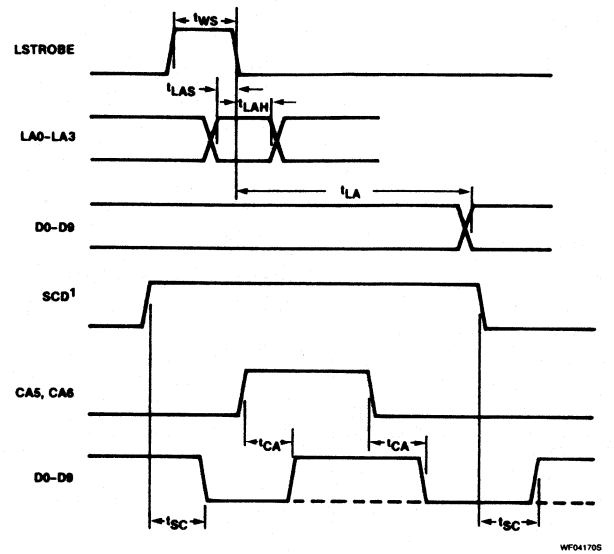
PARAMETER		TEST CONDITIONS	LIMITS			UNIT
			Min	Typ	Max	
V_{IL}	Input low voltage				0.8	V
V_{IH}	Input high voltage		2.0			V
V_{OL}	Output low voltage	$I_O = 1.6mA$	0		0.4	V
V_{OH}	Output high voltage	$I_O = -100\mu A$	2.4		V_{CC}	V
I_{IL}	Input leakage current	$V_{IN} = 0$ to 4.25V			10	μA
I_{OL}	Output leakage current	$V_O = 0.4$ to 4V	−10		+10	μA
I_{CC}	Supply current	$V_{CC} = 5.25V$		35	80	mA
C_{IN}	Input capacitance	All other pins grounded			10	pF
C_{OUT}	Output capacitance				15	pF

AC ELECTRICAL CHARACTERISTICS T_A = 0°C to 70°C, V_{CC} = 5.0V±5% [1,2,3,4]

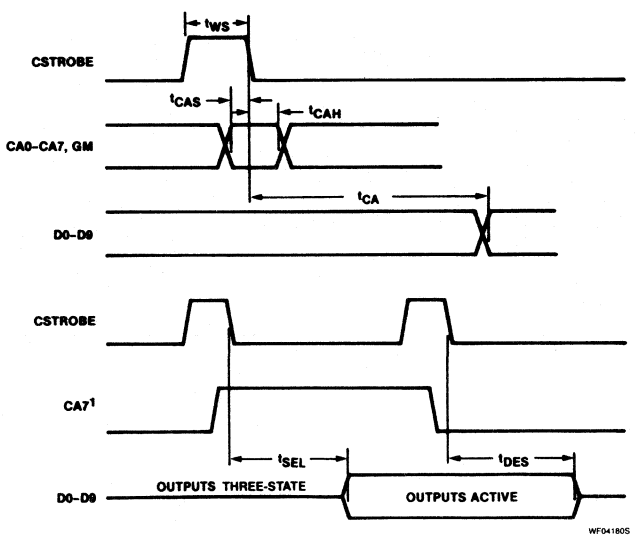
PARAMETER		LIMITS				UNIT
		300ns		200ns		
		Min	Max	Min	Max	
t_{WS}	Strobe pulse width	100		100		ns
t_{LAS}	Line address set-up	50		50		ns
t_{LAH}	Line address hold	25		25		ns
t_{CAS}	Character address set-up	25		15		ns
t_{CAH}	Character address hold	25		15		ns
t_{CA}	Character select access	30	300	30	200	ns
t_{LA}	Line select access	30	500	30	350	ns
t_{SEL}	Chip select delay		250		150	ns
t_{DES}	Chip deselect delay		200		125	ns
t_{SC}	Special character blank/unblank time		300		200	ns

NOTES:
1. Parameters are valid over operating temperature range unless otherwise specified.
2. All voltage measurements are referenced to ground. All time measurements are at the 0.8V or 2.0V level for inputs and outputs. Input levels are 0V and 2.4V.
3. Typical values are at + 25°C, typical supply voltages and typical processing parameters.
4. Test conditions: C_L = 100pF and 1 TTL load.

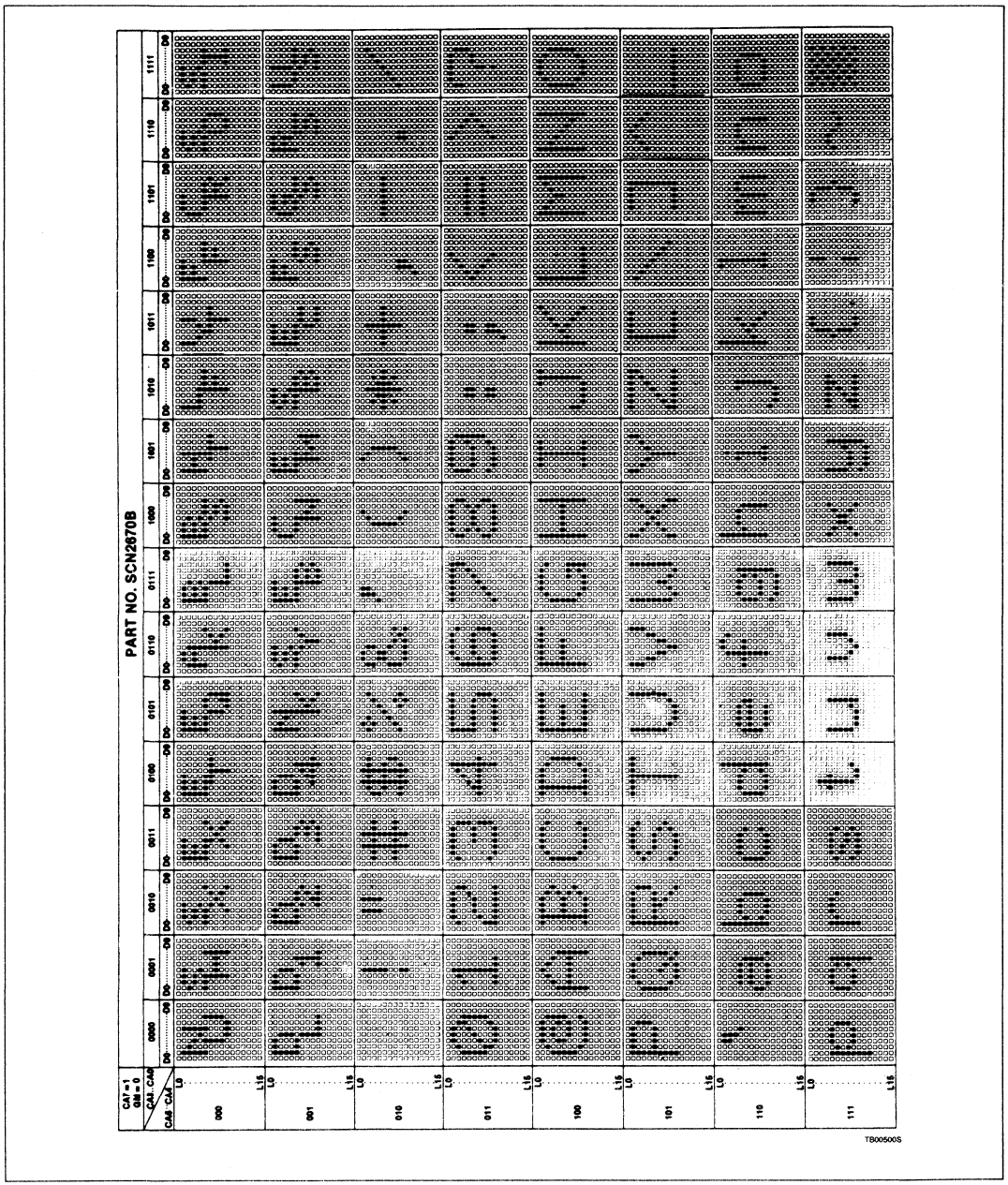
TIMING DIAGRAMS

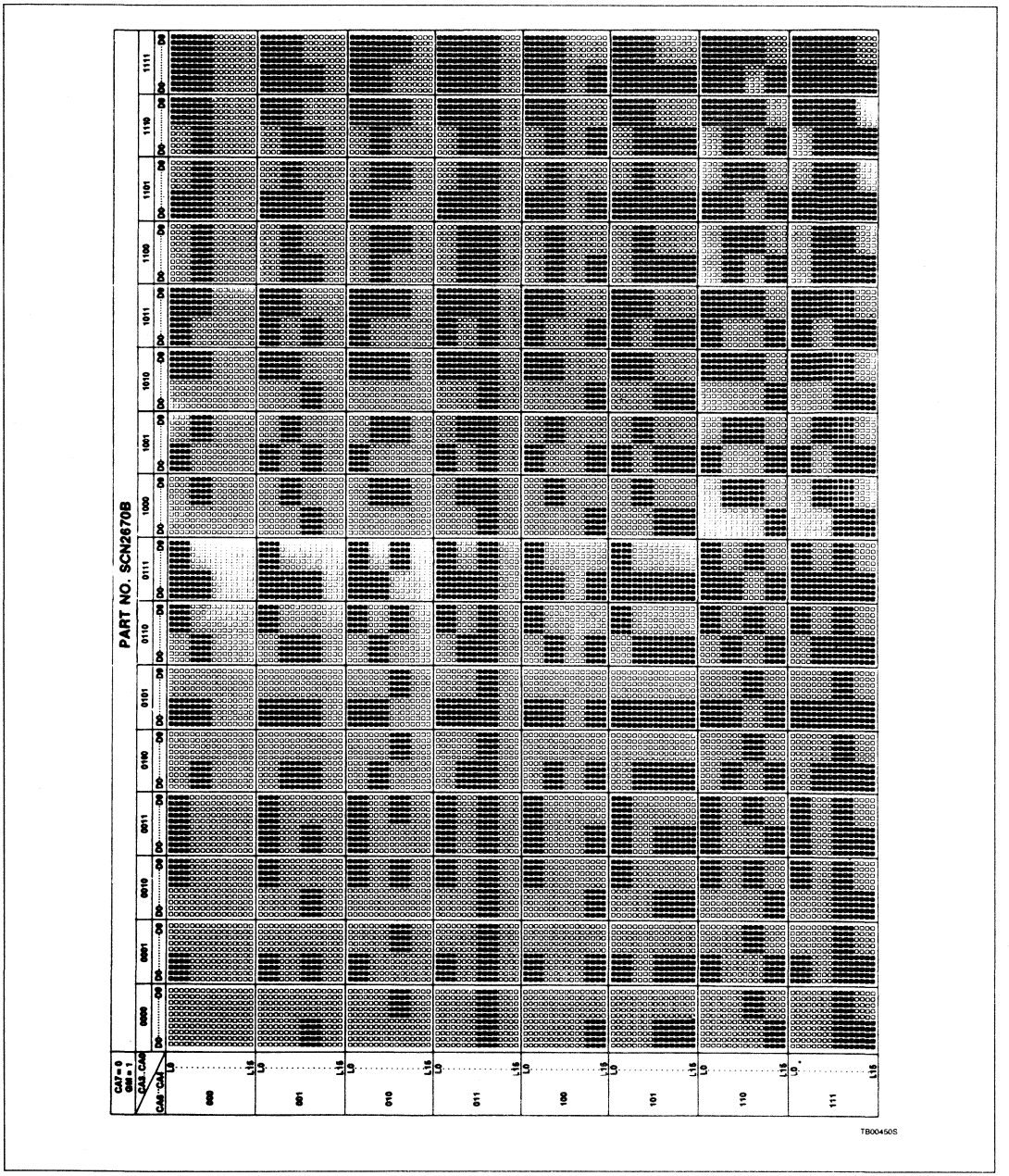


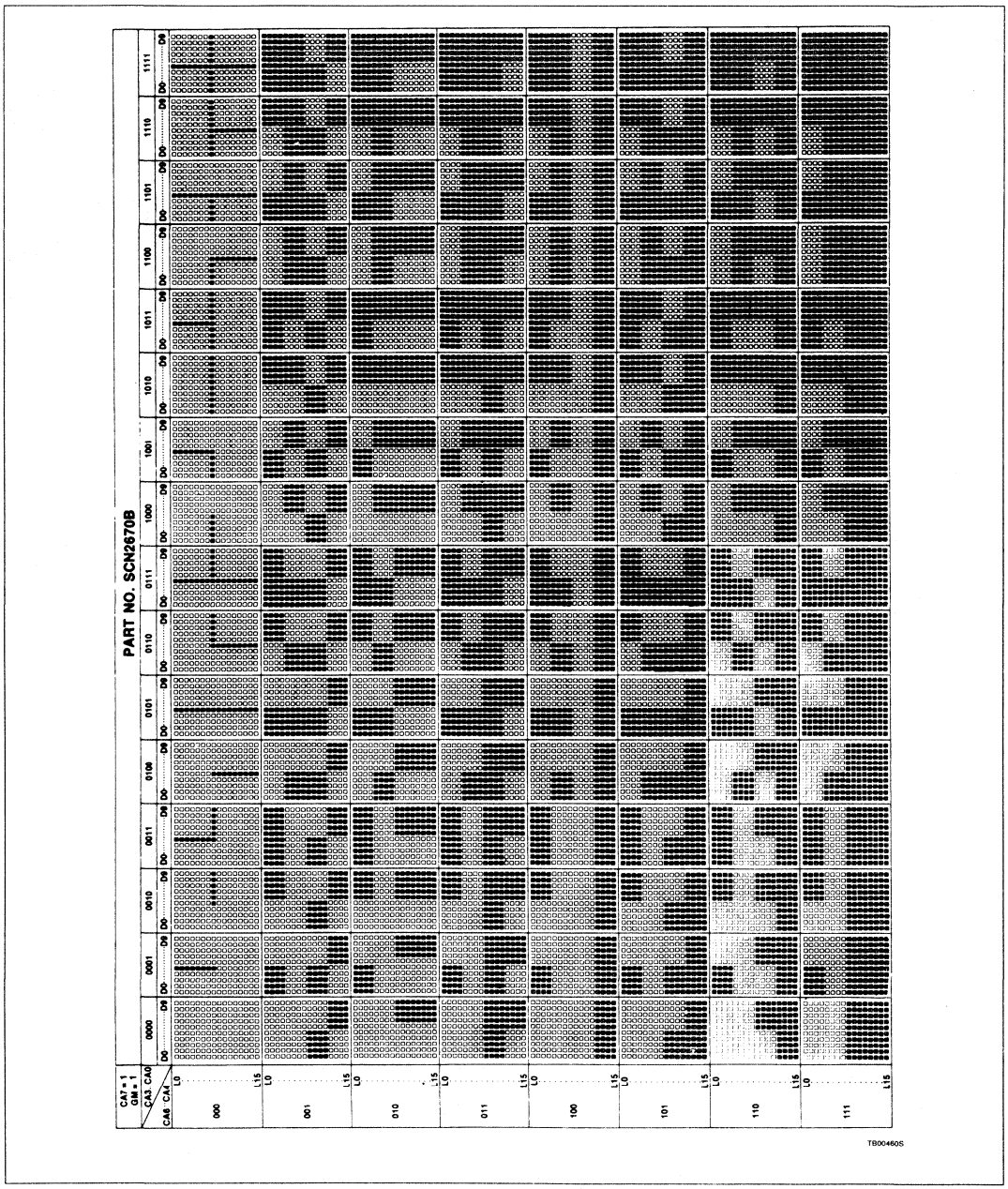
NOTE:
1. When GM = 1, SCD input is inactive.

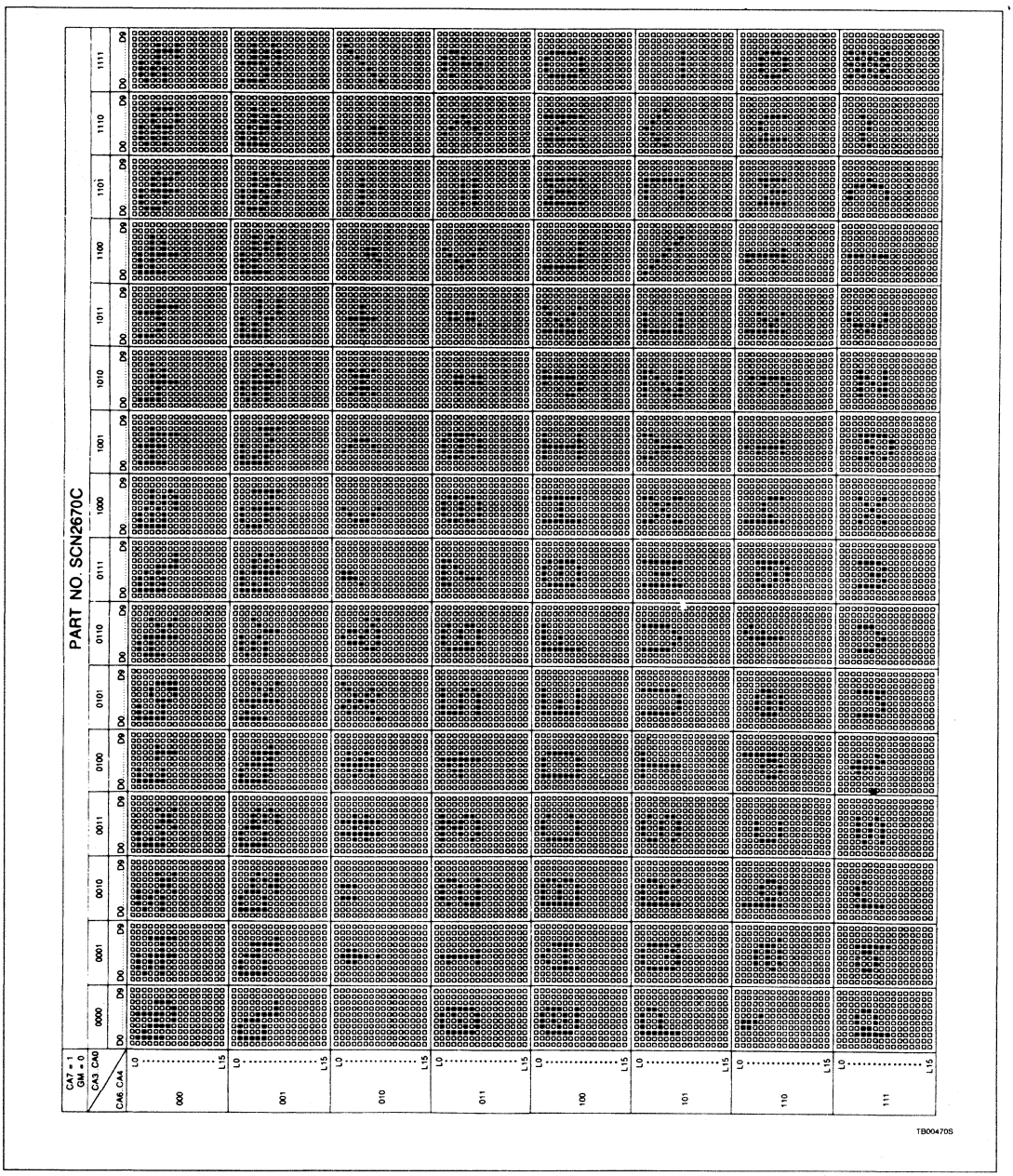


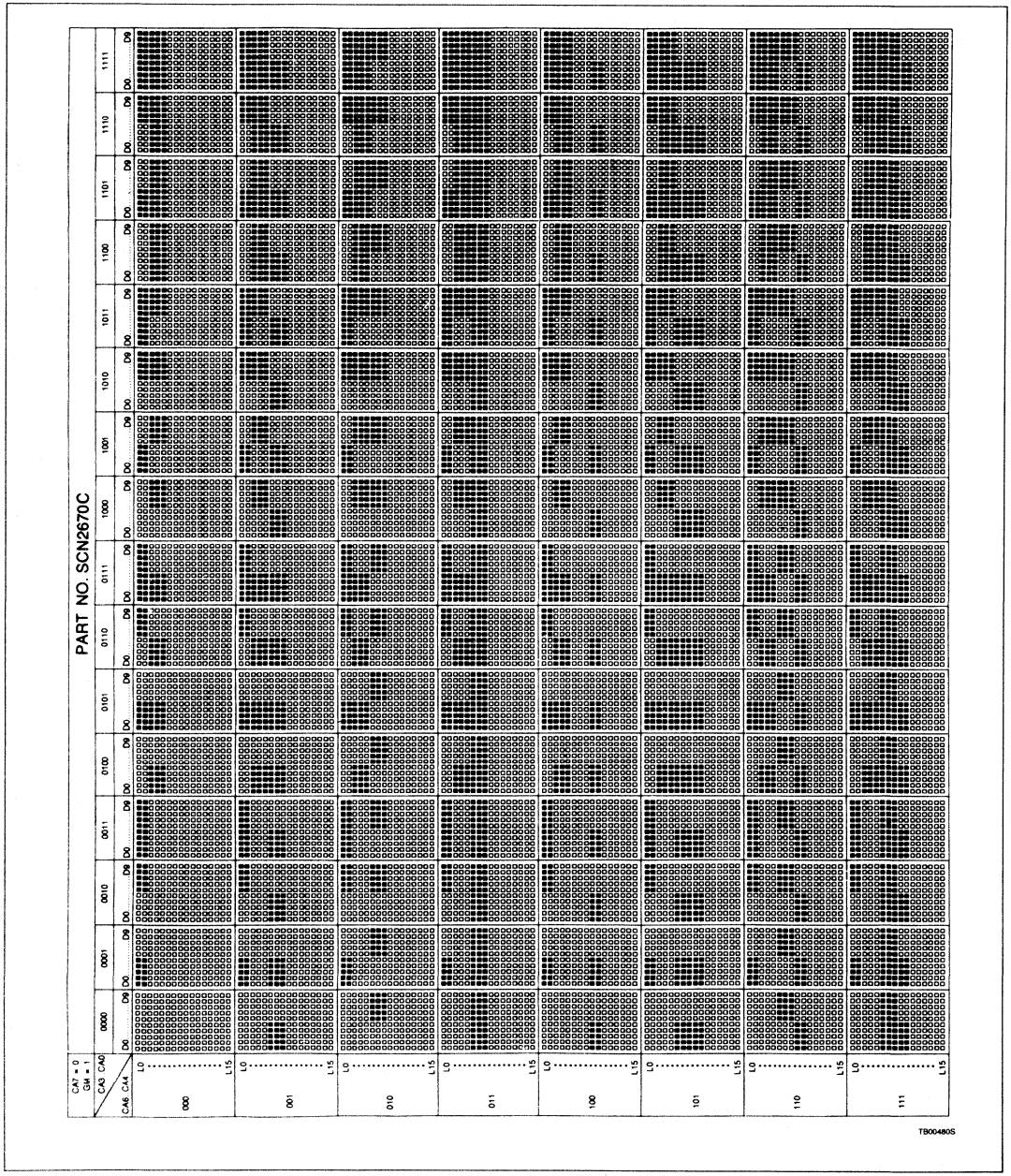
NOTE:
1. CA7 operates as output enable only in character mode (GM = 0).

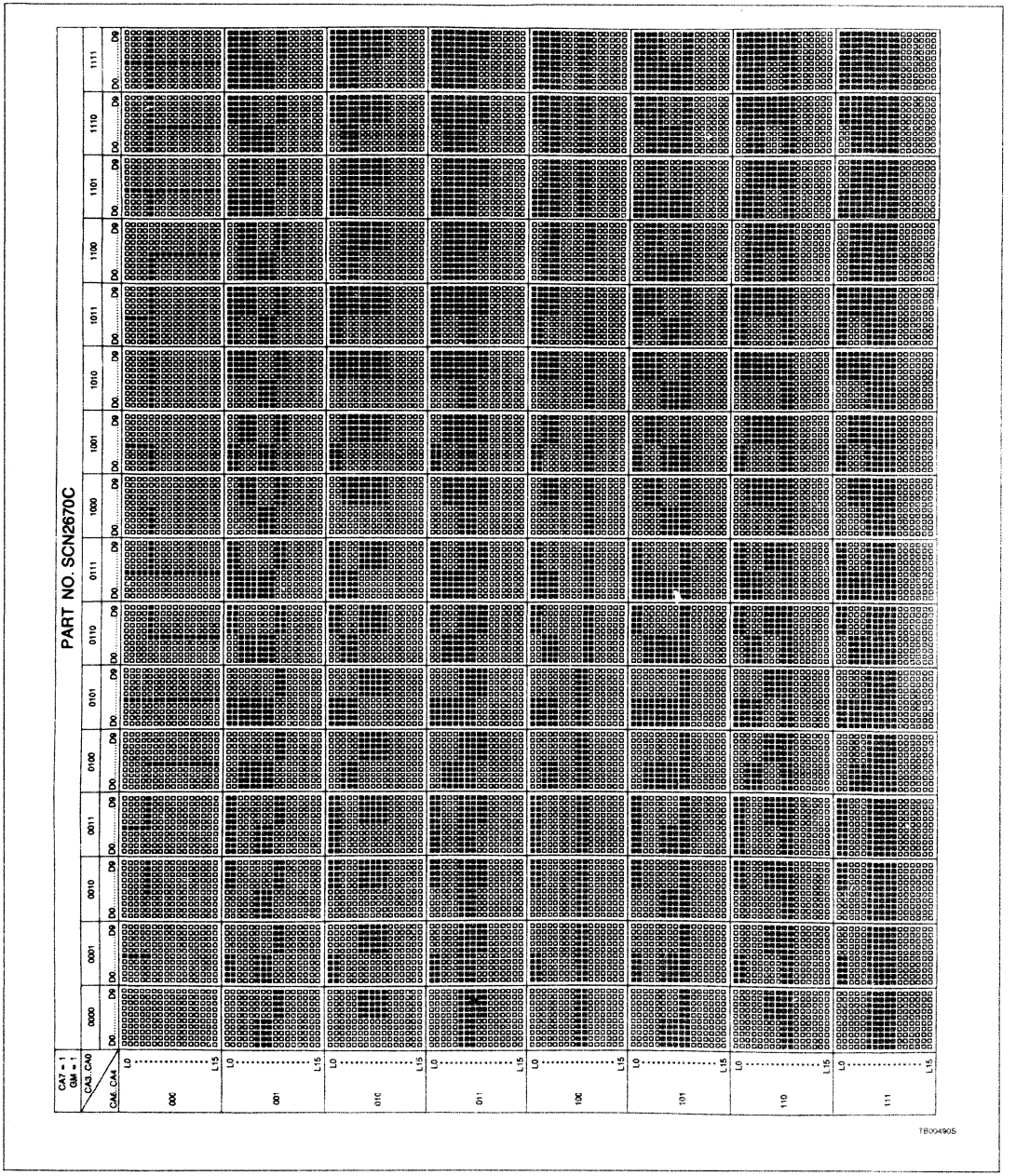

PART NO. SCN2670B
CAT = 1
DM = 0
CAT, CAB
CAB, CAT
L0
L15
000
001
010
011
100
101
110
111
0000
0001
0010
0011
0100
0101
0110
0111
1000
1001
1010
1011
1100
1101
1110
1111
D0
D1
D2
D3
TB005005


PART NO. SCN2670B
CAT = 0
GM = 1
CA3 CA2 CA1 CA0
0000
0001
0010
0011
0100
0101
0110
0111
1000
1001
1010
1011
1100
1101
1110
1111
D0
D1
D2
D3
L0
L15
000
001
010
011
100
101
110
111
TB00450S


PART NO. SCN2670B
CA7 = 1
GM = 1
CA3 CA0
CA6 CA5
0000
0001
0010
0011
0100
0101
0110
0111
1000
1001
1010
1011
1100
1101
1110
1111
D0
D6
L0
L15
000
001
010
011
100
101
110
111
TB00460S


PART NO. SCN2670C
CAT = 1
CM = 0
CA3 CA0
CA6 CA4
0000
0001
0010
0011
0100
0101
0110
0111
1000
1001
1010
1011
1100
1101
1110
1111
D0
D9
L0
L15
000
001
010
011
100
101
110
111
TB00470S


PART NO. SCN2670C
CA7 = 0
GM = 1
CA5 CA0
CA6 CA4
0000
0001
0010
0011
0100
0101
0110
0111
1000
1001
1010
1011
1100
1101
1110
1111
D0
D9
L0
L15
000
001
010
011
100
101
110
111
TB00480S


PART NO. SCN2670C
CA7 = 1
GM = 1
CA5, CA0
CA6, CA4
0000
0001
0010
0011
0100
0101
0110
0111
1000
1001
1010
1011
1100
1101
1110
1111
D0
D6
L0
L15
000
001
010
011
100
101
110
111
TB004905

Programmable Keyboard and Communications Controller (PKCC)

Product Specification

Originally published by Signetics February 1985

DESCRIPTION

The Signetics SCN2671 Programmable Keyboard and Communications Controller (PKCC) is an MOS LSI device which provides a versatile keyboard encoder and an independent full duplex asynchronous communications controller. It is intended for use in microprocessor based systems and provides an eight bit data bus interface.

The keyboard encoder handles the scanning, debounce, and encoding of a keyboard matrix with a maximum of 128 keys. It provides four levels of key encoding corresponding to the separate SHIFT and CONTROL input combinations. Four keyboard rollover modes can be programmed including provisions for up to 16 latched keys. Control outputs are provided for interfacing with contact or capacitive keyboards. An eight bit keyboard status register provides status information to the CPU.

The receiver section of the communications controller accepts serial data from the RxD pin and converts it to parallel data characters. Simultaneously, the transmitter section accepts parallel data from the data bus and outputs serialized data onto the TxD pin. Received data is checked for parity and framing errors, and break conditions are flagged. Character lengths can be programmed as 5, 6, 7, or 8 bits not including parity, start or stop bits. An internal baud rate generator (BRG) with 16 divider ratios can be used to derive the receive and/or transmit clocks. The BRG can accept an external clock or operate directly from a crystal. An eight bit communications status register provides status information to the CPU.

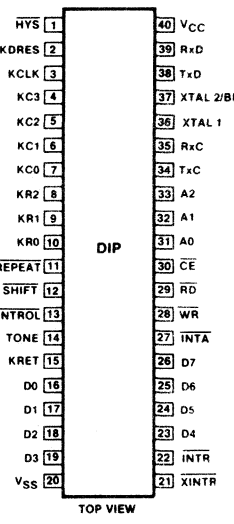
FEATURES

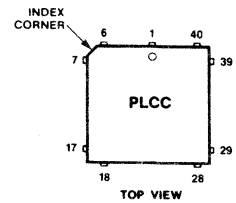
- **Keyboard Interface**
  - **Contact or capacitive keyboard**
  - **Up to 128 keys on an 8 x 16 matrix**
  - **Encoded or unencoded operation**
  - **Four code levels per key**
  - **Latched key option–separate depress and release codes**
  - **Programmable scan rate and debounce time**
  - **Programmable rollover modes**
  - **Programmable auto-repeat for selected keys**
  - **Tone output–two frequencies**
- **Asynchronous communication interface**
  - **Internal baud rate generator–16 rates**
  - **Full duplex operation**
  - **Detection of start and end of break**
  - **Programmable break generation**
  - **Programmable character parameters**
  - **Auto-echo and maintenance loopback modes**
- **Polled or interrupt operation**
- **Interrupt priority controller and vector generator**
- **Operates directly from crystal or external clocks**
- **TTL compatible**
- **Single + 5 volt power supply**

APPLICATIONS

- **CRT terminals**
- **Hard copy terminals**
- **Word processing systems**
- **Data entry terminals**
- **Small business computers**

PIN CONFIGURATION



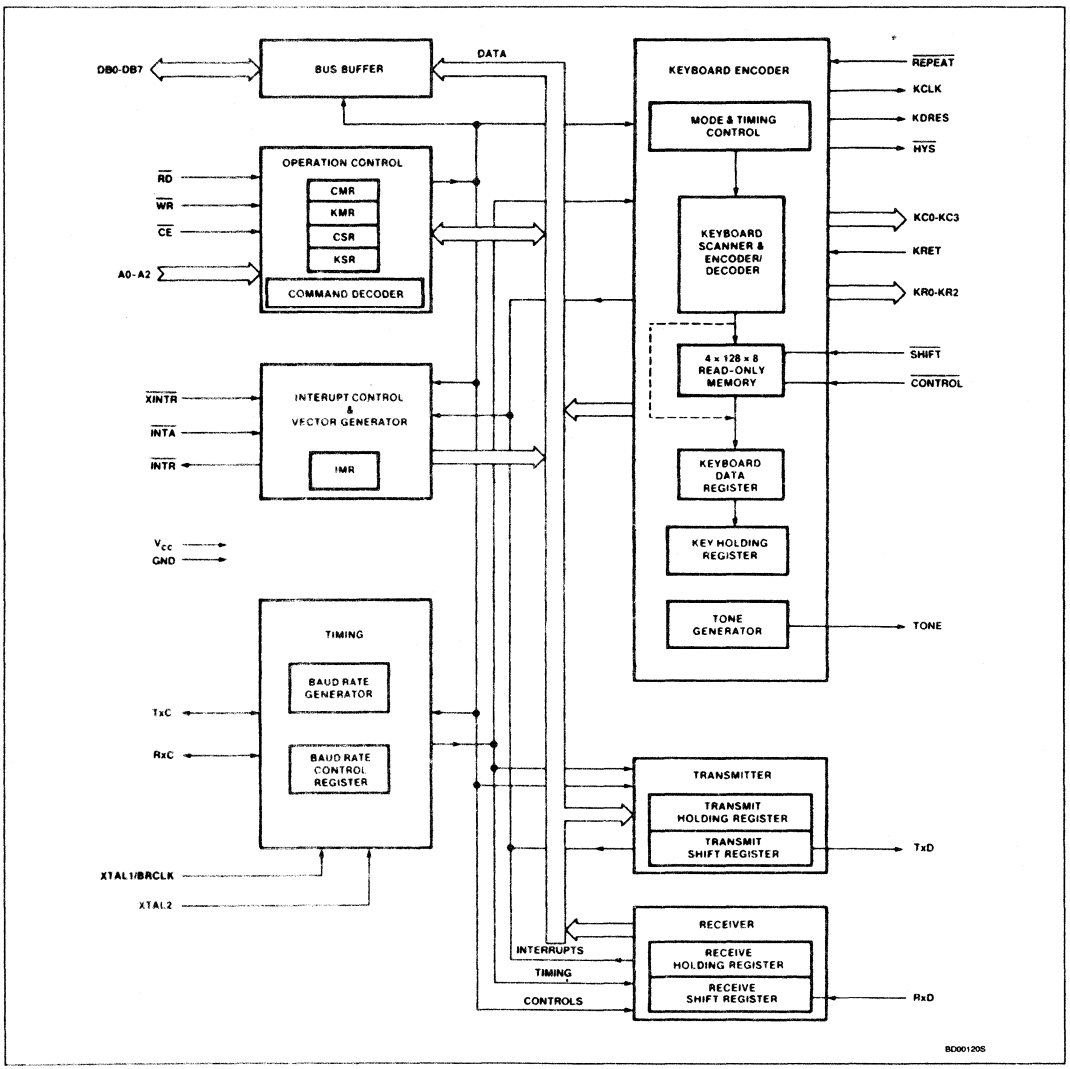


Pin	Function	Pin	Function
1	NC	23	$\overline{\text{XINTR}}$
2	$\overline{\text{HYS}}$	24	$\overline{\text{INTR}}$
3	KDRES	25	D4
4	KCLK	26	D5
5	KC3	27	D6
6	KC2	28	D7
7	KC1	29	$\overline{\text{INTA}}$
8	KC0	30	$\overline{\text{WR}}$
9	KR2	31	$\overline{\text{RD}}$
10	KR1	32	$\overline{\text{CE}}$
11	KR0	33	A0
12	NC	34	NC
13	$\overline{\text{REPEAT}}$	35	A1
14	$\overline{\text{SHIFT}}$	36	A2
15	$\overline{\text{CONTROL}}$	37	TxC
16	TONE	38	RxC
17	KRET	39	XTAL1
18	D0	40	XTAL2/BRCLK
19	D1	41	TxD
20	D2	42	RxD
21	D3	43	NC
22	V_{SS}	44	V_{CC}

ORDERING CODE

PACKAGES	V_{CC} = 5V±5%, T_A = 0°C to 70°C
Ceramic DIP	SCN2671AC1I40
Plastic DIP	SCN2671AC1N40
Plastic LCC	SCN2671AC1A44

BLOCK DIAGRAM



PIN DESCRIPTION

MNEMONIC	PIN NO.	TYPE	NAME AND FUNCTION
D0 – D7	16 – 19, 23 – 26	I/O	**Data Bus:** 8-bit three-state bidirectional data bus. All data, command and status transfers are made using this bus. D0 is the least significant bit; D7 is the most significant bit.
A0 – A2	31 – 33	I	**Address Lines:** Used to select internal PKCC registers or commands.
$\overline{RD}$	29	I	**Read Strobe:** When low, gates the selected PKCC register onto the data bus if $\overline{CE}$ is also low.
$\overline{WR}$	28	I	**Write Strobe:** When low, gates the contents of the data bus into the selected PKCC register if $\overline{CE}$ is also low.
$\overline{CE}$	30	I	**Chip Enable:** When high, places the D0 – D7 output drivers in a three-state condition. If $\overline{CE}$ is low, data transfers are enabled in conjunction with the $\overline{RD}$ and $\overline{WR}$ inputs.
$\overline{INTR}$	22	O	**Interrupt Request:** Several conditions may be programmed to request an interrupt to the CPU. It is an active low open-drain output. This pin will be inactive after power on reset or a master reset command.
$\overline{INTA}$	27	I	**Interrupt Acknowledge:** Used to indicate that an interrupt request has been accepted by the CPU. When $\overline{INTA}$ goes low, the PKCC outputs an 8-bit address vector on D0 – D7 corresponding to the highest priority interrupt currently active.
$\overline{XINTR}$	21	I	**External Interrupt:** An active low external interrupt input to the PKCC interrupt priority resolver.
TxC	34	I/O	**Transmitter Clock:** The function of this pin depends on bit 7 of the baud rate control register (BRR7). If external transmitter clock is selected (BRR7 = 0), it is an input for the transmitter clock. If internal transmitter clock is selected (BRR7 = 1), this pin is an output which is a multiple of the actual baud rate (1X, 16X) as selected by BRR5. The data is transmitted on the falling edge of TxC. It is an input after power on and after master reset or communications reset commands.
RxC	35	I/O	**Receiver Clock:** The function of this pin depends on BRR6. If external receiver clock is selected (BRR6 = 0), it is an input for the receiver clock. If internal receiver clock is selected (BRR6 = 1), this pin is an output which is a multiple of the actual baud rate (1X, 16X) as selected by BRR4. The received data is sampled on the rising edge of RxC. It is an input after power on and after master reset or communications reset commands.
TxD	38	O	**Transmitter Data:** This output is the transmitted serial data; the least significant bit is transmitted first. This pin is high after power on reset or a reset command that affects the transmitter.
RxD	39	I	**Receiver Data:** This input is the serial data input to the receiver. The least significant bit is received first.
XTAL1, XTAL2/BRCLK	36,37	I	**Connections for Crystal:** Provides an on-chip clock generator for the internal baud rate generator and the keyboard interface logic. If an external clock is provided, use XTAL2 as the clock input. See figures 20 and 21. All timing parameters such as keyboard scan time, tone frequency, and baud rate assume a clock input at the specified BRG input frequency. If this frequency is different the timing parameters will vary proportionately.
KR0 – KR2	10 – 8	O	**Keyboard Row Scan:** Decoded externally; selects one of eight rows.
KC0 – KC3	7 – 4	O	**Keyboard Column Scan:** Decoded externally; selects one of 16 columns.
KRET	15	I	**Key Return:** An active high level indicates that the key being scanned is closed.
$\overline{SHIFT}$	12	I	**SHIFT Key:** Active low input from the SHIFT key. The combination of $\overline{SHIFT}$ and $\overline{CONTROL}$ inputs select one of four possible codes from the internal key encoding ROM.
$\overline{CONTROL}$	13	I	**CONTROL Key:** Active low input from the CONTROL key. The combination of $\overline{SHIFT}$ and $\overline{CONTROL}$ inputs select one of four possible codes from the internal key encoding ROM.
$\overline{REPEAT}$	11	I	**REPEAT Key:** Active low input from the $\overline{REPEAT}$ key. Causes the key depression currently active to be repeated at a rate of approximately 15 times per second.
KCLK	3	O	**Keyboard Clock:** High frequency (approximately 400kHz) output used to scan capacitive keyboards.
KDRES	2	O	**Key Detect Reset:** Resets the analog detector before scanning a key. Used for capacitive keyboards.
$\overline{HYS}$	1	O	**Hysteresis Output:** Sent to the analog detector for capacitive keyboard applications. A low indicates the key currently being scanned has been recognized on previous scan cycles.
TONE	14	O	**Square Wave Output:** Used for tone generation.
V_{CC}	40	I	+5V power supply.
V_{SS}	20	I	Ground.

The PKCC has an interrupt mask register to selectively enable certain keyboard and communications status bits to generate interrupts. Priority encoded interrupt vectoring is available. Upon receipt of an interrupt acknowledge, an interrupt vector will be output on D0 – D7 reflecting the source of the interrupt. The interrupt source can also be read from an interrupt status register.

FUNCTIONAL DESCRIPTION

The PKCC consists of six major sections (see block diagram). These are the transmitter, receiver, timing, operation control, keyboard encoder, and a priority encoded interrupt control unit. These sections communicate with each other via an internal data bus and an internal control bus. The internal data bus interfaces to the microprocessor data bus via a bidirectional data bus buffer.

Operation Control

This functional block stores configuration and operation commands from the CPU and generates appropriate signals to various internal sections to control the overall device operation. It contains read and write circuits to permit communications with the microprocessor via the data bus and contains mode registers KMR and CMR, the command decoder, and status registers KSR and CSR. Details of operating modes and status information are presented in the Operation section of this data sheet. The register addressing is specified in table 1.

Timing

The PKCC contains a baud rate generator (BRG) which is programmable to accept external transmit or receive clocks or to divide an external clock to perform data communications. The unit can generate 16 baud rates, any of which can be selected for full duplex operation. The external clock to the baud rate generator can be applied directly to the XTAL2 input (see figure 21) or can be generated internally by connecting a crystal across the XTAL1, XTAL2 input pins. The clock input is also utilized by the keyboard encoder section. Thus, a clock must be provided even if external transmitter and receiver clocks are used.

Receiver

The receiver accepts serial data on the RxD pin, converts this serial input to parallel format, checks for break conditions, framing and parity errors, and loads an "assembled" character in the receive holding register for access by the CPU.

Transmitter

The transmitter accepts parallel data loaded by the CPU into the transmit holding register and converts it to a serial bit stream framed by the start bit, calculated parity bit (if specified), and stop bit(s). The composite serial stream of data is transmitted on the TxD output pin.

Table 1. REGISTER ADDRESSING

$\overline{CE}$	A2	A1	A0	$\overline{RD}/\overline{WR}$	FUNCTION
1	X	X	X	X	Three-state data bus
0	0	0	0	$\overline{WR}$	Reset command (see table 6)
0	0	0	0	$\overline{RD}$	Read interrupt status register (ISR)
0	0	0	1	$\overline{RD}$, $\overline{WR}$	Read/write communications mode register (CMR)
0	0	1	0	$\overline{WR}$	Write transmit holding register (TxHR)
0	0	1	0	$\overline{RD}$	Read receiver holding register (RxHR)
0	0	1	1	$\overline{WR}$	Write baud rate mode register (BRR)
0	0	1	1	$\overline{RD}$	Read communications status register (CSR)
0	1	0	0	$\overline{RD}$, $\overline{WR}$	Read/write interrupt mask register (IMR)
0	1	0	1	$\overline{RD}$, $\overline{WR}$	Read/write keyboard mode register (KMR)
0	1	1	0	$\overline{RD}$	Read keyboard holding register (KHR)
0	1	1	1	$\overline{RD}$	Read keyboard status register (KSR)
0	1	1	1	$\overline{WR}$	Miscellaneous commands (see description)

NOTE:
X = don't care.

Keyboard Encoder

The keyboard encoder provides encoded scanning signals for a matrix keyboard. Key depressions are detected on the KRET input. The debounced and verified key codes (or matrix addresses) are loaded into the key holding register for access by the CPU. Figures 1 and 2 illustrate the PKCC interface to contact and capacitive keyboards, respectively.

Interrupt Control

The interrupt controller unit contains a software programmable interrupt mask register which selectively enables status conditions from the keyboard encoder and communication controller to generate interrupts. The interrupts are priority encoded and individually generate an eight bit vector which is output on the data bus in response to a CPU interrupt acknowledge on the $\overline{INTA}$ input pin.

OPERATION

Keyboard Encoder

The keyboard is continuously scanned by KC0 – KC3 and KR0 – KR2 which are decoded externally to handle 128 possible keys (see figures 1 and 2). KC0 – KC3 select one of 16 columns and KR0 – KR2 multiplex the eight row return lines into the KRET pin. Debouncing is accomplished by remembering a 1 state at the KRET pin when a key is being addressed and verifying it one scan later. Once the key is verified, a key code is loaded into the keyboard data register (KDR). If the keyboard holding register (KHR) is empty, the contents of the KDR will be transferred to the KHR immediately; if the KHR is full (i.e., the CPU has not read the previous key code), the transfer will be held off until the KHR is read. The data transfer to the KHR causes keyboard data ready (KRDY) to be set in the keyboard status register.

For capacitive keyboards, the high frequency output KCLK can be used to gate the column scan to the keyboard (see figure 2). The key detector reset (KDRES) output resets the analog detector prior to scanning each key location. The output from the analog multiplexer is sensed and then latched in the analog detector. The $\overline{HYS}$ output controls the sense level. A 0 will lower the sense level causing hysteresis, and a 1 will raise the sense level with no hysteresis.

The $\overline{REPEAT}$ input enables the keyboard logic to recognize any key repeatedly, 15 times per second. Additionally, certain keys can be programmed to repeat automatically if depressed for more than one-half second.

A square wave is output on the TONE pin when the CPU issues a ring tone command to the PKCC.

Keyboard Mode Register

Operating modes are selected by programming the keyboard mode register (KMR), see figure 3. Bit KMR7 is used for testing the device. For normal operation, this bit should always be written to a 0. Bits KMR6 – KMR5 select the rollover modes for keyboard processing:

N-key Rollover: In this mode, the code corresponding to each key depression is loaded into the KDR as soon as that key is debounced, independent of the release of other keys. Two or more closures occurring within one scan cycle are considered to be simultaneous, which will set keyboard error in the keyboard status register (KSR1). As soon as the keyboard holding register is empty, the code in the KDR is transferred to the KHR and the KRDY status bit is set (KSR0).

N-Key Rollover With Latched Keys: This mode is the same as regular N-key rollover,

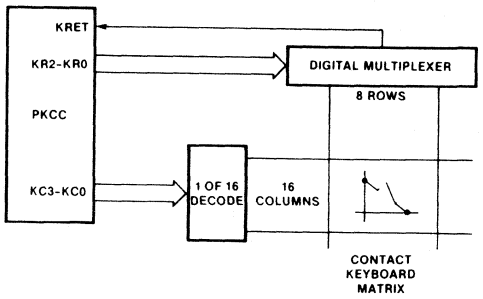


Figure 1. Contact Keyboard Interface

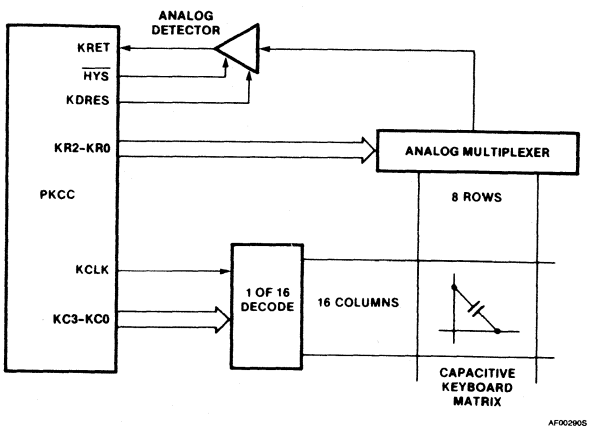


Figure 2. Capacitive Keyboard Interface

except that the keys which are assigned to row 0 of the keyboard matrix (KR2 - KR0 = 000) produce a code both when depressed and when released. The codes are independent of the states of the inputs at $\overline{\text{SHIFT}}$ and $\overline{\text{CONTROL}}$. If one or more of the latched keys are depressed when the keyboard is enabled (after a keyboard reset), the corresponding codes will be sent out as the keys are scanned and debounced. Note that simultaneous latched keys will not set KERR (KSR1) and that latched keys will not be auto-repeat and will not be affected by the $\overline{\text{REPEAT}}$ input.

Two-Key Rollover: The first key code is loaded into the KDR immediately and the second code is loaded only after the first key is released. Simultaneous keys will set KERR (KSR1), if three or more keys remain closed at any given time, the KERR bit will also be set. All keys must then be released before the next KRET will be processed.

Two-Key Inhibit: All keys must be released between keystrokes; otherwise, KERR (KSR1) will be set.

Bit KMR4 specifies the key encoding mode. Each key is assigned four 8-bit codes, corresponding to the states of the $\overline{\text{SHIFT}}$ and $\overline{\text{CONTROL}}$ inputs. If the encoded mode is programmed, the row/column address of the detected key is used to load one of the four key codes into the KDR. See table 2 for key code assignments. If the non-encoded mode is programmed, the row/column address is loaded directly into the KDR with the following format:

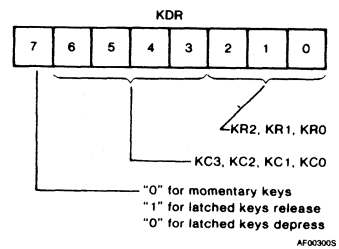
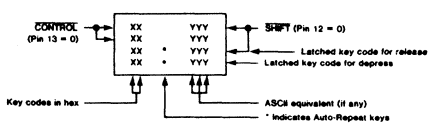


Table 2. STANDARD KEY CODES (HEX)

COLUMN (KC3 – KC0)	ROW (KR2 – KR0)															
	0	1	2		3		4		5		6		7			
0	E0	C0	1B	ESC	09	HT	1F	US	1A	SUB	30	0	2B	+
	F0	D0	1B	ESC	09	HT	1F	US	1A	SUB	30	0	3B	;
	E0	C0	1B	ESC	09 *	HT	1F	US	5A	Z	30	0	2B	+
	F0	D0	1B	ESC	09 *	HT	1F	US	7A	z	30	0	3B	;
1	E1	C1	21	!	11	DC1	01	SOH	18	CAN	3D	=	2A	*
	F1	D1	31	1	11	DC1	01	SOH	18	CAN	2D	–	3A	:
	E1	C1	21	!	51	Q	41	A	58	X	3D *	=	2A	*
	F1	D1	31	1	71	q	61	a	78	x	2D *	–	3A	:
2	E2	C2	22	"	17	ETB	13	DC3	03	ETX	1E	RS	1F	US
	F2	D2	32	2	17	ETB	13	DC3	03	ETX	1E	RS	1F	US
	E2	C2	22	"	57	W	53	S	43	C	7E	~	7F *	DEL
	F2	D2	32	2	77	w	73	s	63	c	5E	↑	5F *	—
3	E3	C3	23	#	05	ENQ	04	EOT	16	SYN	1C	FS	1B	ESC
	F3	D3	33	3	05	ENQ	04	EOT	16	SYN	1C	FS	1B	ESC
	E3	C3	23	#	45	E	44	D	56	V	7C	|	7B	{
	F3	D3	33	3	65	e	64	d	76	v	5C	\	5B	[
4	E4	C4	24	$	12	DC2	06	ACK	02	STX	08	BS	1D	GS
	F4	D4	34	4	12	DC2	06	ACK	02	STX	08	BS	1D	GS
	E4	C4	24	$	52	R	46	F	42	B	08 *	BS	7D	}
	F4	D4	34	4	72	r	66	f	62	b	08 *	BS	5D	]
5	E5	C5	25	%	14	DC4	07	BEL	0E	SO	10	DLE	08	BS
	F5	D5	35	5	14	DC4	07	BEL	0E	SO	10	DLE	08	BS
	E5	C5	25	%	54	T	47	G	4E	N	50	P	08	BS
	F5	D5	35	5	74	t	67	g	6E	n	70	p	08 *	BS
6	E6	C6	26	&	19	EM	08	BS	0D	CR	00	NUL	09	HT
	F6	D6	36	6	19	EM	08	BS	0D	CR	00	NUL	09	HT
	E6	C6	26	&	59	Y	48	H	4D	M	60	`	09 *	HT
	F6	D6	36	6	79	y	68	h	6D	m	40	@	09 *	HT
7	E7	C7	27	'	15	NAK	0A	LF	3C	<	7F	DEL	20	SP
	F7	D7	37	7	15	NAK	0A	LF	2C	,	7F	DEL	20	SP
	E7	C7	27	'	55	U	4A	J	3C	<	7F	DEL	20 *	SP
	F7	D7	37	7	75	u	6A	j	2C	,	7F	DEL	20 *	SP
8	E8	C8	28	(	09	HT	0B	VT	3E	>	0A	LF	0B	VT
	F8	D8	38	8	09	HT	0B	VT	2E	.	0A	LF	0B	VT
	E8	C8	28	(	49	I	4B	K	3E	>	0A	LF	0B *	VT
	F8	D8	38	8	69	i	6B	k	2E	.	0A	LF	0B *	VT
9	E9	C9	29	)	0F	SI	0C	FF	3F	?	0D	CR	0A	LF
	F9	D9	39	9	0F	SI	0C	FF	2F	/	0D	CR	0A	LF
	E9	C9	29	)	4F	O	4C	L	3F	?	0D	CR	0A *	LF
	F9	D9	39	9	6F	o	6C	l	2F	/	0D	CR	0A *	LF
A	EA	CA	37	7	34	4	31	1	30	0	A0		A6	
	FA	DA	37	7	34	4	31	1	30	0	B0		B6	
	EA	CA	37	7	34	4	31	1	30	0	A0		A6	
	FA	DA	37	7	34	4	31	1	30	0	B0		B6	
B	EB	CB	38	8	35	5	32	2	2E	.	A1		A7	
	FB	DB	38	8	35	5	32	2	2E	.	B1		B7	
	EB	CB	38	8	35	5	32	2	2E	.	A1		A7	
	FB	DB	38	8	35	5	32	2	2E	.	B1		B7	
C	EC	CC	39	9	36	6	33	3	BF		A2		A8	
	FC	DC	39	9	36	6	33	3	AF		B2		B8	
	EC	CC	39	9	36	6	33	3	9F		A2		A8	
	FC	DC	39	9	36	6	33	3	8F		B2		B8	
D	ED	CD	90		93		82		95		A3		A9	
	FD	DD	90		93		82		95		B3		B9	
	ED	CD	90		93		82 *		95		A3 *		A9 *	
	FD	DD	90		93		82 *		95		B3 *		B9 *	
E	EE	CE	91		80		84		81		A4		AA	
	FE	DE	91		80		84		81		B4		BA	
	EE	CE	91		80 *		84		81 *		A4 *		AA *	
	FE	DE	91		80 *		84		81 *		B4 *		BA *	
F	EF	CF	92		94		83		96		A5		AB	
	FF	DF	92		94		83		96		B5		BB	
	EF	CF	92		94		83 *		96		A5 *		AB *	
	FF	DF	92		94		83 *		96		B5 *		BB *	

This row contains the latched keys when that mode is selected (KMR6, KMR5 = 00)



TB00320S

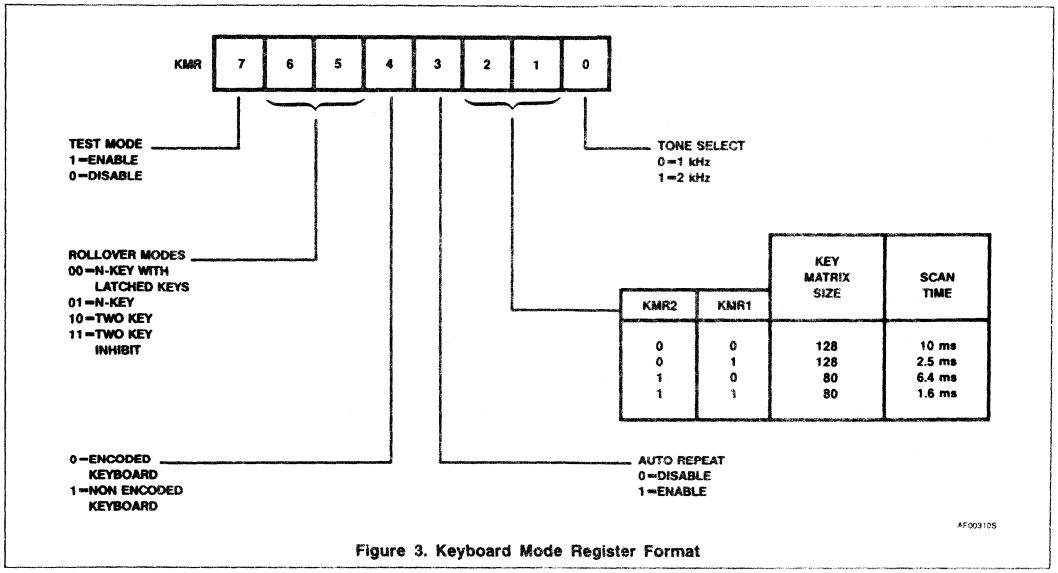


Figure 3. Keyboard Mode Register Format

Bit KMR3 enables the auto-repeat mode. In this mode, if a key that is programmed for auto-repeat is depressed for longer than one-half second, the key code will be loaded into the KDR approximately 15 times per second until that key is released. Only the non-control key codes will auto-repeat, i.e. $\overline{\text{CONTROL}}$ = 1. Table 2 specifies the auto-repeat keys.

KMR2 and KMR1 select the key matrix size and debounce time (scan rate). The keyboard row outputs (KR2, KR1, KR0) always scan from 0 to 7. The column outputs (KC3, KC2, KC1, KC0) scan from 0 to 15 for a 128 key matrix and from 0 to 9 for an 80 key matrix.

KMR0 selects between a 1kHz and 2kHz frequency to be output on the TONE pin in response to a ring tone command.

Keyboard Status Register

The keyboard status register (KSR) provides operational feedback to the CPU. Its format is illustrated in figure 4.

KSR7, 6 and 4 reflect the state of the inputs at the corresponding pins. $\overline{\text{CONTROL}}$ and $\overline{\text{SHIFT}}$ are latched at the time the key is accepted. As the verified codes are loaded into the KDR, the corresponding states of $\overline{\text{CONTROL}}$ and $\overline{\text{SHIFT}}$ are loaded into the KSR. $\overline{\text{REPEAT}}$ is updated on every matrix sample. The status bits are the complements of the input levels.

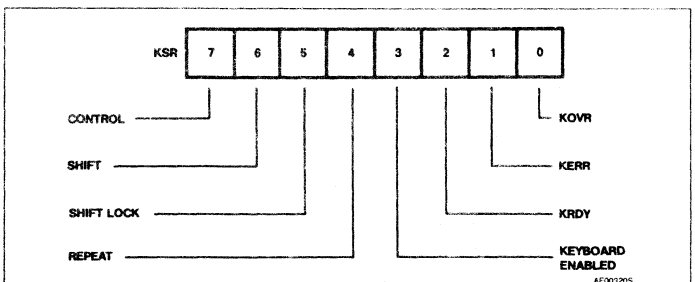


Figure 4. Keyboard Status Register Format

KSR5 reflects the state of the internal shift lock flag which is controlled by the set/reset shift lock commands.

KSR3 indicates that the keyboard controller is enabled. It is controlled by the set/clear keyboard enable command.

Keyboard overrun (KSR2) is set when both the KHR and KDR are full and a third key is validated. The original content of the KHR is preserved and the content of the KDR is overwritten with the new key code. This bit can be specified (by IMR1) to generate an interrupt and is cleared by the reset command with D2 = 1.

Keyboard error (KSR1) is set when the operator depresses more keys than are allowed in the selected rollover mode, or when keys are depressed simultaneously (within one scan cycle). This bit can be specified (by IMR3) to generate an interrupt and is cleared by the reset command with D1 = 1.

Keyboard data ready (KSR0) is set when the key code or address is transferred from the KDR to the KHR. This bit can be specified (by IMR2) to generate an interrupt. It is cleared when the CPU reads the KHR.

Communications Controller

The communications controller section of the PKCC comprises a full duplex asynchronous receiver/transmitter (UART) with a baud rate generator. Registers associated with these elements are the communications mode register (CMR), the baud rate control register (BRR), and the communications status register (CSR).

Receiver

The receiver accepts serial data on the RxD pin, converts the serial input to parallel format, checks for start bit, stop bit, parity bit (if any), or break condition, and presents the assembled character to the CPU. The receiver looks for a high to low (mark to space) transition of the start bit on the RxD input pin. If a transition is detected, the state of the RxD pin is sampled again after a delay of one half of the bit time. If RxD is then high, the start bit is invalid and the search for a valid start bit begins again. If RxD is still low, a valid start bit is assumed and the receiver continues to sample the input at one bit time intervals at the theoretical center of the bit, until the proper number of data bits and the parity bit (if any) have been assembled, and one stop bit has been detected. The least significant bit is received first. The data is then transferred to the receive holding register (RxHR) and the RxRDY bit in the CSR is set to a 1. If the character length is less than eight bits, the most significant unused bits in the RxHR are set to zero.

After the stop bit is detected, the receiver will immediately look for the next start bit. However, if a non-zero character was received without a stop bit (i.e. framing error) and RxD remains low for one half of the bit period after the stop bit was sampled, then the space is interpreted as a start bit.

The parity error, framing error and overrun error (if any) are strobed into the CSR at the received character boundary. If a break condition is detected (RxD is low for the entire character including the stop bit) only one character consisting of all zeros will be transferred to the RxHR and the received break bit in the CSR is set to 1 (RxRDY is not set when a break is received). The RxD input must return to a high condition for one bit time before a search for the next start bit begins.

Transmitter

The transmitter accepts parallel data from the CPU and converts it to a serial bit stream on the TxD output pin. It automatically sends a start bit followed by the data bits, an optional parity bit, and the programmed number of stop bits. The least significant bit is sent first. Following the transmission of the stop bits, if a new character is not available in the transmit holding register (TxHR), the TxD output remains high and the TxEMT bit in the CSR will be set to 1. Transmission resumes and the TxEMT bit is cleared when the CPU loads a new character into the TxHR. The transmitter can be forced to send a continuous low condition by a transmit break command.

If the transmitter is disabled, it continues operating until the character currently being transmitted is completely sent out.

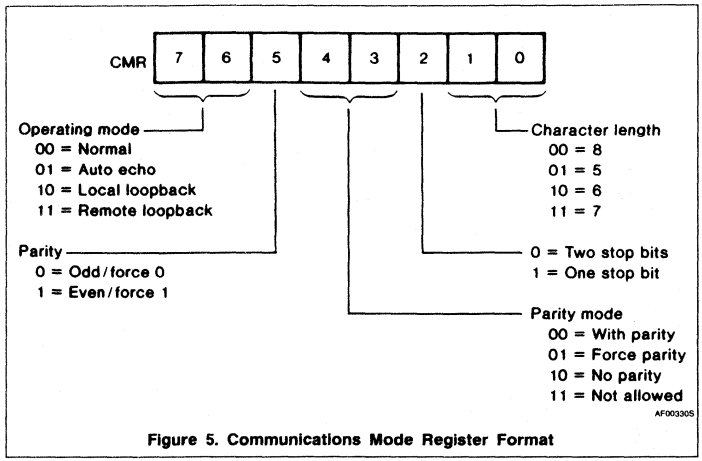


Figure 5. Communications Mode Register Format

Communication Mode Register

Figure 5 illustrates the bit format of the CMR, which controls the operational mode of the communications controller and the character parameters.

Bits CMR1 – CMR0 select a character length of 5, 6, 7, or 8 bits. The character length does not include the parity, start, or stop bits.

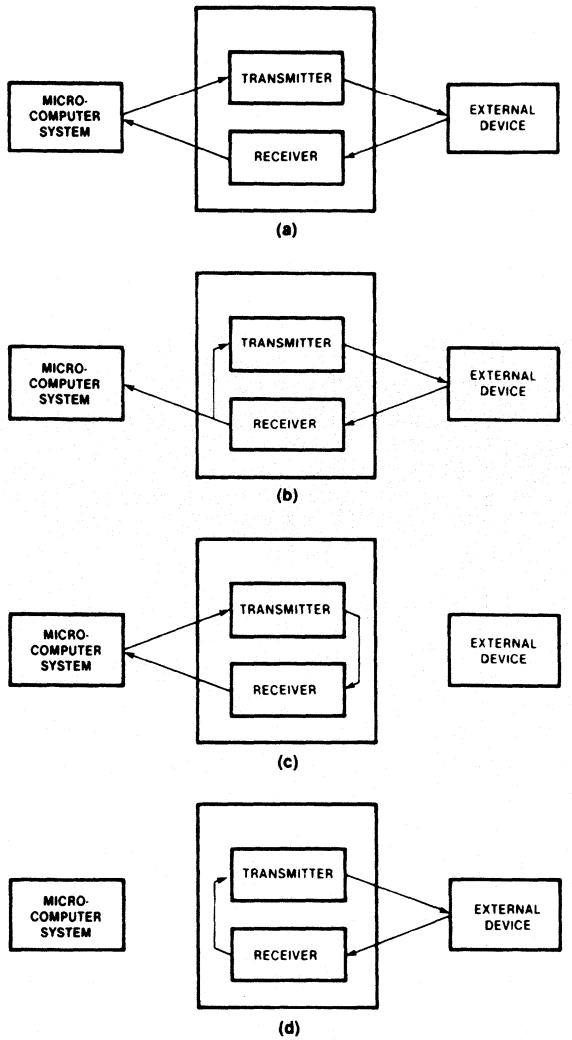
CMR2 selects the transmitted character framing as one or two stop bits. The receiver always checks for one stop bit.

The parity format is selected by bits CMR4 and CMR3. If parity or force parity is selected, a parity bit is added to the transmitted character and the receiver performs a parity check on incoming data. CMR5 selects odd or even parity and determines the polarity of the parity bit in the force parity mode.

The bits in the mode register affecting character assembly and disassembly (CMR5 – CMR0) can be changed dynamically and affect the characters currently being assembled in RxSR and transmitted by TxSR. To affect assembly of a received character, the CMR must be updated within n–1 bit times of the receipt of that character's start bit. To affect a transmitted character, the CMR must be updated within n–1 bit times of transmitting that character's start bit. (n = the smaller of the new and old character lengths).

The UART can operate in one of four modes, as illustrated in figure 6. The operating modes are selected by bits CMR7 and CMR6, which should only be changed when both the transmitter and receiver are disabled. CMR7-CMR6 = 00 is the normal mode, with the transmitter and receiver operating independently. CMR7-CMR6 = 01 places the UART in the automatic echo mode, which automatically retransmits the received data. The following conditions are true while in automatic echo mode:

1. Data assembled by the receiver is automatically placed in the transmit holding register and retransmitted on the TxD output.
2. The receive clock is used for the transmitter.
3. The receiver must be enabled, but the transmitter need not be enabled.
4. Status bit TxRDY is not set. TxEMT operates normally.
5. The receiver parity is checked, but is not regenerated for transmission, i.e., transmitted parity bit is as received.
6. Only the first character of a break condition is echoed; the TxD output will go high until the next received character is assembled.
7. CPU to receiver communication continues normally, but the CPU to transmitter link is disabled.



NOTES:
(a) Normal operating mode.
(b) Automatic echo mode.
(c) Local loopback mode.
(d) Remote loopback mode.

Figure 6. Operating Modes of the 2671 UART

Two diagnostic modes can also be configured. In local loopback mode (CMR7-CMR6 = 10):

1. The transmitter output is internally connected to the receiver input.
2. The transmit clock is used for the receiver.
3. The TxD output is held high.
4. The RxD input is ignored.
5. The transmitter must be enabled, but the receiver need not be enabled.
6. CPU to transmitter and receiver communications continue normally.

The second diagnostic mode is the remote loopback mode (CMR7-CMR6 = 11). In this mode:

1. Data assembled by the receiver is automatically placed in the transmit holding register and retransmitted on the TxD output.
2. The receive clock is used for the transmitter.
3. No data is sent to the local CPU, but the error status conditions (parity and framing) are set if required.
4. The received parity is checked, but is not regenerated for transmission, i.e., transmitted parity bit is as received.
5. The receiver must be enabled, but the transmitter need not be enabled.

Baud Rate Control Register

The baud rate control register (BRR) controls the frequency generated by the baud rate generator (BRG) and the clock source used by the receiver and transmitter. Its format is illustrated in figure 7.

BRR3 – BRR0 select one of sixteen frequencies to be generated by the BRG. See table 3.

BRR7 and BRR6 select the source of the transmit and receive clocks. If external clocks are chosen, (BRR7 = 0 or BRR6 = 0), then the clock rate factor is determined by BRR5 and BRR4. The external clock input(s) should be the desired baud rate multiplied by the clock rate factor.

If internal clock(s) are specified, (BRR7 = 1 or BRR6 = 1), the clock is supplied by the internal baud rate generator at the selected baud rate. The clock rate factor for internally generated clocks is always 16. Pins 35 and 34 become outputs for transmit or receive clocks, respectively. See table 4 for the description and selection of these outputs.

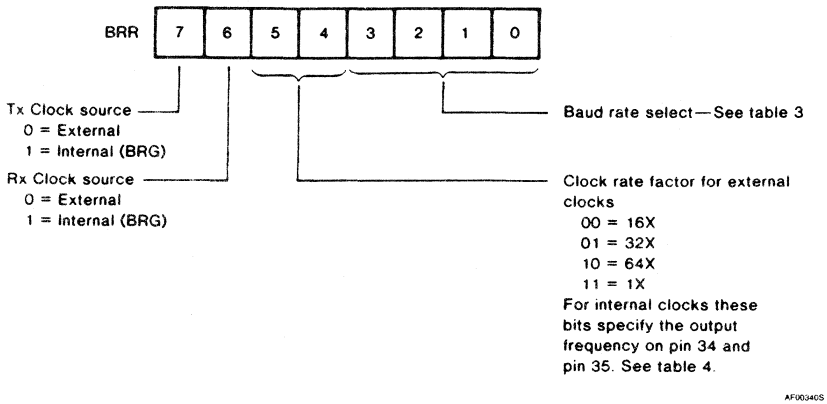


Figure 7. Baud Rate Control Register Format

Table 3. BAUD RATE GENERATOR CHARACTERISTICS (BRCLK = 4.9152MHz)

BRR3-0	BAUD RATE	ACTUAL FREQUENCY 16X CLOCK	PERCENT ERROR	DIVISOR
0000	50	0.8 kHz	–	6144
0001	110	1.7598	-0.01	2793
0010	134.5	2.152	–	2284
0011	150	2.4	–	2048
0100	200	3.2	–	1536
0101	300	4.8	–	1024
0110	600	9.6	–	512
0111	1050	16.8329	+0.20	292
1000	1200	19.2	–	256
1001	1800	28.7438	-0.20	171
1010	2000	31.9168	-0.26	154
1011	2400	38.4	–	128
1100	4800	76.8	–	64
1101	9600	153.6	–	32
1110	19200	307.2	–	16
1111	38400	614.4	–	8

Table 4. BAUD RATE CONTROL REGISTER

BRR7 – BRR4	CLOCK SOURCE		PIN FUNCTIONS		BRR3 – BRR0 BAUD RATE SELECTION
	TxC	RxC	PIN 34	PIN 35	
00**	E	E	TxC	RxC	The baud rates are listed in table 3.
01**	E	I	TxC	1X	
10**	I	E	16X	RxC	
1100	I	I	1X	1X	
1101	I	I	1X	16X	
1110	I	I	16X	1X	
1111	I	I	16X	16X	

NOTES:
1. ** = Clock rate factor for external clocks: 00 = 16X
   01 = 32X
   10 = 64X
   11 = 1X
2. E = External clock.
3. I = Internal clock (BRG).
4. 1X and 16X are clock outputs at 1 or 16 times the actual baud rate. For receive, the 1X output is the actual data sample clock.
5. BRR7 – BRR6 = 01 or 10 not permitted in automatic echo or remote loopback modes unless BRR5 – BRR4 = 00.

Communications Status Register

Figure 8 illustrates the bit format of the communications status register (CSR), which provides UART status to the CPU.

Receiver ready (CSR0) indicates that a received character is assembled and transferred to the RxHR and is ready to be read by the CPU. This bit can be specified (by IMR0) to generate an interrupt and is reset by reading the RxHR.

Transmitter ready (CSR1) indicates that the TxHR is empty and ready to be loaded with a character. This bit will be cleared when the TxHR is loaded and has not yet transferred the character to the transmit shift register (TxSR). TxRDY is reset when the transmitter is disabled. It will be set when the transmitter is enabled, provided that no data was loaded into the TxHR during the time the transmitter was disabled. This bit can be specified (by IMR7) to generate an interrupt.

Transmitter empty (CSR2) indicates that the transmitter has underrun, i.e., both the TxHR and TxSR are empty. This bit can only be set after transmission of at least one character, and is cleared when the TxHR is loaded by the CPU. TxEMT is reset when the transmitter is disabled. This bit can be specified (by IMR6) to generate an interrupt.

CSR3 will be set when the PKCC receives a command to transmit a break. This bit will be cleared after the break is completed.

Received break (CSR4) indicates that an all zero character of the programmed length has been received without a stop bit. Breaks originating in the middle of a received character can be detected. This bit is cleared when RxD returns to a high state for at least one bit time.

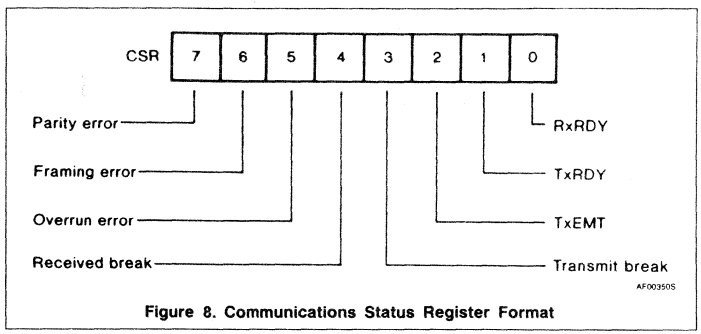


Figure 8. Communications Status Register Format

Receiver overrun (CSR5) indicates that the previous character in the RxHR has not been read by the CPU and that a new character has been loaded into the RxHR. This bit is cleared by a reset command with D3 = 1.

Framing error (CSR6) indicates that the stop bit has not been detected. The stop bit check is made in the middle of the first stop bit position. This bit is cleared by a reset command with D3 = 1.

Parity error (CSR7) indicates that a character was received with incorrect parity when 'with parity' or 'force parity' is enabled. This bit is cleared by a reset command with D3 = 1.

Interrupt Controller

The SCN2671 contains a maskable interrupt status register (ISR) which can be enabled to generate an active low interrupt request on the $\overline{INTR}$ output. The eight interrupt conditions in the ISR are individually enabled by writing a 1 into the corresponding bit of the interrupt mask register (IMR).

Each of the interrupt conditions is assigned a priority and a vector. When an enabled ISR bit is set, the SCN2671 asserts the $\overline{INTR}$ output. If the CPU activates the $\overline{INTA}$ input, the SCN2671 responds by placing the corresponding 8-bit vector on the data bus (D7 – D0). If multiple interrupts are pending, the vector corresponds to the condition with the highest priority. The interrupt will persist until all pending interrupt conditions are cleared.

The ISR can also be polled by reading at address A2 – A0 = 000. All pending interrupt conditions which are enabled by the IMR will be read independent of priority.

The bit assignments of the ISR and IMR and corresponding vectors and priorities are listed in table 5.

Table 5. INTERRUPT MASK REGISTER (IMR) AND INTERRUPT STATUS REGISTER (ISR)

BIT IN IMR/ISR	INTERRUPT CONDITION	PRIORITY	VECTOR ON D7 – D0		CONDITION RESET BY:
			BINARY	HEX	
IMR0/ISR0	RxRDY	1	11001111	CF	Read RxHR
IMR1/ISR1	KOVR	2	11010111	D7	Reset CMD (D2 = 1)
IMR2/ISR2	KRDY	3	11011111	DF	Read KHR
IMR3/ISR3	KERR	4	11100111	E7	Reset CMD (D1 = 1)
IMR4/ISR4	XINT[1]	5	11101111	EF	External
IMR5/ISR5	△BREAK[2]	6	11110111	F7	Reset CMD (D4 = 1)
IMR6/ISR6	TxEMT	7	11000111	C7	Load TxHR
IMR7/ISR7	TxRDY	8	11000111	C7	Load TxHR

NOTES:
1. XINT is an input from an external interrupt source, active low (pin 21).
2. △BREAK refers to the change of a received break condition.

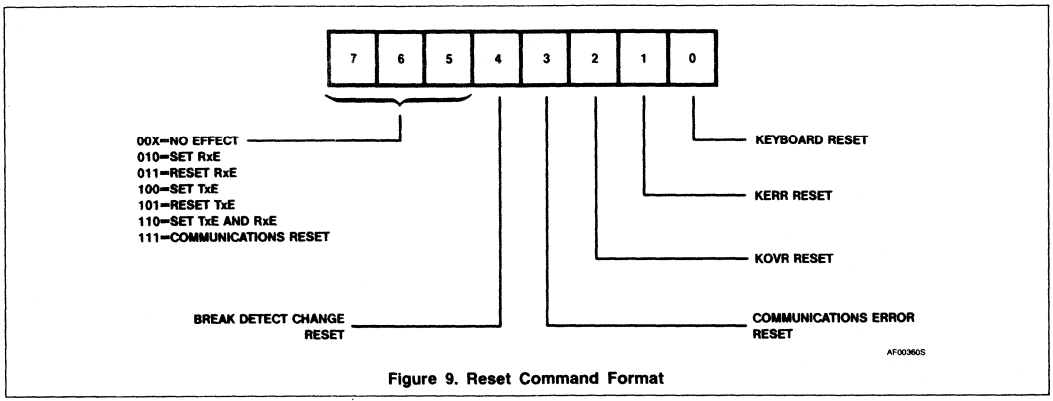


Figure 9. Reset Command Format

COMMANDS

In addition to the control exercised by programming of the PKCC control registers, several functions can be performed by executing command operations. There are two classes of commands which are initiated by writing to the SCN2671 at address A2 – A0 = 000 (reset command) and address A2 – A0 = 111 (miscellaneous commands). Individual commands are specified by the bit pattern on the data bus (D7 – D0).

Reset Commands

The reset command bit format is illustrated in figure 9 and the detail command descriptions are given in table 6.

A reset command with D7 – D0 = 111XXXX1 is a master reset for the SCN2671. This command must be given following a power on condition to release the internal power on reset latch which deactivates the SCN2671 on power up.

Miscellaneous Commands

The miscellaneous command format is illustrated in figure 10.

The transmit break commands force a break (steady low output) on the TxD pin immediately or after the character in the TxSR (if any) is transmitted. A timed break lasts for approximately 200ms, and a character break lasts for one character time including parity and stop bit time. In either case, TxRDY (CSR1) will be set at the beginning of the break which can be extended indefinitely (by 200ms or one character time increments) by reasserting the command in response to TxRDY. Note that these commands reset TxRDY. When a transmit break command is asserted, CSR3 will be set. The bit will be cleared after the break is completed.

The ring tone commands cause the tone generator to output a square wave on the TONE output. The tone durations are specified by the commands:

Ring tone short = 25ms
Ring tone long = 100ms

The tone frequency is either 1kHz or 2kHz, as specified by KMR0.

Table 6. RESET COMMAND DESCRIPTION

COMMAND	RESETS	COMMENTS
Keyboard reset	KMR7 – KMR0 KSR5, KSR3 – KSR0 IMR3 – IMR1	The keyboard controller is reset, ignoring the input at KRET.
KERR reset	KSR1	Keyboard error status bit reset.
KOVR reset	KSR2	Keyboard overrun status bit reset.
Communications error reset	CSR7-CSR5	Resets the receiver overrun, parity, and framing error status bits.
Break detect change reset	ISR5	Resets the break detect change bit in the interrupt status register.
Set RxE	See note.	Enables receiver operation.
Reset RxE	CSR7 – CSR4, CSR0 See note.	Disables the receiver.
Set TxE	See note.	Enables transmitter operation.
Reset TxE	CSR3-CSR1 See note.	Disables the transmitter. Sets the TxD output to a 1 after transmitting the character in TxSR.
Communications reset	CMR, CSR, BRR, TxE, RxE, IMR7-IMR5, IMR0	Resets the communication controller. The RxD input is ignored and the TxD output is set to a 1.
Master reset	CMR, CSR, BRR, TxE, RxE, KMR, KSR5, KSR3-KSR0, IMR7-IMR0 Releases the internally latched power on reset.	Resets the keyboard and communication controllers. Inputs at KRET and RxD are ignored and the TxD output is set to a 1.

NOTE:
Command does not affect the CMR or the BRR.

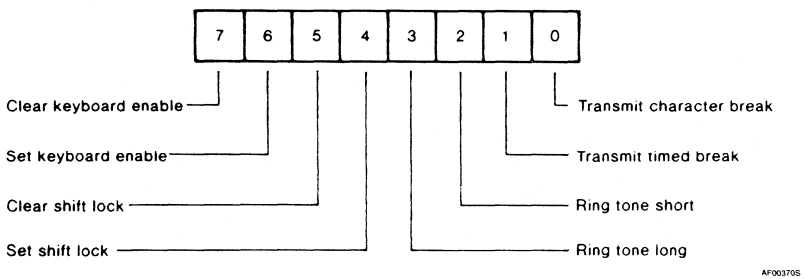


Figure 10. Miscellaneous Commands Format

The set/clear shift lock commands control the state of the internal shift lock flip flop. When shift lock is set, the keyboard controller encodes all key depressions as if the $\overline{\text{SHIFT}}$ input was asserted. The state of the shift lock flip flop is reflected in KSR5.

The set keyboard enable command enables the keyboard controller and sets KSR3 in the keyboard status register. THe clear keyboard enable command resets KSR3 and disables key processing at the KRET input. The keyboard controller is not reset by this command, and the current state of the keyboard (key depressions and latched key states) is preserved internally. When the keyboard is subsequently enabled, key processing resumes, old and new keys are debounced, and latched keys are encoded if there has been a change in their state.

MASK PROGRAMMABLE OPTIONS

Characteristics of certain portions of the PKCC are internally programmed by means of a read only memory. The items which can be programmed are:

- Key codes
- Auto-repeat keys
- Scan times, tone frequency, and tone duration
- Baud rates
- Interrupt vectors

Consult your local Signetics representative for costs, minimum quantities, and data submission requirements for customized versions of the PKCC.

ABSOLUTE MAXIMUM RATINGS

PARAMETER	RATING	UNIT
Operating ambient temperature[2]	0 to +70	°C
Storage temperature	−65 to +150	°C
All voltages with respect to ground[3]	−0.5 to +6.0	V

DC ELECTRICAL CHARACTERISTICS $T_A = 0°C$ to $+70°C$, $V_{CC} = 5V \pm 5\%$[4,5,6]

PARAMETER		TEST CONDITIONS	LIMITS			UNIT
			Min	Typ	Max	
V_{IL}	Input low voltage				0.8	V
V_{IH}	Input high voltage					
	XTAL1, XTAL2/BRCLK		4.0			V
	All other inputs		2.0			V
V_{OL}	Output low voltage	$I_{OL} = 1.6mA$			0.4	V
V_{OH}	Output high voltage (except $\overline{INTR}$)	$I_{OH} = -100\mu A$	2.4			V
I_{IL}	Input leakage current (except XTAL1, XTAL2/BRCLK)	$V_{IN} = 0$ to V_{CC}	−10		10	μA
I_{XTLIL}	Input low current	$V_{IN} = 0$				
	XTAL1		−80	−30		μA
	XTAL2/BRCLK[7]		−4.0	−1.5		mA
I_{XTLIH}	Input high current	$V_{IN} = V_{CC}$				
	XTAL1			30	80	μA
	XTAL2/BRCLK[7]			0.2	1.0	mA
I_{LL}	Data bus 3-state leakage current	$V_O = 0$ to V_{CC}	−10		10	μA
I_{CC}	Power supply current				150	mA

AC ELECTRICAL CHARACTERISTICS $T_A = 0°C$ to $+70°C$, $V_{CC} = 5V \pm 5\%$[4,5,6]

PARAMETER		TEST CONDITIONS	LIMITS			UNIT
			Min	Typ	Max	
Read timing (See figure 11)						
t_{AS}	Address set-up to $\overline{RD}$		50			ns
t_{CS}	$\overline{CE}$ set-up to $\overline{RD}$		50			ns
t_{PW}	$\overline{RD}$ pulse width		250			ns
t_{AH}	Address hold from $\overline{RD}$		20			ns
t_{CH}	$\overline{CE}$ hold from $\overline{RD}$	$C_L = 150pF$	0			ns
t_{DD}	Data delay for read				200	ns
t_{DF}	Data bus floating time for read	$C_L = 150pF$			100	ns
t_{AD1}	Access delay from any read to next read or write		250			ns
Write timing (See figure 12)						
t_{AS}	Address setting to $\overline{WR}$		50			ns
t_{CS}	$\overline{CE}$ set-up to $\overline{WR}$		50			ns
t_{PW}	$\overline{WR}$ pulse width		250			ns
t_{AH}	Address hold from $\overline{WR}$		20			ns
t_{CH}	$\overline{CE}$ hold from $\overline{WR}$		0			ns
t_{DS}	Data set-up		100			ns
t_{DH}	Data hold		10			ns
t_{AD2}	Access delay from any write to next read or write		250			ns
	Access delay from reset command to next read or write		1.0			μs

AC ELECTRICAL CHARACTERISTICS (Continued)

PARAMETER		TEST CONDITIONS	LIMITS			UNIT
			Min	Typ	Max	
Interrupt acknowledge timing (see figure 13)						
t_{PWI}	$\overline{INTA}$ pulse width		300			ns
t_{DDI}	Data delay time for interrupt vector	C_L = 150pF			250	ns
t_{DFI}	Data bus floating time after $\overline{INTA}$	C_L = 150pF	10		100	ns
t_{ADI}	$\overline{INTA}$ to $\overline{INTA}$ access delay		300			ns
$\overline{INTR}$ reset timing (see figure 14)						
t_{RI}	$\overline{INTR}$ delay from:					
	Read RxHR (RxRDY)				400	ns
	Read KHR (KRDY)				400	ns
	Reset commands (KOVR,KERR,BREAK)				450	ns
	Load TxHR (TxEMT,TxRDY)				400	ns
	Mask bit reset				300	ns
Keyboard timing (see figure 15 and 16)						
f_{KCLK}	KCLK frequency			409		kHz
t_{KBD}	KR_i,KC_i to KRET sample delay:					
	FAST SCAN		12.0			μs
	SLOW SCAN		55.0			μs
t_{POS}	Scan time per matrix position:					
	FAST SCAN			20		μs
	SLOW SCAN			80		μs
t_{KRD}	KDRES delay from KCLK	C_L = 150pF			400	ns
t_{KRH}	KDRES hold from KLCK	C_L = 150pF			400	ns
t_{HYSD}	$\overline{HYS}$ delay from KCLK	C_L = 150pF			600	ns
t_{RCD}	KR_i,KC_i delay from KLCK	C_L = 150pF			400	ns
UART timing (see figure 17, 18, 19)						
t_{RXS}	RxD set-up time		200			ns
t_{RXH}	RxD hold time		200			ns
t_{TXD}	TxD delay from falling edge of TxC	C_L = 150pF			300	ns
t_{TCS}	Skew between TxD transition and falling edge of TxC output	C_L = 150pF		0		ns
t_{BRH}	XTAL1 clock high[8]		70			ns
t_{BRL}	XTAL1 clock low[8]		70			ns
f_{BRG}	BRG input frequency		1.0	4.9152	5.075	MHz
$f_{R/T}$	TxC or RxC input frequency	Clock rate factor = 16X, 32X, 64X			1.3	MHz
		Clock rate factor = 1X			1.0	MHz
$t_{R/TH}$	TxC or RxC clock high		350			ns
$t_{R/TL}$	TxC or RxC clock low		350			ns

NOTES:

1. Stresses above those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is stress rating only and functional operation of the device at these or at any other condition above those indicated in the operation section of this specification is not implied.
2. For operating on elevated temperatures, the device must be operated based on +150°C maximum function temperature.
3. This product includes circuitry specifically designed for the protection of its internal devices from damaging effects of excessive static charge. Nonetheless, it is suggested that conventional precautions be taken to avoid applying any voltages larger than the rated maxima.
4. Parameters are valid over operating temperature range unless otherwise specified.
5. All voltage measurements are referenced to ground (V_{SS}). All input signals swing between 0.4V and 2.4V with a transition time of 20ns maximum and time measurements are referenced at input voltages of 0.8V, 2.0V and at output voltages of 0.8V, 2.0V as appropriate, unless otherwise specified.
6. Typical values are at +25°C, typical supply voltages and typical processing parameters.
7. XTAL2 input currents are measured with XTAL grounded.
8. See figures 20 and 21 for XTAL1, WTAL2 connections for driving XTAL2 with an external clock. Input levels for XTAL1 and XTAL2 are $V_{IL} \leq 0.8V$, $V_{IH} \geq 4.0V$, and t_{BRL} and t_{BRH} are measured at these levels.

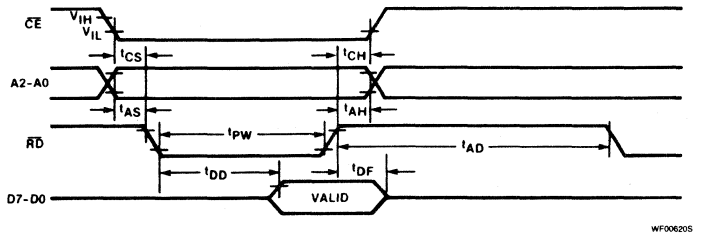


Figure 11. Read Timing

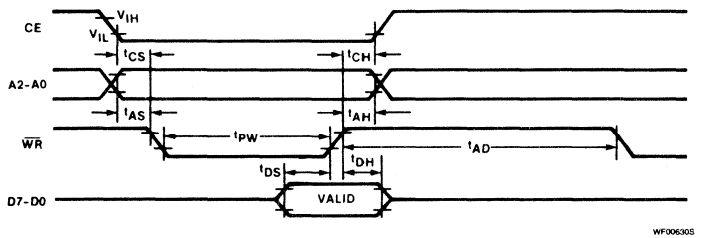


Figure 12. Write Timing

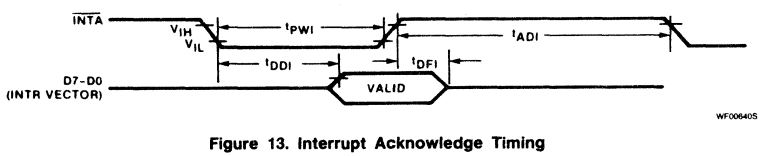


Figure 13. Interrupt Acknowledge Timing

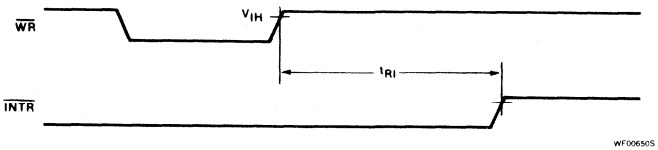
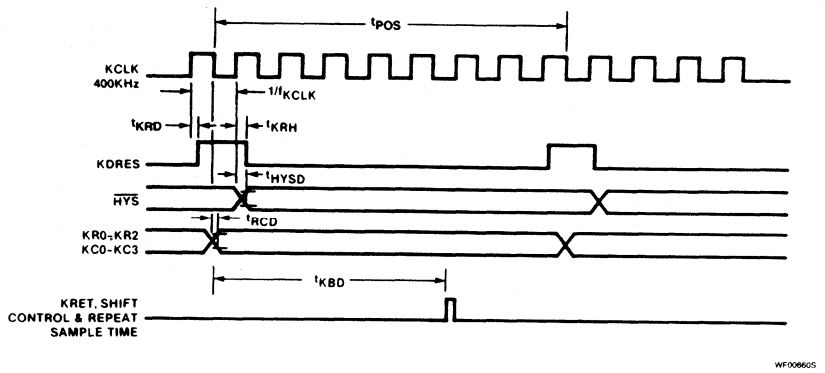


Figure 14. Interrupt Reset Timing



NOTE:
Scan timing shown is for fast scan (KMR1 = 1). For slow scan (KMR1 = 0) all signals except KCLK run at ¼ the shown rates.

Figure 15. Keyboard Scan Timing

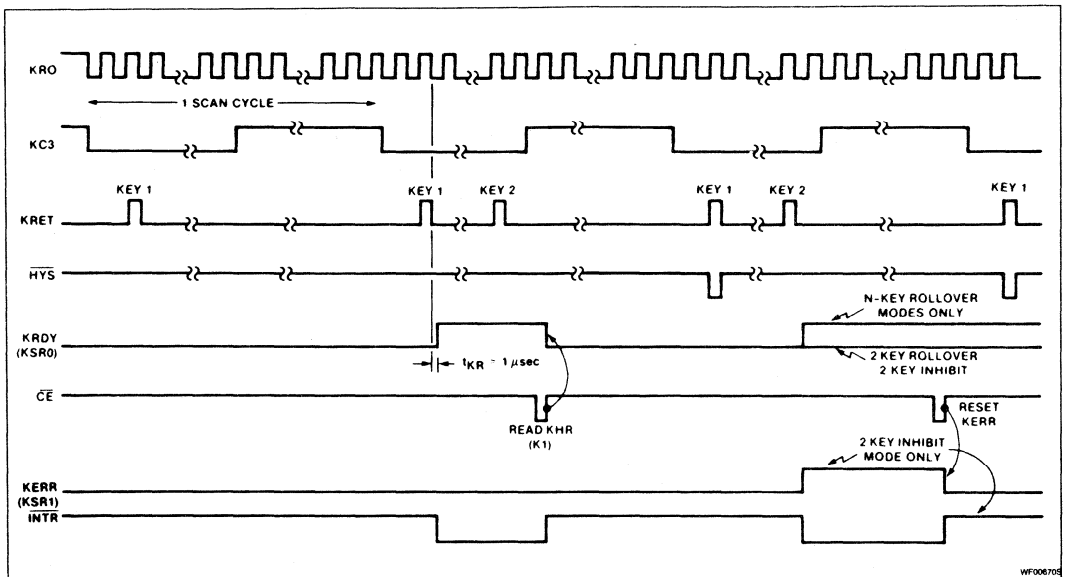
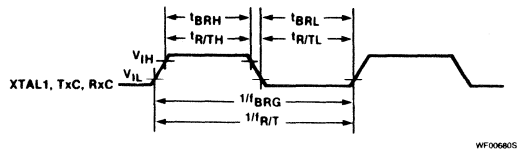
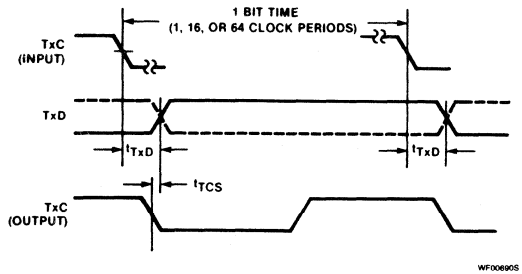


Figure 16. Keyboard Timing

Clock



Transmit



Receive

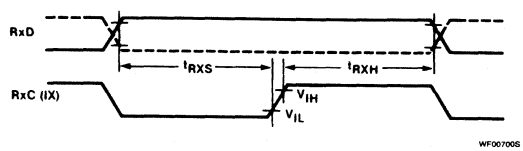


Figure 17.

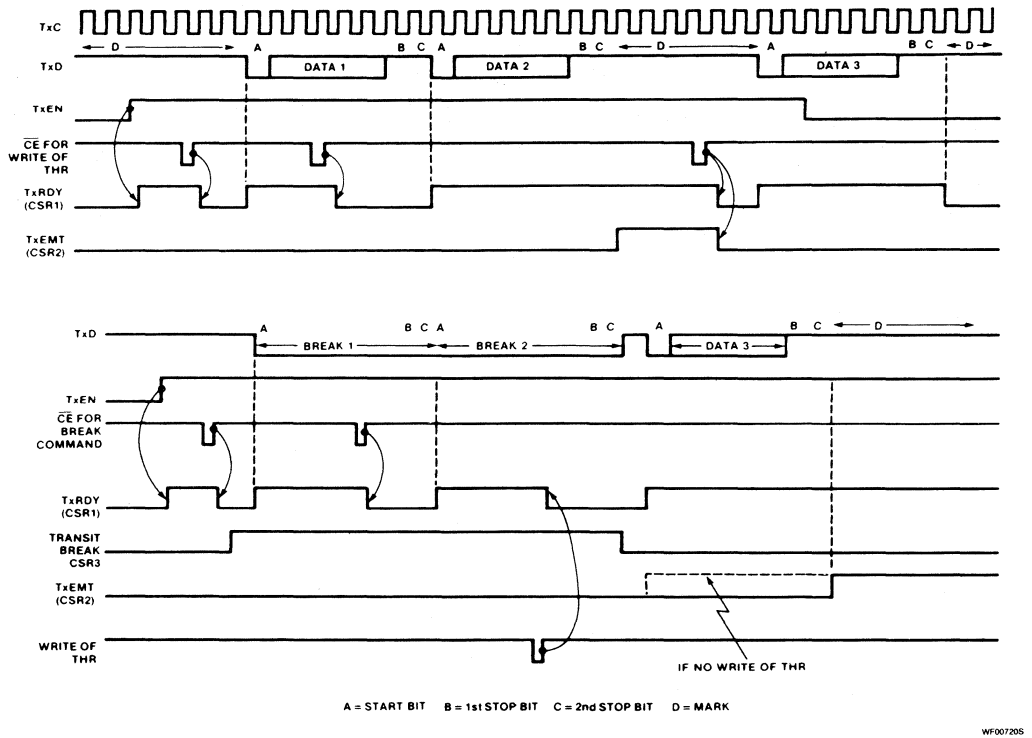


Figure 18. Transmitter Timing (5-Bit Characters, no Parity, 2 Stop Bits)

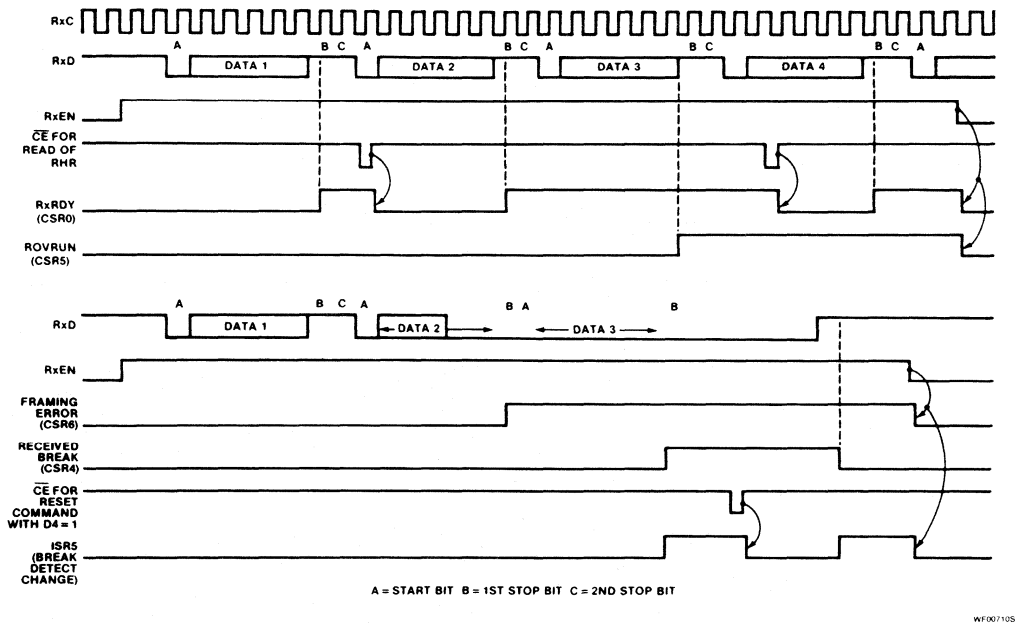


Figure 19. Receiver Timing (5-Bit Characters, no Parity, 2 Stop Bits)

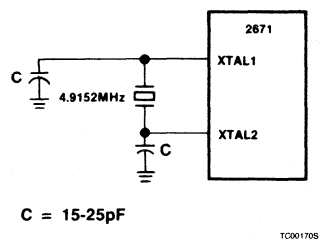


Figure 20. Crystal Connections BRG Clock

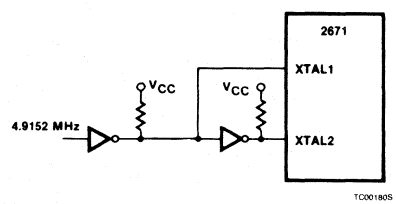


Figure 21. Connection for External BRG Clock Source

Table 7. REGISTER FORMAT SUMMARY

	7	6	5	4	3	2	1	0
	Test Mode	Rollover modes		Keyboard	Auto repeat			Tone select
KMR	1 = Enable 0 = Disable	00 = N-key with latched keys 01 = N-key 10 = Two keys 11 = Two key inhibit		0 = Encoded 1 = Nonen-coded	0 = Disable 1 = Enable	KMR2 KMR1 0 0 0 1 1 0 1 1	Key Matrix Size / Scan Time 128 10ms 128 2.5ms 80 6.4ms 80 1.6ms	0 = 1kHz 1 = 2kHz
KSR	CONTROL	SHIFT	SHIFT LOCK	REPEAT	Keyboard Enabled	KOVR	KERR	KRDY
	Operating Mode		Parity	Parity Mode		Stop Bits	Character Length	
CMR	00 = Normal 01 = Auto echo 10 = Local loopback 11 = Remote loopback		0 = Odd/ force 0 1 = Even/ force 1	00 = With parity 01 = Force parity 10 = No parity 11 = Not allowed		0 = Two 1 = One	00 = 8 01 = 5 10 = 6 11 = 7	
	Tx Clock source	Rx Clock source	Clock rate factor for external clocks	Baud rate select (BRR3-BRR0 in hex)				
BRR	0 = External 1 = Internal (BRG)	0 = External 1 = Internal (BRG)	00 = 16X 01 = 32X 10 = 64X 11 = 1X For internal clocks these bits specify the output frequency on pins 34 and 35 (table 4).	0 = 50 1 = 110 2 = 134.5 3 = 150 (BRCLK = 4.9152MHz)	4 = 200 5 = 300 6 = 600 7 = 1050	8 = 1200 9 = 1800 A = 2000 B = 2400	C = 4800 D = 9600 E = 19200 F = 38400	
CSR	Parity error	Framing error	Overrun error	Received break	Transmit break	TxEMT	TxRDY	RxRDY
IMR/ISR	TxRDY	TxEMT	BREAK CHANGE	XINT	KERR	KRDY	KOVR	RxRDY
Reset Command Format	00X = No effect 010 = Set RxE 011 = Reset RxE 100 = Set TxE	101 = Reset TxE 110 = Set TxE and RxE 111 = Communica-tions reset		Break detect change reset	Communi-cations error reset	KOVR reset	KERR reset	Keyboard reset
Miscellaneous Commands Format	Clear keyboard enable	Set keyboard enable	Clear shift lock	Set shift lock	Ring tone long	Ring tone short	Transmit timed break	Transmit character break

Signetics

AN401
Using the 2670/71/72/77 CRT Terminal Chip Set
Application Note

Microprocessor Products

INTRODUCTION

Microprocessors and LSI have had a dramatic impact on the implementation and capabilities of alphanumeric CRT terminals. The first generation of CRT terminals were little more than 'glass teletypes'. Current designs, implemented with microprocessors, are characterized by an abundance of sophisticated features that were previously not economically feasible: a universal hardware design that can adapt to different user requirements simply by changing software or firmware; programmability to provide end users with the flexibility to execute specialized routines; and local intelligence and storage which off-loads the host CPU by permitting data manipulation and verification at the terminal site.

Just as the impact of microcomputers has been felt in the functional capabilities of terminals, advances in semiconductor technology have revolutionized the hardware implementation. Designs that previously consisted of 100 to 200 ICs can now be realized with a few dozen MSI and LSI devices. The majority of the LSI manufacturers' effort with respect to CRT terminals has been concentrated in the 'CRT controller' area. These circuits provide the character timing, display addressing, and sync generation functions required by all terminals. However, these controllers need to be supported by many other external circuits to implement a complete terminal.

The purpose of this application note is to provide information on the use of four new Signetics CRT terminal products which, when combined with standard CPUs, memories, and TTL, allow the implementation of a wide spectrum of CRT terminal capabilities in as few as 15 total packages. These devices are:

- 2670 Display Character and Graphics Generator (DCGG)
- 2671 Programmable Keyboard and Communications Controller (PKCC)
- 2672 Programmable Video Timing Controller (PVTC)
- 2677 Video and Attributes Controller (VAC)

MAJOR ELEMENTS OF A CRT TERMINAL

Figure 1 shows the major elements of a typical low-end microcomputer-based CRT terminal. In this system, the CPU examines inputs from the data communications line and the keyboard and places the data to be displayed in a display buffer memory, which is typically a RAM which holds the data for a single or multiple screenload (page) or for a single character row. High-end ('smart'

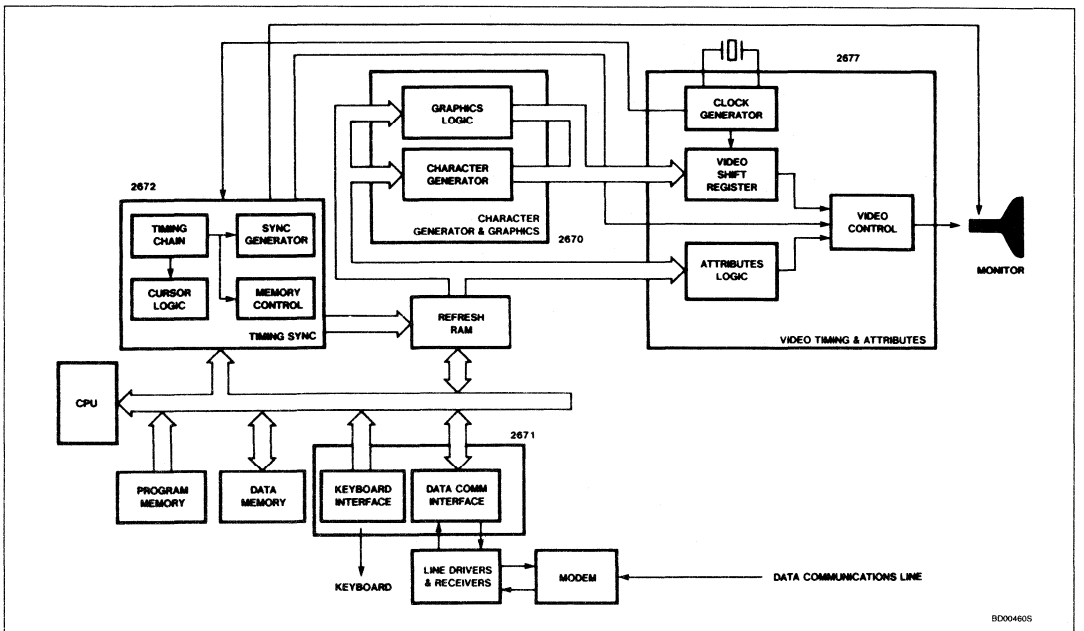


Figure 1. CRT Terminal Block Diagram

Using the 2670/71/72/77 CRT Terminal Chip Set AN401

and 'intelligent') terminals start with the same base, but append additional circuits to provide more features and capabilities. The following sections describe the functions of each of the major blocks.

Character Timing and Sync Generation

The major function of this block is to generate the horizontal and vertical timing signals required to produce the TV raster on the CRT monitor. Other functions include the generation of display memory addresses in synchronism with the monitor scan and in accordance with a defined screen format (characters per row, scan lines per row and rows per screen), generation of a cursor signal at the appropriate scan position, and generation of video blanking signals during retrace intervals.

I/O Interface

In its simplest form, this block provides an interface to a keyboard to identify the key depressed and a serial communications link, normally operating in an asynchronous format, between the terminal and the host computer. Although these functions could be performed programmatically by the terminal CPU system, removing these functions to intelligent controllers unburden the system CPU and allow it to be used more effectively to provide additional features with a relatively small cost impact.

Character and Graphics Generation

These circuits convert the data stored in the display memory to the line by line dot patterns required to display the data on the CRT monitor.

Video Timing and Visual Attributes

This section contains the high speed (dot rate) circuits necessary to convert the parallel data from the character and graphics generation circuits to the serial video stream required by the CRT. Also included are circuits to sum visual display attributes such as blinking, high/low intensity, reverse video, and underlining into the video stream.

SIGNETICS' CRT CHIP SET

As mentioned previously, the Signetics CRT 'set' consists of four circuits. The functions of these circuits correspond closely to the four major CRT terminal blocks described above. The circuits have been partitioned so as to allow each to be used independent of the others, allow several alternative methods of implementing the display memory interface so that the hardware can be tailored to the system requirements, provide a full complement of programmable capabilities, and minimize the number of support circuits required.

The following sections give a brief description of each of the circuits. The reader is referred to the individual data sheets for full operational details.

2670 Display Character and Graphics Generator (DCGG)

The DCGG, figure 2, is a mask-programmable 11,648-bit line select character generator. It contains 128 10 x 9 characters placed in a 10 x 16 matrix, and has the capability of shifting certain characters, such as j, y, g, p and q, that normally extend below the baseline; effectively, the 9 active lines are lowered within the matrix to compensate for the character's position.

Seven bits of an 8-bit address code are used to select 1 of the 128 available characters. The eighth bit functions as a chip enable signal. Each character is defined by a pattern of logic 1s and 0s stored in a 10 x 9 matrix. When a specific 4-bit binary line address code is applied, a word of 10 parallel bits appears to the output. The lines can be sequentially selected, providing a 9-word sequence of 10 parallel bits per word for each character selected by the address inputs. As the line address inputs are sequentially addressed, the device will automatically place the 10 x 9 character in 1 of 2 preprogrammed positions on the 16-line matrix with the positions defined by the 4-line address inputs. One or more of the 10 parallel outputs can be used as control signals to selectively enable functions such as half-dot shift, color selection, etc.

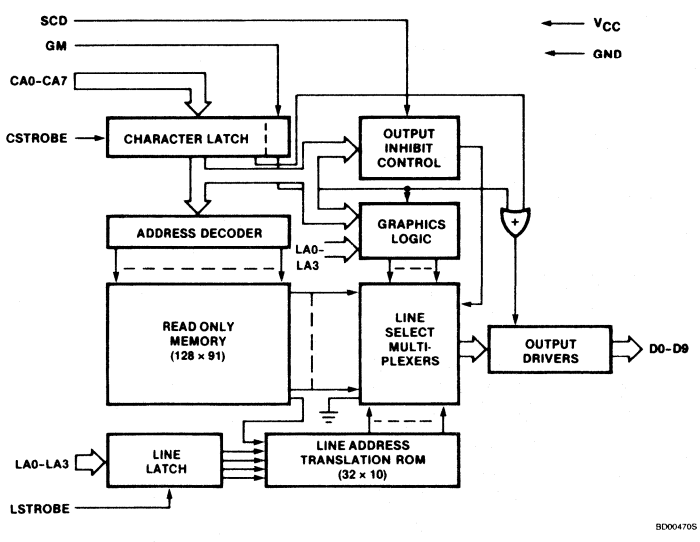


Figure 2. 2670 Display Character and Graphics Generator

Using the 2670/71/72/77 CRT Terminal Chip Set | AN401

The 2670 DCGG includes latches to store the character address and line address data. A control input to inhibit character data output for certain groups of characters is also provided. The 2670 also includes a graphics capability, wherein the 8-bit character code is translated directly into 256 possible user programmable graphic patterns. Thus, the DCGG can generate data for 384 distinct patterns, of which 128 are defined by the mask programmable ROM.

2671 Programmable Keyboard and Communications Controller (PKCC)

The 2671, figure 3, is an MOS LSI device which provides a versatile keyboard interface and also functions as an asynchronous communications controller. It is intended for use in microprocessor based systems and provides an eight bit data bus interface.

The keyboard controller handles the scanning, debounce, and encoding of mechanical or capacitive keyboards with a maximum of 128 keys utilizing any of four programmable rollover modes. A mask programmable ROM provides four levels of key encoding, corresponding to the separate shift and control input combinations. An eight bit keyboard status register transmits status information to the CPU. Programmable features include rollover mode, scan rate and debounce time, coded or uncoded operation, and automatic repeat operation.

The communications section of the PKCC is a universal asynchronous receiver and transmitter (UART). The receiver accepts serial input data and converts it to parallel data characters. Simultaneously, the transmitter accepts parallel data from the CPU data bus and outputs it in serialized form. Received data is checked for parity and framing errors, and break conditions are flagged. Character lengths can be programmed as 5, 6, 7, or 8 bits not including parity, start or stop bits. An internal baud rate generator (BRG) operating from an external clock or directly from a crystal can be used to derive one of sixteen receive and/or transmit clocks. An eight bit communications status register provides status information to the CPU.

The PKCC has an interrupt mask register to selectively enable keyboard and communications status bits to generate interrupts. Priority encoded interrupt vectoring is available. Upon receipt of an interrupt acknowledge, a mask programmable interrupt vector will be output on the data bus reflecting the source of the interrupt. The mask enabled interrupt sources can also be read directly.

2672 Programmable Video Timing Controller (PVTC)

The 2672 PVTC, figure 4, is a programmable device designed for use in CRT terminals and display systems that employ raster scan techniques. The PVTC generates the vertical and horizontal timing signals necessary for the display of interlaced or non-interlaced data on a CRT monitor. Also, the 2672 provides consecutive addressing to a user specified display buffer memory domain and controls the CPU-display buffer interface for various buffer configuration modes. A variety of operating modes, display formats, and timing profiles can be implemented by programming the control registers in the PVTC.

The CPU initializes the 2672 control and timing registers for the desired timing profiles and memory configuration. The PVTC provides the handshake control for CPU access to the display buffer. One of four memory access modes may be programmed: independent mode, transparent mode, shared mode, and row mode. These modes are described in the System Configurations section of this application note.

In all modes, the PVTC provides addresses for the display buffer which outputs the character codes to the 2670 Display Character and Graphics Generator (DCGG) and visual attribute codes to the 2673 Video Attributes Controller (VAC). The DCGG and PVTC supply the dot data and sync timing to the VAC which generates the serialized video.

Programmable features of the PVTC include screen format (characters/row, rows/screen, scan lines/row), horizontal and vertical timing parameters, cursor type (block or underline) and blink rate, character blink rate, interlaced or non-interlaced operation, and single or double height characters.

The PVTC is capable of producing interrupts based upon several internal conditions. By using these interrupts (or by polling the equivalent status register) display features such as non-consecutive buffer addressing for split screen operation, multiple cursors, horizontal and vertical scrolling, and smooth vertical scroll can be implemented.

2677 Video and Attributes Controller (VAC)

The 2677, figure 5, is a bipolar LSI device designed for CRT terminals and display systems that employ raster scan techniques. It contains a high speed video shift register, field and character attributes logic, attribute latch, cursor format logic and half dot shift control, and can be programmed for a light or dark screen background.

The VAC visual attribute capabilities are reverse video, character blank, blink, underline, highlight, and light pen strike-thru or, optionally, graphics. Each attribute has a separate control input which is latched internally when the AFLAG input is asserted. If the AMODE input is low, the attributes are valid for one character time. If AMODE is high, the attributes remain valid until the field is terminated by strobing in a new attributes set. The attributes are double buffered on a row by row basis internally so that field attributes can extend across character row boundaries thereby eliminating the necessity of starting each row with an attribute set.

The horizontal dot frequency is the basic timing input element to the VAC; internally, this clock is divided down to provide a character clock output for system synchronization. Ten bits of dot data are parallel loaded into the video shift register on each character boundary. The video data is shifted out on three outputs at the dot frequency. On the video output, the video is presented as a three level signal representing low, medium and high intensities, and the three intensities are also encoded on the two TTL compatible video outputs.

SYSTEM CONFIGURATIONS

The PVTC supports four common system configurations of display buffer memory interface, designated the independent, transparent, shared, and row buffer modes. The first three modes utilize a single or multiple page RAM and differ primarily in the means used to transfer display data between the RAM and the CPU. The row buffer mode makes use of a single row buffer (which can be a shift register or a small RAM) that is updated in real time to contain the appropriate display data.

Independent Mode

The CPU to RAM interface configuration for this mode is illustrated in figure 6. Transfer of data between the CPU and display memory is accomplished via a bidirectional latched port and is controlled by the PVTC signals read data buffer ($\overline{RDB}$), write data buffer ($\overline{WDB}$), and buffer chip enable ($\overline{BCE}$). This mode provides a non-contention type of operation that does not require address multiplexers. The CPU does not address the memory directly – the read or write operation is performed at the address contained in the cursor address register or the pointer address register as specified by the CPU. The PVTC enacts the data transfers during blanking intervals in order to prevent visual disturbances of the displayed data.

The CPU manages the data transfers by supplying commands to the PVTC. The commands used are:

1. Read/Write at pointer address.

Using the 2670/71/72/77 CRT Terminal Chip Set AN401

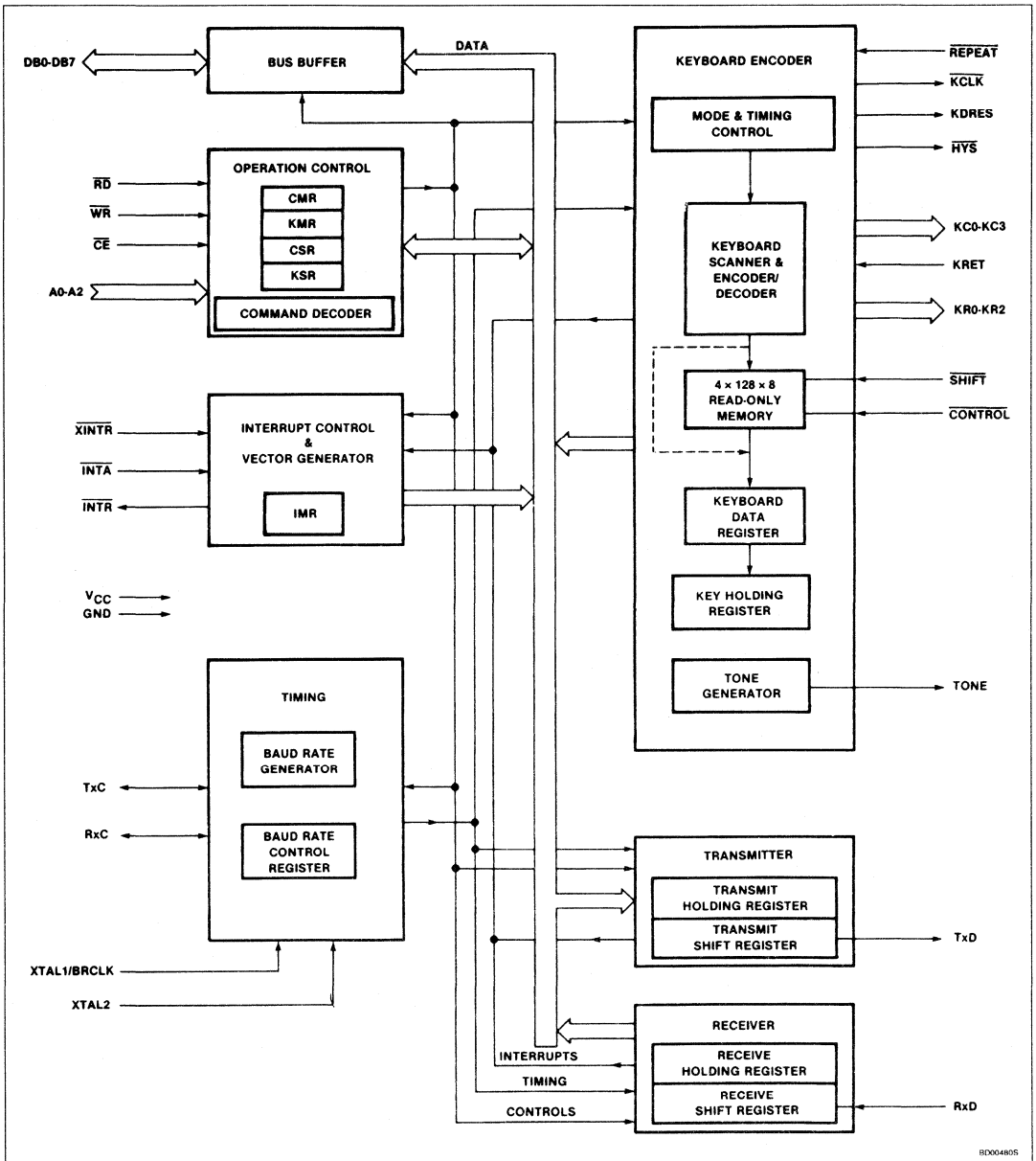


Figure 3. 2671 Programmable Keyboard and Communications Controller

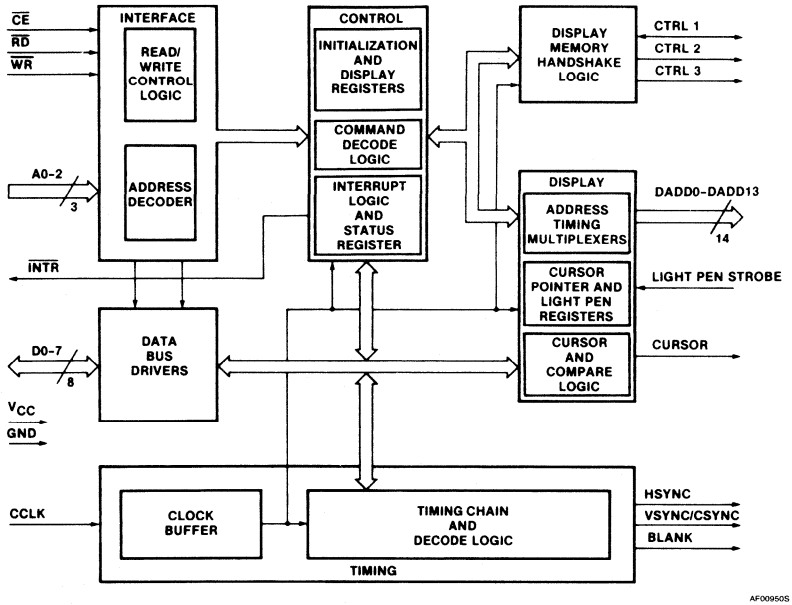


Figure 4. 2672 Programmable Video Timing Controller

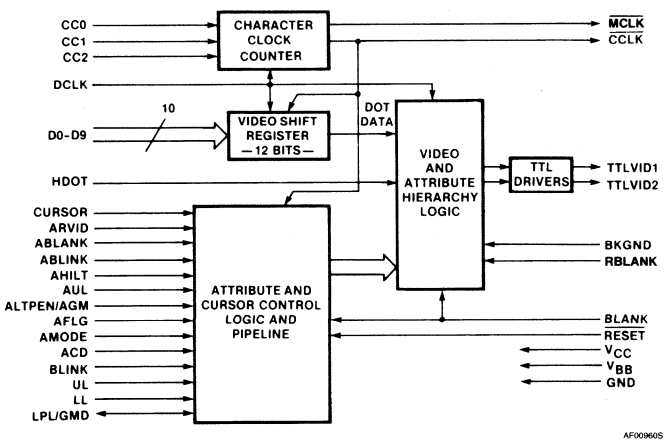


Figure 5. 2677 Video and Attributes Controller

2. Read/Write at cursor address (with optional increment of address).
3. Write from cursor address to pointer address.

The operational sequence for a write to memory operation is:

1. The CPU loads data to be written into the display memory into the interface latch.
2. The CPU writes the destination address into the PVTC's cursor or pointer registers.
3. The CPU checks the PVTC 'RDFLG' status bit to assure that any previous operation has been completed.
4. The CPU issues a 'write at cursor with/without increment' or a 'write at pointer' command to the PVTC.

Using the 2670/71/72/77 CRT Terminal Chip Set AN401

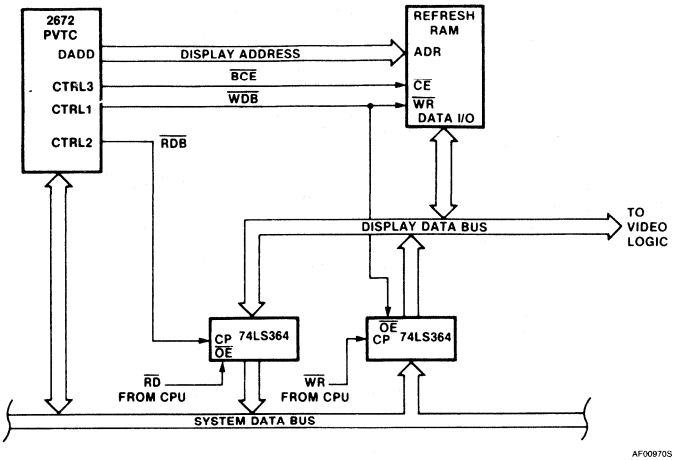


Figure 6. Independent Buffer Mode Configuration

5. The PVTC negates 'RDFLG', outputs the specified address, and generates control signals to perform requested operation. Data is copied from the interface latch into the memory.
6. The PVTC sets its 'RDFLG' status to indicate that the write operation is completed.

Similarly, a read operation proceeds as follows:

1. Steps 2 and 3 as above.
2. The CPU issues a 'read at cursor with/without increment' or 'read at pointer' command.
3. The PVTC negates 'RDFLG', outputs the specified address, and generates control signals to perform the read operation. Data is copied from the memory to the interface latch and the PVTC sets its 'RDFLG' status to indicate that the operation is completed.
4. The CPU checks the 'RDFLG' status to see if the read is completed.
5. The CPU reads the data from the interface latch.

Loading the same data into a block of display memory is accomplished via the 'write from cursor to pointer' command:

1. The CPU loads the data to be written into the display memory into the interface latch.
2. The CPU writes the beginning address of the memory block into the PVTC's cursor address register and the ending address of the block into the pointer address register.
3. The CPU checks the 'RDFLG' status bit to assure that any previous operation has been completed.
4. The CPU issues a 'write from cursor to pointer' command to the PVTC.
5. The PVTC negates 'RDFLG'' and outputs block addresses and control signals to copy the data from the interface latch into the specified block of memory.
6. the PVTC sets its 'RDFLG' status to indicate that the block write is completed.

Similar sequences can be implemented on an interrupt driven basis using the READY interrupt output from the PVTC to inform the CPU that a previously requested command has been completed.

Two timing sequences are possible for the 'read/write at cursor/pointer' commands. If the command is given during the active display window (defined as first scan line of the first character row to the last scan line of the last character row), the operation takes place during the next horizontal blanking interval. If the command is given during the vertical blanking interval, or while the display has been commanded blanked, the operation takes place immediately.

For the 'write from cursor to pointer' operation, the PVTC's BLANK output is asserted automatically and remains asserted until the horizontal retrace interval following completion of the command. The memory is filled at a rate of one location per two character times, plus a small amount of overhead.

Shared and Transparent Buffer Modes

In these modes the display buffer RAM is a part of the CPU memory domain and is addressed directly by the CPU. Both modes use the same hardware configuration with the CPU accessing the display buffer via three-state drivers (see figure 7). The processor bus request ($\overline{PBREQ}$) control signal informs the PVTC that the CPU is requesting access to the display buffer. In response to this request, the PVTC raises bus acknowledge ($\overline{BACK}$) until its bus external ($\overline{BEXT}$) output has freed the display address and data busses for CPU access. $\overline{BACK}$, which can be used as a 'hold' input to the CPU, is then lowered to indicate that the CPU can access the buffer.

In transparent mode, the PVTC delays the granting of the buffer to the CPU until a vertical or horizontal blanking interval, thereby causing minimum disturbance of the display. In shared mode, the PVTC will blank the display and grant immediate access to the CPU.

Row Buffer Mode

Figure 8 shows the hardware implementation for the row buffer mode. During the first scan line (line 0) of each character row, the PVTC halts the CPU and DMA's the next row of character data from the system memory to the row buffer memory. The PVTC then releases the CPU and displays the row buffer data for the programmed number of scan lines. The bus request ($\overline{BREQ}$) control signal

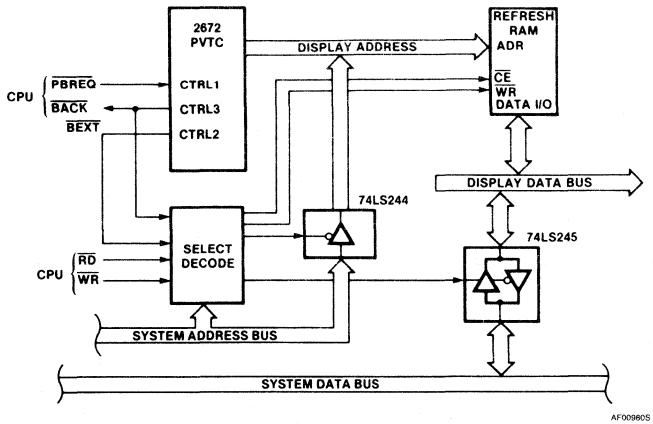


Figure 7. PVTC Shared or Transparent Buffer Modes

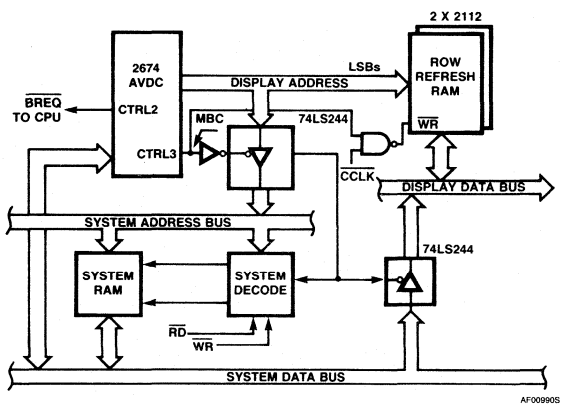


Figure 8. Row Buffer Mode Configuration

informs the CPU that character addresses and the memory bus control (MBC) signal will start at the next falling edge of BLANK. The CPU must release the address and data busses before this time to prevent bus contention. After the row of character data is transferred to the CPU, BREQ returns high to grant memory control back to the CPU.

A MINIMUM CHIP COUNT TERMINAL IMPLEMENTATION

Figure 9 is the schematic of a minimum chip count CRT terminal using the four CRT set devices. Only 15 IC packages are required for the complete implementation, including all keyboard encoding and RS-232 level conversion for the serial interface. Despite this low chip count the terminal is capable of providing an impressive array of features including:

Display Format:
- 24 or 25 character rows
- 80 characters per row

Character Format:
- 7 x 9 dot matrix character in a 9 x 12 character block
- 96 ASCII alphanumeric characters
- 32 special symbols
- Block graphics
- Line drawing character set

Cursor:
- Underline or block cursor
- Optional blinking

Keyboard:
- 128 keys maximum
- Non-encoded
- Cursor control keys
- Numeric keypad

Serial Interface:
- Full or half duplex
- RS-232 compatible
- 16 baud rates with internal baud rate generator
- Character or block transmission

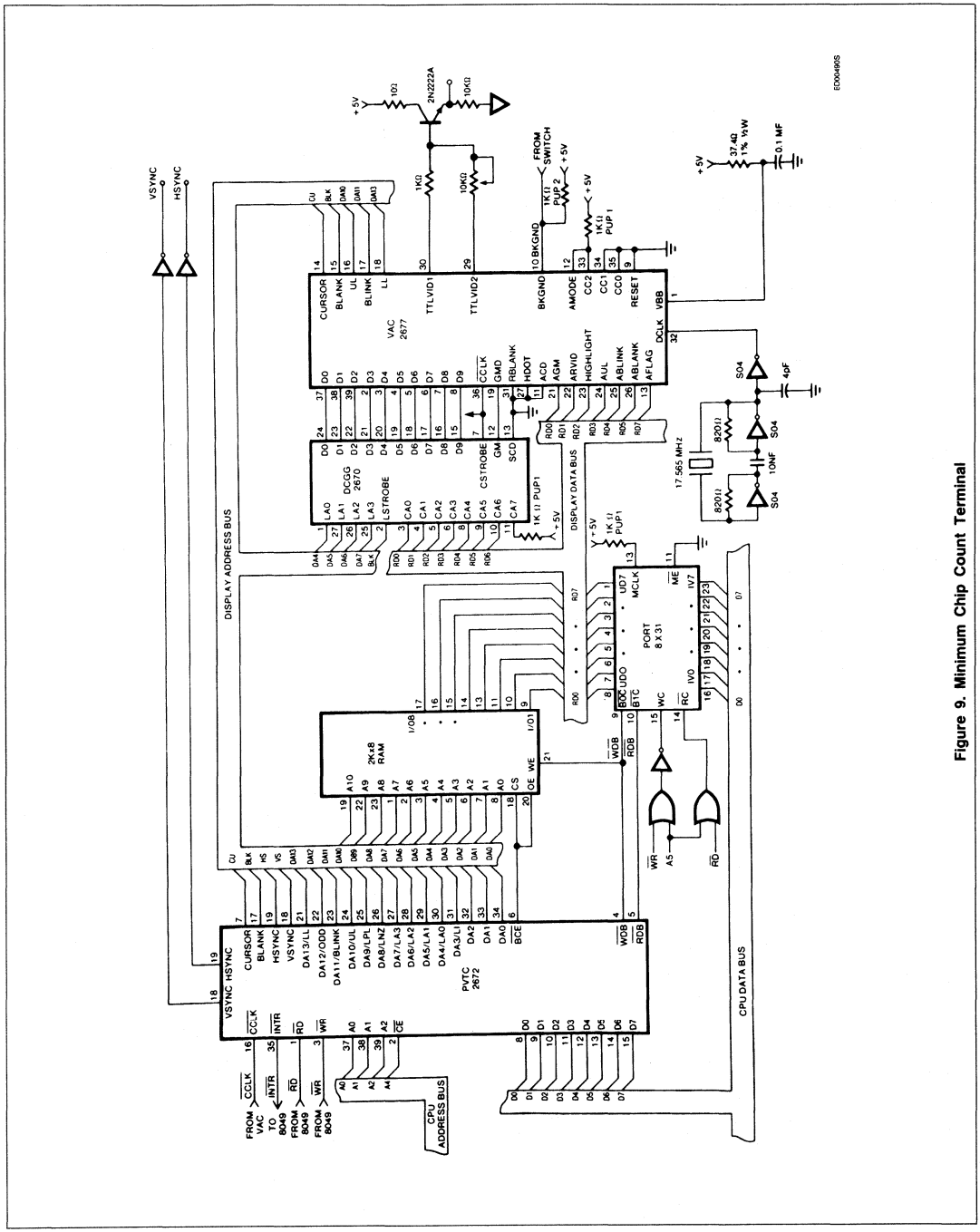


Figure 9. Minimum Chip Count Terminal

Using the 2670/71/72/77 CRT Terminal Chip Set AN401

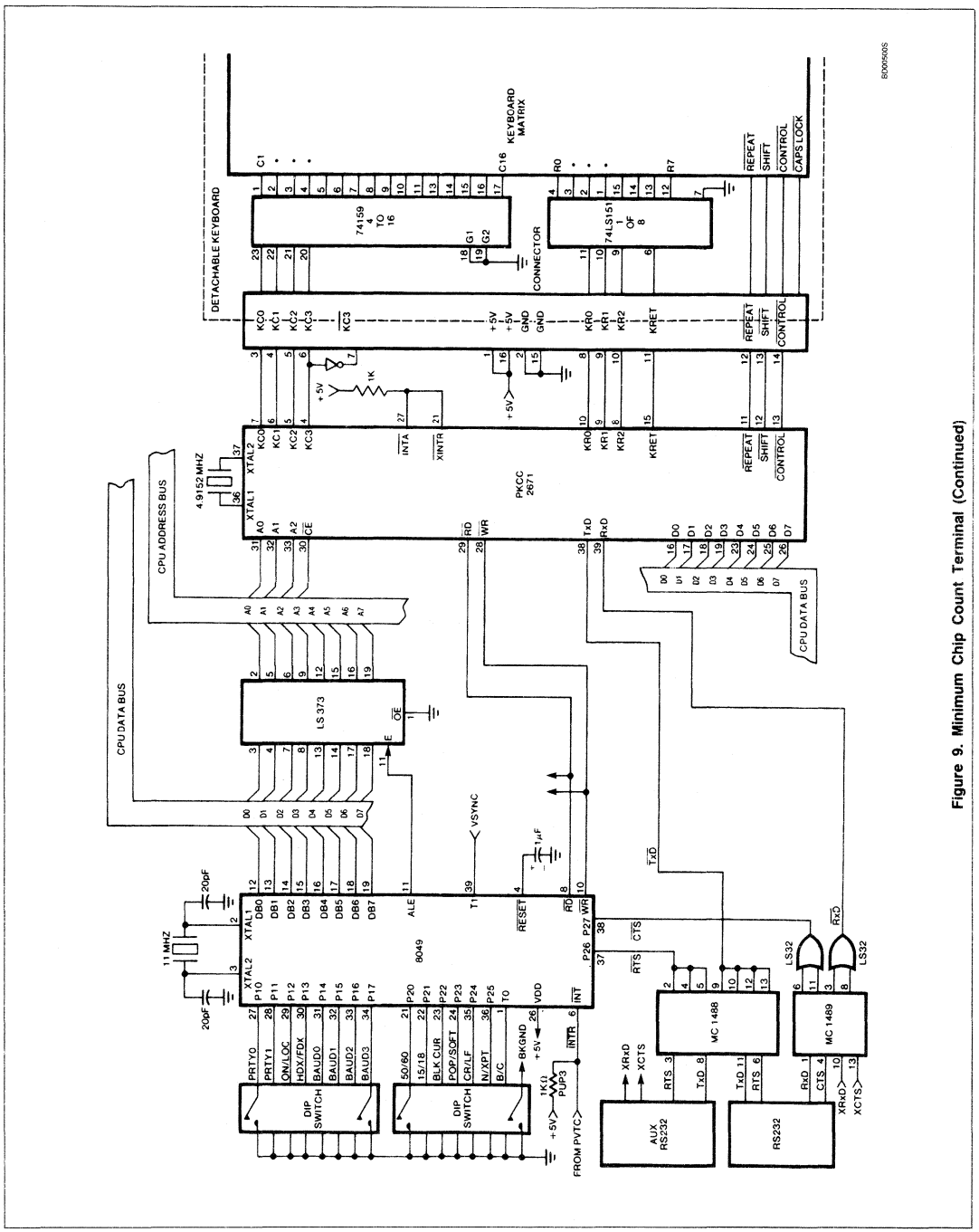


Figure 9. Minimum Chip Count Terminal (Continued)

Using the 2670/71/72/77 CRT Terminal Chip Set AN401

Operating Modes:
- Normal
- Transparent (displays graphic and control characters)
- Page or scroll with optional smooth scroll

Visual Attributes:
- Blink
- Reverse video
- Highlight
- Underline
- Non-display

The system utilizes the independent buffer mode to minimize hardware requirements. The dual port interface to the 2K x 8 display buffer is via a Signetics 8 x 31 bidirectional latch. This may be replaced by a unidirectional latch such as the 74LS374 if reading of the RAM's contents by the CPU is not required.

The operating program for the terminal is contained in the internal ROM of the 8049 microcomputer, which also provides the RAM required by the system program. Since the majority of the terminal's features are tailored by firmware, the ROM size can be increased, either internally or externally, to support additional functions.[1]

BASIC TERMINAL SOFTWARE

The software for a microcomputer based terminal is closely tied to the system hardware configuration and its characteristics. If an interrupt driven mode of operation is desired, the system hardware/software design must be capable of prioritizing the interrupts so that the system will correctly service interrupts from different sources. In a typical system, there are three interrupt sources: the keyboard, the communications interface, and the video timing controller. The latter must usually be assigned the highest priority since failure to service an interrupt from the video timing controller on a timely basis may result in visual perturbations on the display. The keyboard and datacomm interrupts can, in most cases, absorb some time delay before they are serviced since they include one or more levels of data buffers.

Often, a multi-level interrupt structure will be required so that a high priority interrupt requiring immediate service can be serviced even while the system is in the process of servicing a lower priority interrupt.

A simplified flowchart for the software for an interrupt driven terminal is shown in figure 10. After application of power, the microprocessor first performs a system initialization routine which consists of five parts:

1. Clear the microcomputer's scratchpad RAM.
2. Initialize the 2672 PVTC for the desired screen format, monitor timing parameters, cursor parameters, and display start address.
3. Clear the CRT display by loading a non display-code (usually an ASCII 'space', 20 hex) into the buffer memory.
4. Initialize the 2671 PKCC for the desired keyboard and serial interface modes.
5. Read any mode switches (e.g., full or half duplex, baud rate, cursor type, etc.) and set system parameters as required.

The processor can now enable its interrupts and wait in a loop until an interrupt is received. When this happens, the processor first determines the source of the interrupt and then performs the required system operation.

An interrupt from the CRT timing controller usually indicates that some information is required for proper screen refresh operation. For example, the PVTC may issue a 'split screen' interrupt to indicate that a new address must be loaded into its screen start registers in order for the next character row to be displayed from other than the next sequential address in memory. The CPU must service this interrupt within a finite time in order for the display to operate correctly.

An interrupt from the keyboard interface may be either a displayable character or a control function. Displayable characters are usually transmitted to the host computer and also placed into the buffer memory for display on the terminal. Certain control characters, such as cursor control keys or keyboard error codes, may cause only local actions, while others will also require transmission to the host.

An interrupt from the data communications interface may also be a displayable character or a system control character. In either case the microprocessor must determine the type of character and perform the necessary system operation.

A DESIGN EXAMPLE

A fully operational emulation of an IBM 3101 terminal was designed and constructed using the Signetics CRT chip set. The terminal incorporates the majority of the 3101's functions. Selected functions were not incorporated due to program memory limitations. For example, the tabbing functions were developed and tested but were left out in deference to the block transmission functions.

Table 1. TERMINAL FEATURES

Display Screen Format
- — 2000 character screen capacity (25 rows x 80 columns)
- — Operator information area (25th line)
- — Block-shaped cursor with optional blinking

Displayable Graphic Set
- — 95 ASCII characters for non-transparent mode
- — 128 characters for transparent mode
- — 7 x 9 character matrix in 9 x 12 field

Keyboard
- — 63-key main keyboard
- — 12-key control key cluster
- — 12-key numeric keypad
- — Keyboard lock/unlock under software control
- — Keyboard clicker
- — Typamatic operation

Edit Functions
- — Cursor controls: up, down, left, right, home
- — Cursor address read and write
- — Erase functions: erase EOL, erase EOS, clear screen

Visual Attributes
- — Highlighted field
- — Blinking field
- — Non-displayed field
- — Underlined field

Modes of Operation
- — Transmission modes: character or block (page or line)
- — Normal or transparent

Line Protocol
- — Asynchronous
- — 7-bit ASCII with programmable parity
- — One or two stop bits
- — Full or half duplex
- — Online or local
- — Programmable line turnaround character for block mode (EOT/ETX/CR/XOFF)
- — EIA RS232 interface
- — Communication line speed: 50 to 9,600 baud

Screen Refresh Rate
- — 60Hz

More features could have been included by selecting another of the numerous microprocessor devices on the market with greater program memory capacity. Major features of

[1] A data package for this terminal is available upon request from Signetics.

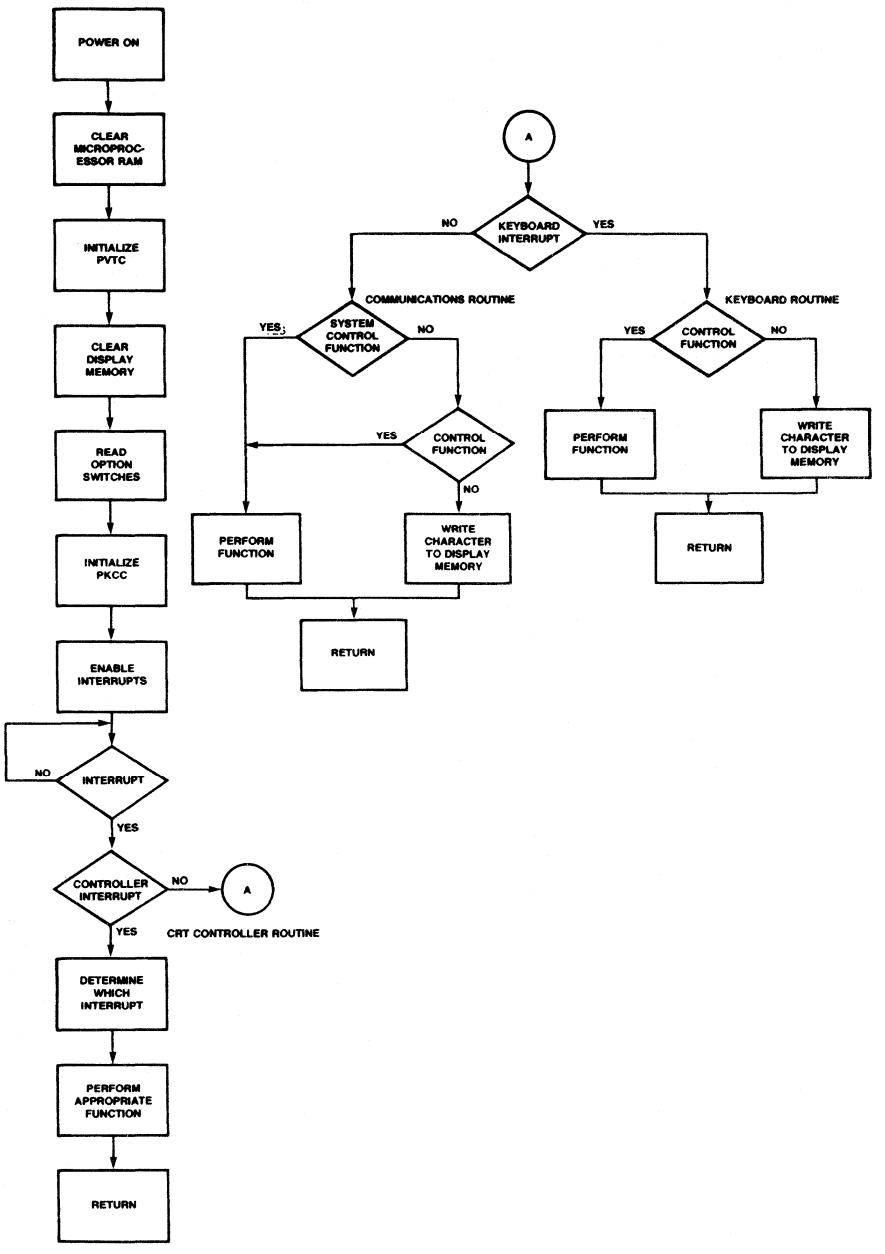


Figure 10. Basic Terminal Software Flowchart

Using the 2670/71/72/77 CRT Terminal Chip Set

AN401

the terminal are summarized in table 1.[1]

Terminal Hardware

The block diagram of the 8035 based terminal is illustrated in figure 11. It is an expanded version of the logic shown in figure 9, the major difference being a larger display RAM, to provide up to two pages of screen data, and the addition of several input ports to handle the large number of option and set-up switches. The terminal's software is contained in 4K of program storage external to the 8035.

The 2672 PVTC is programmed to operate in the independent buffer mode with the CPU isolated from the display RAM by two 74LS364 eight-bit latches, which provide the path for data transfers between the CPU and RAM. The PVTC, responding to commands from the CPU, completely controls the data transfer. To avoid display interference, the PVTC is instructed to complete the access during a blanking interval. For massive display updates (clear screen, load form, etc.) the PVTC is instructed to blank the display and service the data transfer immediately and continuously. Additional memory contention circuitry is not necessary since the PVTC provides all of the timing and addressing (via cursor and pointer) necessary to complete the transfer. An interrupt from the PVTC informs the CPU when an operation is completed.

The PVTC addresses the display buffer memory, which contains both character and attribute data. An attribute byte is identified by the software by setting bit 7 of the byte to a logic 1. The RAM data outputs are applied to the 2670 DCGG, which provides the character dot data information, and to the 2673 VAC.

The VAC is hardwired to operate in the field attributes mode for this application. An attribute character occupies a screen position but is not displayed unless the ACD input to the VAC is asserted. Bit 7 of the character byte identifies a character as an attribute character if it is a 1. When bit 7 on the RAM data bus is a 1, the attribute byte is latched into the VAC to begin a new attributes field. Since the attributes are double buffered in the VAC, only one byte (at any character position) is required to specify a field.

The bipolar VAC circuit serializes the dot data from the DCGG into a 17.5MHz data stream for the monitor. Two TTL-level video outputs provide three levels of video: black, white, and gray.

The PKCC provides the asynchronous data communications link at one of sixteen selectable baud rates. The PKCC addresses two 74LS145s which act as a 4-to-16 decoder to drive a 16 x 8 matrix keyboard. Key depressions are detected on the KRET input from a 74LS151 8-to-1 multiplexer. Each key depression is debounced, encoded according to the states of the SHIFT and CONTROL inputs, and presented to the CPU. Repeat and 'typomatic' (auto-repeat) functions are processed automatically by the PKCC.

Memory Allocation

The 4K bytes of available buffer memory were allocated as follows (all addresses are in hex):

- 0000 to 004F: display data for row 25, status line
- 0050 to 0075: not used
- 0076 to 007F: CPU scratchpad
- 0080 to 07FF: display data for rows 1 to 24
- 0800 to 0FFF: not used, available for second page of display data

The PVTC's 'display buffer first address' and 'display buffer last address' registers are loaded with the values 0080 and 07FF respectively so as to cause this portion of the RAM to act as a circular buffer. Initially the display data is organized in the RAM as follows:

- 0080 to 00CF: row 1 data
- 00D0 to 011F: row 2 data

  •
  •
  •
- 07B0 to 07FF: row 24 data

When a scroll operation is required, the CPU changes the value in the PVTC's 'screen start' register from 0080 to 00D0. This effectively shifts the displayed data up one row. Upon reaching the specified last buffer address (which is now the last character in row 23), the PVTC automatically changes the addressing sequence to resume starting at 0080 for the 24th row. The display data is now organized:

- 00D0 to 011F: row 1 data
- 0120 to 016F: row 2 data

  •
  •
  •
- 07B0 to 07FF: row 23 data
- 0080 to 00CF: row 24 data

The CPU can clear the previous data in 0080 to 00CF so that a blank row appears in the 24th position.

The status line (row 25) data is kept in a separate section of RAM to eliminate the necessity of moving the data whenever the scrolling operation described above occurs. Thus, the PVTC must be instructed to change its addressing sequence at the beginning of the 25th row. This is accomplished by use of the split screen row interrupt capability. IR10, the 'split screen interrupt row' register, is initialized so as to cause an interrupt to be issued at the beginning of row 24. The CPU responds to this interrupt by changing the value in the screen start register to 0000. The PVTC then uses this value as the starting address of the next (25th) row, causing the status line to be displayed in that position. The CPU must re-load the screen start register before the end of the vertical blanking interval with the correct value for the first character to be displayed on the screen.

Terminal Software

Because the 8035 microcomputer used in the terminal provides only a single interrupt level, a totally interrupt driven software design could not be used. The interrupt was assigned to the PVTC to service the split screen interrupt described above and the operations required to implement the smooth scroll feature. The keyboard and datacomm functions are serviced by polling the PKCC status register. Both the keyboard interface and UART receiver are double buffered in the PKCC, preventing overrun even if they are not serviced immediately.

The program generally follows the typical program flow described previously. At system reset the 8035 interrupts are disabled, data memory and display memory are cleared to zeroes, and both the PVTC and PKCC are master reset through software commands. The system option switches are then read and stored and the PVTC and PKCC internal registers are initialized for the selected operation. Finally, the initial data for the status line is loaded, the PVTC, UART, and keyboard are enabled, and the CPU interrupt is enabled. The program then enters a loop where the PKCC is checked for keyboard or UART entries. If an entry has occurred, the character is fetched and stored in a software controlled FIFO (first-in-first-out) memory which is

[1] A data package for the design, including details of operation, schematic, and program listing, is available upon request by writing to:
Signetics Corporation
Microprocessor Applications Dept.
Mail Station 2576
P.O. Box 3409
Sunnyvale, CA 94088-3409

Using the 2670/71/72/77 CRT Terminal Chip Set AN401

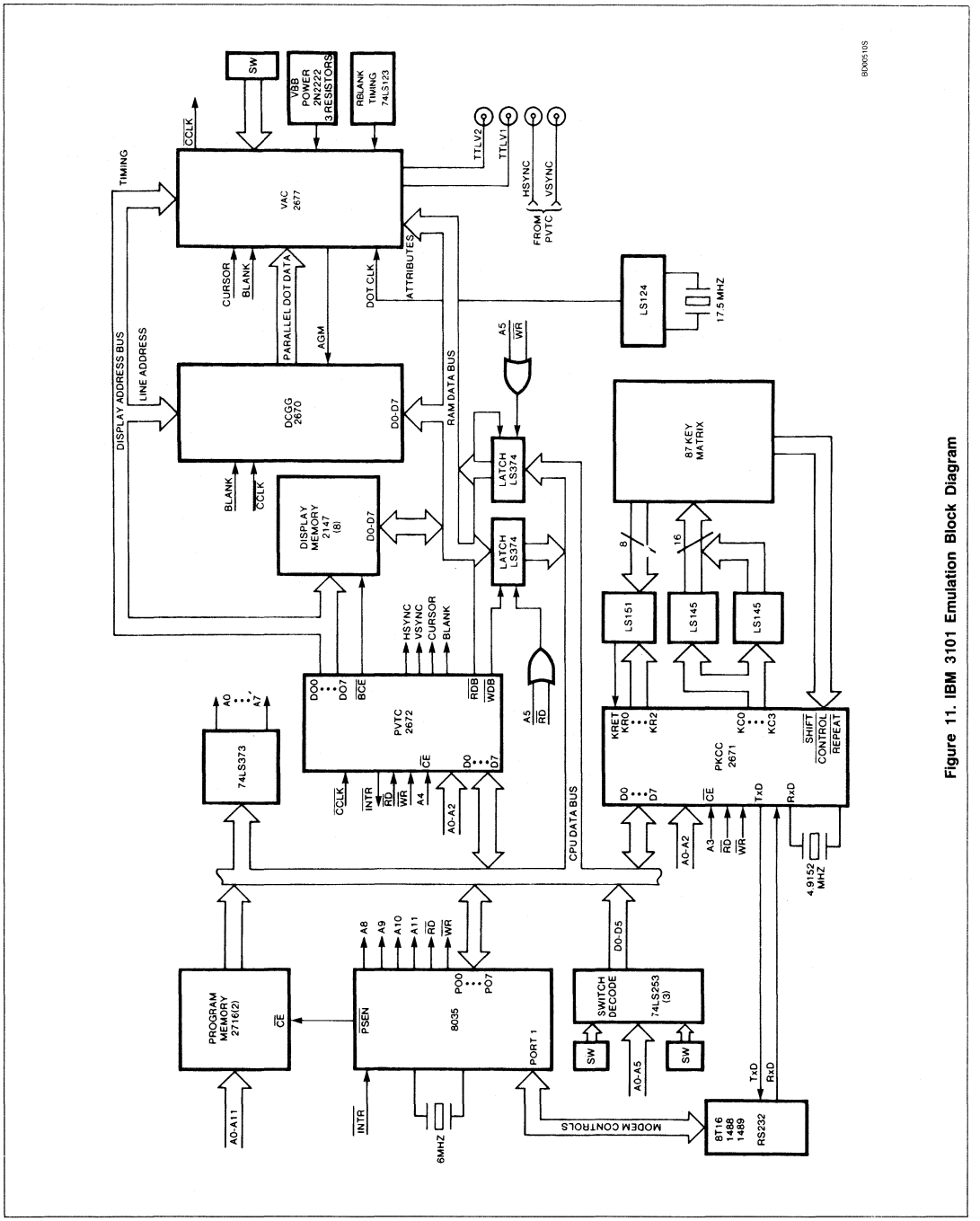


Figure 11. IBM 3101 Emulation Block Diagram

eight bytes deep for both receiving or transmitting characters (the need for the FIFO is described below). If either FIFO has an entry, the program proceeds to a character recognition routine which checks for the type of character (displayable or control) and the appropriate handling subroutine (ESC sequence, control sequence, cursor control, character display, etc.) is called. If the FIFO's are empty, the polling routine checks the option switches for any changes since reset entry and if so reconfigures the system as necessary.

The need from the FIFOs results from the method used to effect the clear row function required when a scroll is performed. Although the PVTC includes a 'clear from cursor to pointer' command that can be used to clear a block of memory rapidly, the display is temporarily blanked during this operation. This would cause undesirable flashes on the display. Instead, the program does the function by a repetitive loop using the 'write at cursor and increment' command. Since the write occurs only once per scan line during the active display window, a worst case total of approximately 80 scan line times is required to execute the routine. This would limit the maximum received character rate to approximately one per 80 scan lines or about 240 characters per second (2400 baud). To overcome this limitation, the PKCC is also polled each time through the clear line subroutine loop, and any entries from the receiver or keyboard are stored in the appropriate FIFO. Since the FIFO is eight deep, this allows eight characters to be received in the same time, increasing the maximum baud rate to 19,200. (Other program limitations actually reduce the maximum baud rate to 9600 baud). However, this does not increase the rate at which characters which cause a scroll function to occur, such as a line feed, can be received. Each character of this type must be followed by 'fill' characters in order for data rates higher than 2400 baud to be used.

An interrupt from the PVTC will occur when the display scan reaches the row count programmed in its split screen address register, row address 24 (for the 24th row). In response to the interrupt, the CPU loads the screen start registers with the address of the status line (0000) and enables the PVTC's line zero interrupt. This causes another interrupt at the beginning of display of the status line. At this time the CPU reloads the screen start register with the proper address to begin the next display frame and disables the line zero interrupt.

If scrolling is required the screen start register value is incremented by 80 (popping off the top row) and the effective bottom row cleared to nulls. If soft scrolling is selected, additional functions are performed during the interrupt routines. To begin the operation, the line zero interrupt routine adds ten lines to the vertical back porch. This causes the next active screen display to begin ten scan lines later than normal and gives the effect of the display moving up two scan lines (12 lines per character row – 10) instead of jumping up 12 lines. If nothing else were changed, however, the bottom of the display would move down ten lines. thus, during the row 24 interrupt the number of scan lines per character row is changed to two (12 – 10), causing only the first two scan lines of that row to be shown. The next line zero interrupt (at row 25) restores the lines per row count back to 12 to keep the whole status line showing, and now changes the vertical back porch to 8. The display moves up two more scan lines and at the next row 24 interrupt four scan lines are shown. The process continues in this manner, providing the effect of the entire display, except for the status line, smoothly scrolling up over a selected interval of six frames, or one tenth of a second.

Timing Calculation Example

The AVDC register values need to be calculated to provide suitable drive signals for the selected monitor. In the following example,

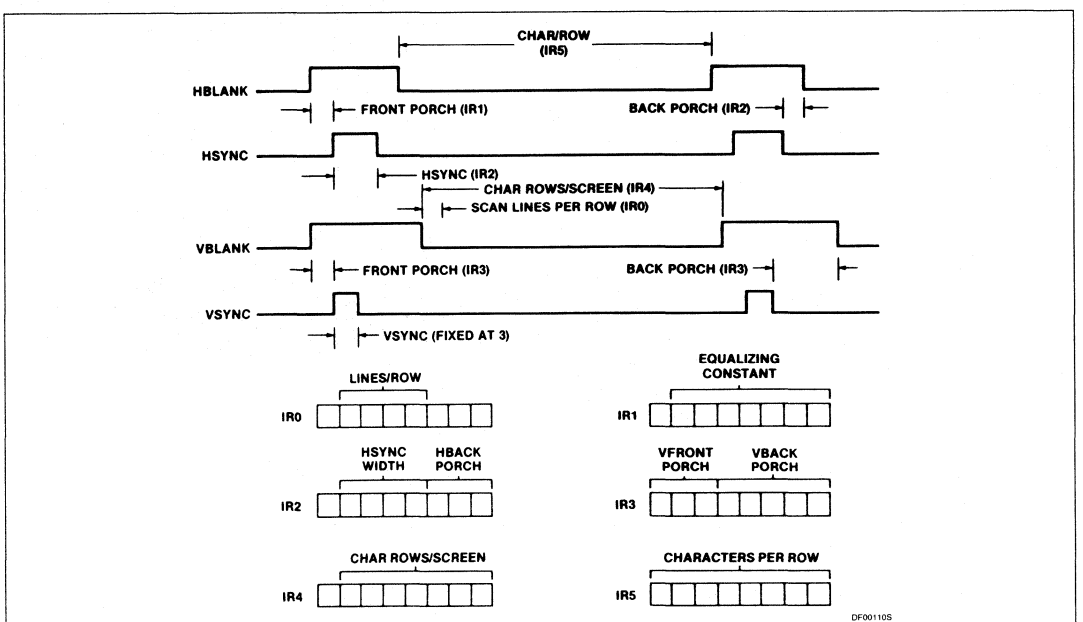


Figure 12. Horizontal and Vertical Timing

each character will be contained in a 8 dot by 10 line field with a total of 24 rows displayed.

The total number of active scan lines will be 10 x 24 or 240. An estimate of the horizontal refresh rate required for the monitor can be made using the following formula assuming a screen refresh rate of 60Hz with a vertical retrace of 10 percent of the active scan lines:

Horizontal frequency =
(60) x (240) x (1.1) = 15,840Hz

In order to calculate the IR register values of the AVDC, the CRT monitor characteristics need to be obtained. An NEC monitor was used for the worksheet. The major timing specifications for the monitor are:

Horizontal frequency:	15.75KHz
Horizontal retrace:	15.5μs
Horizontal sync width:	4.76μs
Vertical frequency:	60Hz
Vertical retrace:	1.43ms
Vertical sync width:	0.19ms

Monitor timing definitions are shown in figure 12. The worksheet illustrated in table 2 can be used to compute the required timing and associated PVTC register values. Some rough guesses are required initially and several iterations through the worksheet may be required.

The worksheet can be used to determine the dot clock frequency required given the monitor characteristics above, the number of characters per row, the number of dots per character, the number of character rows per screen, and the number of scan lines per character row. The horizontal sync width was estimated by the ratio of the sync width time with respect to the retrace time given in the monitor information above. The horizontal and vertical front and back porches were estimated from the remaining time left during the retrace.

Table 2. TIMING WORKSHEET

1.	Horizontal Character Block (number of dots)	8	
2.	Vertical Character Block (number of scan lines)	10	(IR0)
3.	Vertical Refresh Rate, Hz	60	
4.	Horizontal Line Rate, KHz	15.75	
5.	Characters Per Row	64	(IR5)
6.	Character Rows Per Screen	24	(IR4)
7.	Total Active Video Scan Lines (step 2 X step 6)	240	
8.	Total Scan Lines Per Frame (step 4 / step 3)	262.5	
9.	Vertical Retrace Interval (step 8 – step 7)	22.5	
10.	Vertical Front Porch (number of scan lines)	8	(IR3)
11.	Vertical Back Porch (number of scan lines)	12	(IR3)
12.	Vertical Sync Width (number of scan lines)	3	
13.	Horizontal Scan Line Rate, us (1 / step 4)	63.5	
14.	Horizontal Active Video Time, us	48	
15.	Horizontal Retrace Time, us	15.5	
16.	Horizontal Retrace Time (character time units) (step 15 / [step 14 / step 5])	20.7	
17.	Horizontal Back Porch (character time units)	7	(IR2)
18.	Horizontal Sync Width (character time units)	6	(IR2)
19.	Horizontal Front Porch (character time units)	8	
20.	Total Character Time Units in One Horizontal Scan Line (add steps 5, 17, 18, and 19)	85	
21.	Equalizing Constant ([step 20 / 2] – [2 X step 18])	31	(IR1)
22.	Character Clock Rate, MHz (step 20 X step 4)	1.339	
23.	Character Period, us (1 / step 22)	0.747	
24.	Scan Line Period, us (step 23 X step 20)	63.50	
25.	Dot Clock Rate, MHz (step 22 X step 1)	10.712	

AN403
2670/71/72/73 CRT Set Application Briefs

Application Note

INTRODUCTION

The Signetics CRT chip set consists of four LSI devices which, when combined with standard microcomputer, memory, and TTL products, permit the implementation of a CRT terminal in as few as 15 total packages. The four LSI devices are:

2670 Display Character and Graphics Generator (DCGG)

The 2670 is a mask programmable line select character generator which contains the dot patterns for 128 10 x 9 characters. It also provides a semi-graphics capability wherein the 8-bit character code is translated directly into 256 graphic patterns useful for presenting data such as graphs and forms on the CRT display. Additional features of the DCGG include character and line address latches, internal descend logic and thin line graphics.

2671 Programmable Keyboard and Communications Controller (PKCC)

The 2671 provides a versatile keyboard interface and an asynchronous communications interface in a single package. The keyboard section handles the scanning, debounce, and encoding of mechanical or capacitive keyboards with up to 128 keys utilizing any of four programmable rollover modes. An internal ROM provides any of four key codes for a depressed key. The communications section is a universal asynchronous receiver and transmitter (UART) with programmable character length, parity, and stop bits. A baud rate generator providing 16 standard communications frequencies which operates directly from a crystal is also incorporated in the 2671.

2672 Programmable Video Timing Controller (PVTC)

The 2672 is designed for use in CRT terminals and other display systems that employ raster scan techniques. It generates the vertical and horizontal timing for the CRT monitor, and provides the addressing for the display buffer memory. The CPU to display buffer interface can be programmed for several different modes of operation, including full screen buffer, multiple page buffer, or row buffer, as required by the application. Programmable features of the PVTC include screen format (characters per row, scan lines per row, rows per screen), horizontal and vertical timing parameters, cursor type, interlaced or non-interlaced operation, and character and cursor blink timing.

2673 Video and Attributes Controller

The 2673 is a bipolar LSI device that contains the high speed timing circuits required in CRT terminal systems. Included on the 2673 are a dot clock counter, video shift register, field and character attributes logic, and cursor display circuits. The 2673 attributes capabilities are reverse video, character blank, blink, underline, highlight, light pen, and graphics. The device provides both TTL and analog video outputs and operates at dot frequencies up to 25MHz.

Individual data sheets are available which describe each of the devices in full detail. A previously published application note entitled "Using the 2670/71/72/73 CRT Terminal Chip Set" (AN401), describes the implementation of CRT terminals using the chip set. The purpose of this application note is to provide information on the implementation of special end-product features.

SMOOTH (SOFT) SCROLLING

Scrolling is used in CRT terminals to provide the effect of an "endless page" on which the data can be written. In normal implementations, once the screen fills up, the space for the next row of data is provided by removing the top row and moving all the remaining rows up by one row, thus creating a blank data row at the bottom of the screen into which the new information is placed. (The process may be reversed if the new data is to be written at the top of the screen). This technique creates a readability problem when viewing data which is being received at a relatively high speed, since the rows are jumping up (or down) at a fast rate. Smooth or soft scrolling improves readability by moving the data in scan line increments instead of in whole row jumps, thus creating the effect of a sheet of paper slowly being moved through the viewing area. One system restriction of providing the smooth scrolling feature is that the rate at which new rows of data can be received is limited to the rate at which the display is being moved. Thus, successive line feeds must be separated by a minimum number of real or dummy "fill" characters in order to allow the display to keep up with the received data. The number of fill characters will be a function of the number of scan lines per character row, the scroll rate, and the communications line speed, and may also be affected by the inclusion of software and/or hardware features such as a data buffer in the system design.

When using the CRT chip set, smooth scrolling can be implemented in software only, or with a combination of hardware and software.

Software Only Method

The software only method of smooth scrolling uses the 2672's capability to program the scan lines per row and the vertical back porch, and the ability to program an interrupt to occur at any row by programming the row value into the split screen register, IR10. Three limitations of this method are:

1. Scrolling must be in an upward direction.
2. The screen area that is scrolled must start at the top of the display but can end at any row.
3. The minimum scrolling increment is two scan lines.

2670/71/72/73 CRT Set Application Briefs

AN403

The flowchart in figure 1 shows the steps necessary for scrolling starting at the top of the screen and ending at character row 23[1], with an additional non-scrolling row at row 24. The IR10 value is set to 23, to cause an interrupt to occur at row 23. This interrupt routine then enables the line zero interrupt, so that another interrupt occurs at row 24. The line zero (row 24) interrupt routine disables the line zero interrupt so that another line zero interrupt is not asserted until the split screen (row 23) interrupt routine enables it again.

When a scroll is desired, the system software changes the screen start address to the new value (normally an address "n" characters higher than the previous value, where "n" is the number of characters per row) and sets a scroll flag. The top row disappears and normally the display would jump up by one row. However, the line zero interrupt routine notes that the scroll flag is set and adds the value "s-2" (s = scan lines/row) to the vertical back porch (IR3). This causes the next active screen display to begin "s-2" lines later than normal and gives the effect of the display moving up two scan lines instead of jumping a full row. If nothing else were changed, however, the bottom of the display would move down the same number of scan lines. Thus, during the split screen interrupt routine the number of scan lines per character row (IR0) is changed to two, causing only the first two scan lines of that row (which is the new row) to be displayed and maintaining the proper value for the total scan lines. The next line zero interrupt restores the lines per row value back to its normal state to display the whole of the last character row, and now decreases the vertical back porch increment by two. The display moves up two more scan lines and at the next split screen interrupt four scan lines are allowed. The process continues in this manner until the scroll is completed.

Note that the CPU must complete the split screen interrupt routine within two scan line times.

Hardware/Software Method

The hardware/software method of smooth scroll removes the restrictions of the software only method. The scrolling region can begin and end at any character row, scrolling can be performed in either the up or down direction, and the minimum scroll increment can be one scan line.

A block diagram of the required hardware for the case of full screen scroll is illustrated in figure 2. When scrolling is required, the CPU writes the number of scan lines to scroll into the 4-bit latch during the vertical retrace interval. The scan line adder detects when the sum of the offset and the scan line address is greater than the programmed number of scan lines per row and causes the RAM address to be automatically offset to the next character row by the n-bit RAM address adder. The CPU adjusts the value in the scan lines to scroll latch at desired intervals, e.g., each vertical retrace, to effect the smooth scroll. When the scroll is completed, the screen start address is changed to the new value required for the top character row.

If only a partial screen scroll is required, the hardware must be modified to cause the offset to be applied only during the scrolling area. The lines to scroll latch of figure 2 is replaced by the circuit shown in figure 3. The software is written with split screen interrupts at the row before the first row which is to be scrolled (interrupt 1) and at the last scrolled row (interrupt 2). The required program actions are as follows:

1. During vertical retrace, the screen start address for the non-scrolled region at the top of the screen is loaded and IR10 is loaded with the row number required for interrupt 1.
2. During interrupt 1, a new screen start address for the scrolled region is loaded (if required), IR10 is loaded with the value required for interrupt 2, and the lines to scroll latch is loaded with five bits of data consisting of the four bit lines to scroll value plus an asserted "scroll" bit. The hardware will cause the scan line offset to be applied to the adder at the beginning of the next character row, as required.
3. During interrupt 2, a new screen start address for the non-scrolled region at the bottom of the screen is loaded (if required), and the "scroll" bit is negated. The hardware will cause the scan line offset to be removed from the adder at the beginning of the next character row, as required. If the scroll region extends to the bottom of the visible screen, this interrupt is not required. Note that the time required to service this interrupt will determine the minimum number of scan lines per scrolling increment.

HORIZONTAL SCROLL

Horizontal scrolling allows the terminal to be used to read or write pages which are wider than the actual screen width. For example, if an actual page width of 132 characters is to be displayed on a terminal with a capacity of 80 characters per row, horizontal scrolling can be used to display any desired 80-character "window" of the 132-character row. The window can be moved in response to operator keyboard commands, allowing all 132 columns to be observed. The CRT set capabilities allow horizontal scrolling in single or multiple character increments to be implemented using software only. As described in the 2672 data sheet, changing the contents of the screen start address register during a particular row, say row "n", will cause the display of the next row, "n + 1", to begin from the new address. This feature, together with the capability to interrupt at every row via the line zero interrupt, can be used to update the contents of the screen start register once each row to effect the horizontal scroll. The software operations required are as follows (see figure 4):

1. During the vertical retrace interval, the screen start register is initialized with the starting memory address of the display page plus the desired horizontal scroll (in characters).
2. The line zero interrupt is enabled. During each interrupt service the value in the screen start address is incremented by the actual (not display) page width. This may be done either by referencing a table of starting addresses or by performing the required addition.

COLOR DISPLAY INTERFACE

Figure 5 illustrates the block diagram of a color monitor interface. Eight colors for foreground and background with three attributes are supplied. The system operates in the character attribute mode with a 16-bit word of data for each character: seven bits for character select, six bits for color select, and three bits for other attributes.

The two 74LS374s delay the color information by two CCLKs to allow for the two CCLK delays of dot data through the DCGG and VAC. The video output of the VAC selects foreground or background color (active dot or not) for each character cell via the 74LS157 multiplexer. A variable CCLK delay may be required to synchronize the delay of the color information through the latches to the delay of the video data from the VAC.

EXTERNAL VIDEO SYNC

Some applications require overlaying of characters on an existing video display. An example of this is the addition of subtitles to a picture display. Figure 6 illustrates a simple technique of externally synchronizing the 2672 PVTC to an external video source. The dot clock to the 2673 VAC is stopped (character clock falling edge) at the start of the PVTC's sync interval and restarted upon occurrence of the external sync signal. The sync timing programmed in the 2672 must be slightly faster than the external sync rate.

[1]Rows are numbered consecutively starting with row 0. Thus, row 23 is the twenty-fourth row of characters.

2670/71/72/73 CRT Set Application Briefs AN403

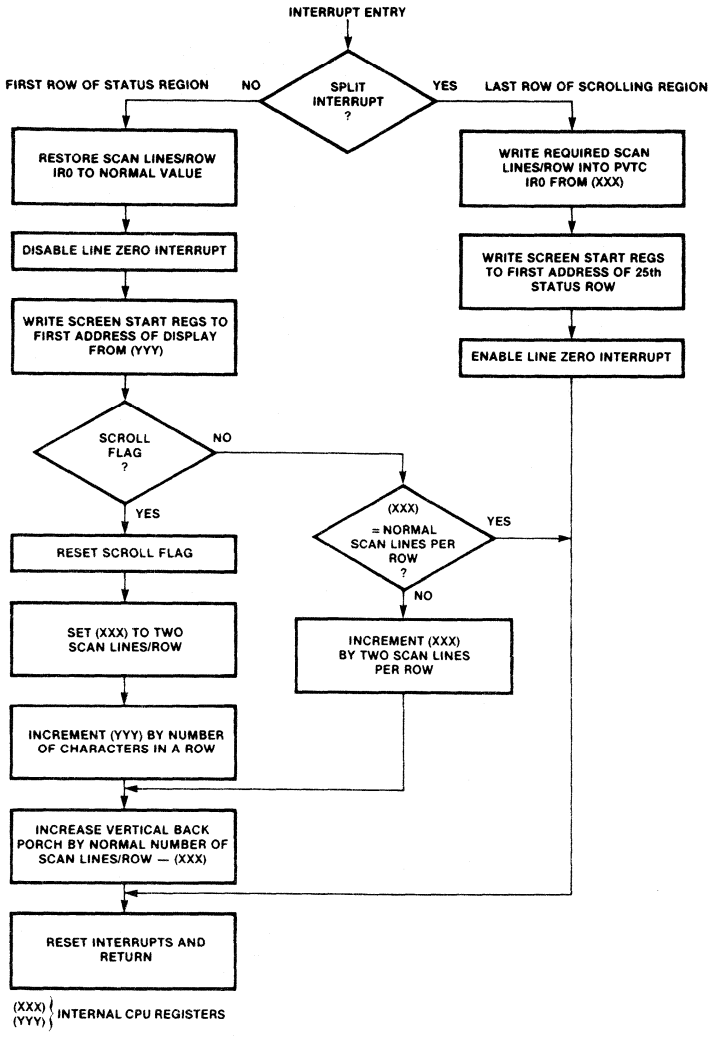
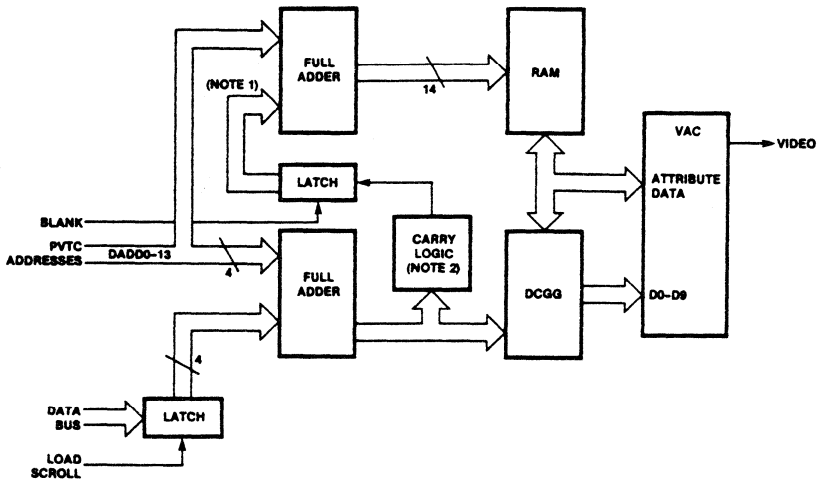


Figure 1. Smooth Scroll Flowchart



NOTES:
1. Hardwire adder inputs to number of characters per row.
2. Carry logic asserts output when output of adder exceeds normal scan lines/row.

Figure 2. Smooth Scroll Hardware Block Diagram

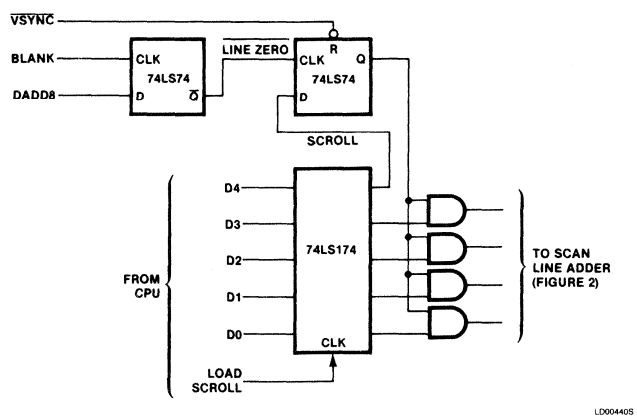


Figure 3. Partial Screen Smooth Scroll Modification

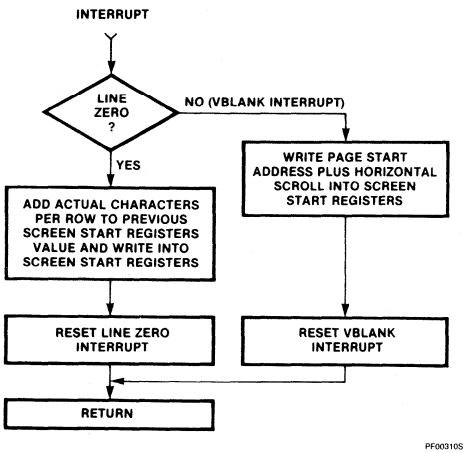


Figure 4. Horizontal Scroll

SCAN LINE COUNT GREATER THAN 16

Certain applications may require more scan lines than the 16 scan lines per character row (non-interlaced) which the 2672 can provide. Figure 7 shows the hardware required to obtain up to 32 scan lines per character row. The PVTC must be programmed for double height character rows. This causes the scan line count outputs from the 2672 (DADD4-DADD7) to increment once every two scan lines. The external flip-flop toggles each scan line to provide a fifth bit of scan line count information for the character generator. The technique permits any even value of scan lines per character row from 2 to 32 to be obtained.

BIT MAPPED GRAPHICS

Figure 8 illustrates an implementation of a bit mapped display with the chip set. In this configuration, the contents of the memory will be displayed without character generator translations. Thus, each bit in the memory corresponds to a single pixel on the display. Each horizontal scan line is defined by a contiguous set of bytes in the RAM. The data is written in groups of 8 bits by the CPU and is accessed by the PVTC in groups of 8 bits. The character generator of a normal alpha-numeric configuration is replaced by an 8-bit latch to implement the one CCLK delay normally provided by the 2670. The PVTC can be programmed for one scan line/row to cause the memory addressing to proceed without repetition of addresses in each row.

The PVTC can be programmed to a maximum of 256 characters/row and 128 rows/screen. Thus, a 2048 by 128 bit map is possible. If more than 128 dots are required vertically, the PVTC can be programmed for more than one scan line/row and the scan line outputs can be used as part of the RAM address, creating several segments of memory. For example, if 256 lines are required, the PVTC must be programmed for two lines/row. The use of LA0 as part of the memory address creates two segments. The CPU writes the data for odd scan lines in one segment of the memory and the data for even scan lines in the other. As the PVTC accesses the RAM the scan line count will go from 0 to 1 addressing the even and then odd portion of the RAM for each character row.

Figure 8 shows the DCGG as optional. By using the 2672's split screen capability and changing 2672 parameters, the CPU can enable the DCGG to be active for a portion of the screen thereby incorporating both bit-mapped and alphanumeric sections on the same display.

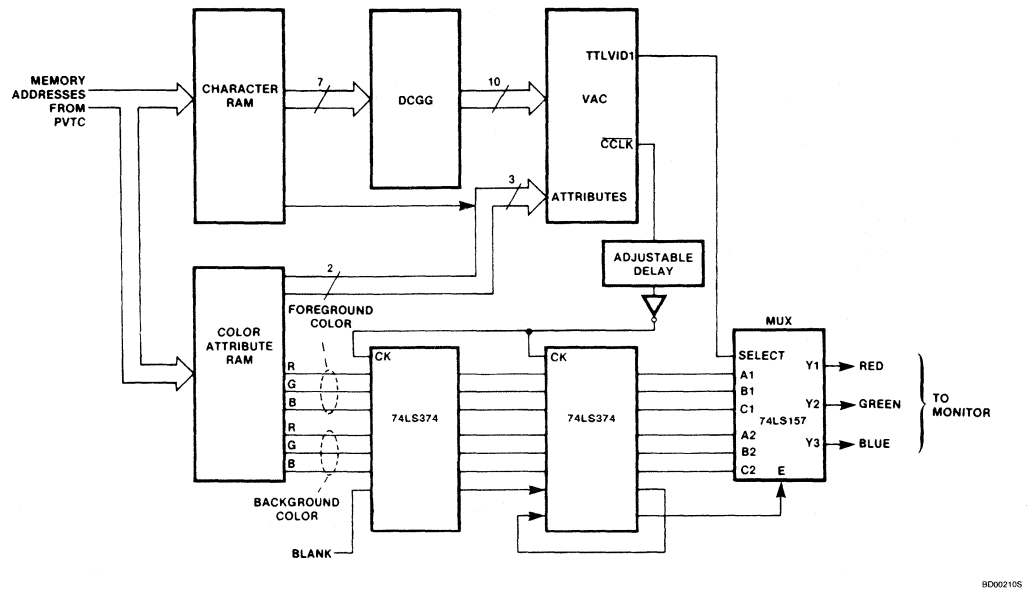


Figure 5. Color Display Block Diagram

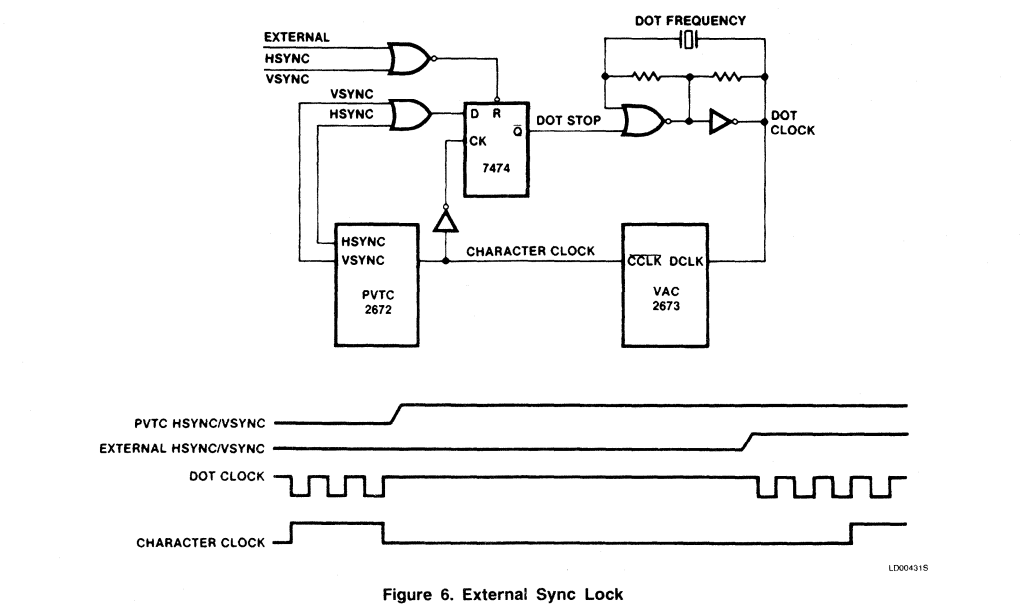


Figure 6. External Sync Lock

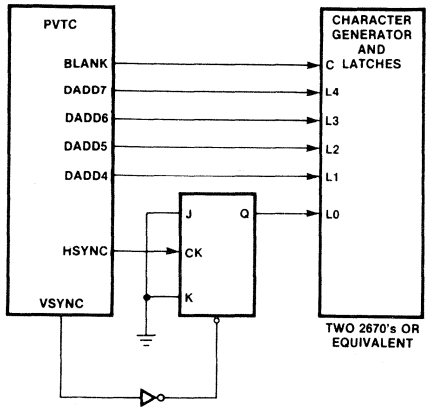


Figure 7. Greater Than 16 Scan Lines Per Row

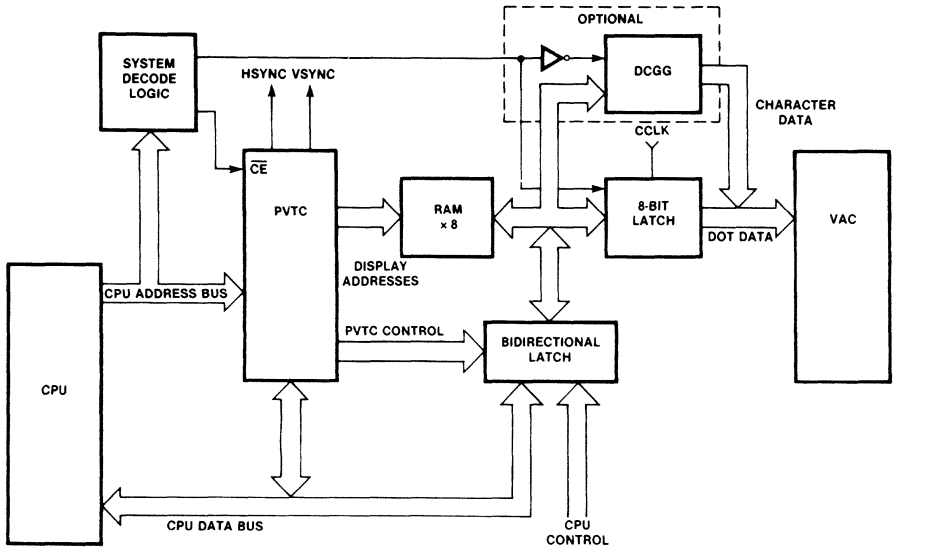


Figure 8. Implementation of Bit Mapped Display

PAL/NTSC COLOUR ENCODER

GENERAL DESCRIPTION

The TEA2000 is a monolithic integrated circuit, which encodes colour information and provides composite video output for driving a VHF or UHF modulator.

Features

- European PAL and American NTSC/M standard selectable
- Internal generation of burst timing and PAL-switch-function
- 6 bit binary TTL compatible input provides 64 different colours
- TTL compatible colour blanking input
- TTL compatible sync input

QUICK REFERENCE DATA

Supply voltage	$V_{11\text{-}9}$	typ.	12 V
Supply current at $V_{11\text{-}9}$ = 12 V	I_{11}	typ.	55 mA
Input voltage pins 1,2,3,4,5,14,16,17,18	V_{IL}	max.	0,8 V
	V_{IH}	min.	2,0 V
Composite video output (sync tip to white)	$V_{6\text{-}9(p\text{-}p)}$	typ.	2,0 V
Operating temperature range	T_{amb}		0 to + 70 °C

PACKAGE OUTLINE

18-lead DIL; plastic with internal heat spreader (SOT-102HE).

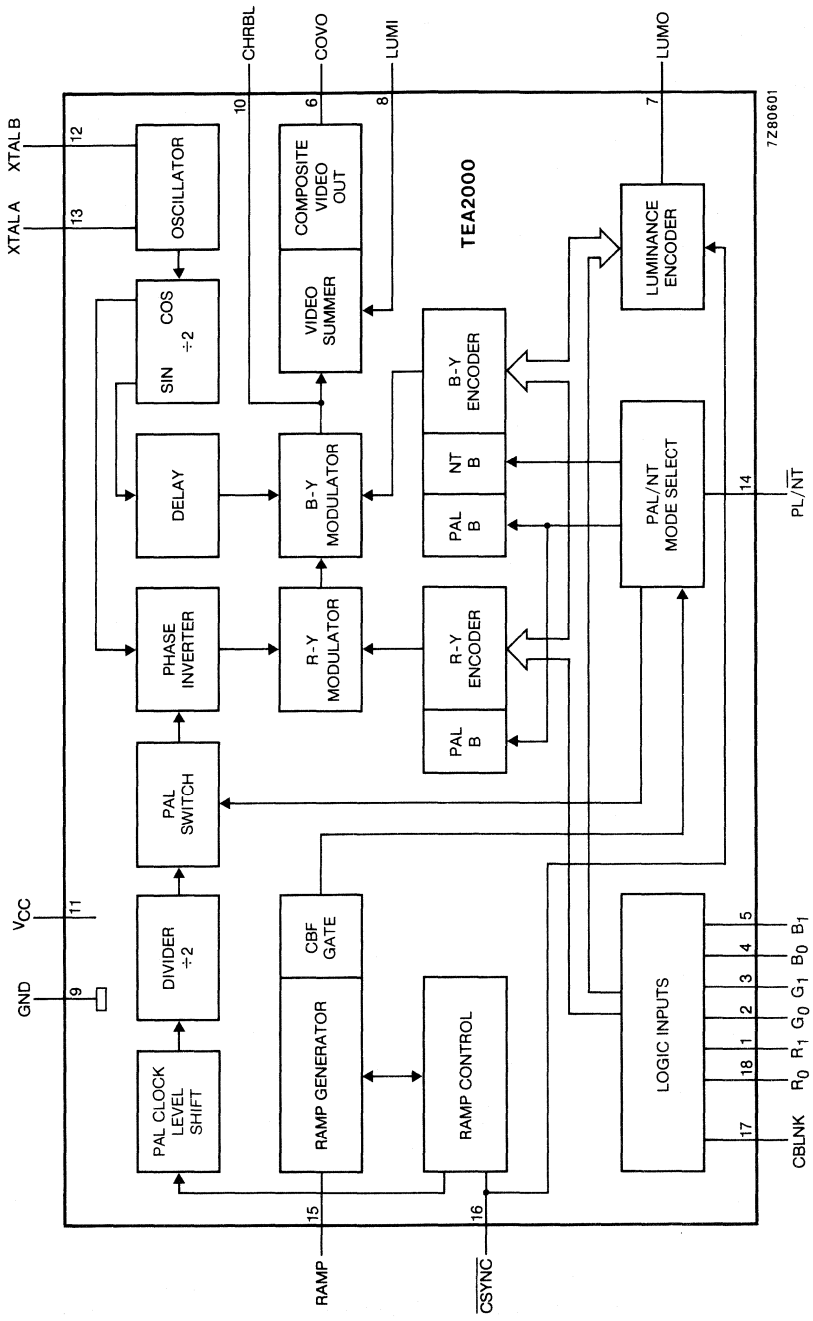


Fig. 1 Block diagram.

PINNING

1. Red 1 binary input
2. Green 0 binary input
3. Green 1 binary input
4. Blue 0 binary input
5. Blue 1 binary input
6. Composite video output
7. Luminance output to delay line
8. Luminance input from delay line
9. Ground 0 volt
10. Chrominance band limiting
11. Supply voltage
12. } Oscillator inputs { 7,16 MHz crystal for NTSC
13. } { 8,86 MHz crystal for PAL
14. PAL/NTSC switch
15. Ramp
16. Composite sync input ($\overline{\text{CSYNC}}$)
17. Composite blanking input (CBLNK)
18. Red 0 binary input

FUNCTIONAL DESCRIPTION

The TEA2000 PAL/NTSC colour encoder and video summer integrated circuit has an internal oscillator from which the (R-Y) and (B-Y) waveforms are generated. The TEA2000 accepts timing signals (composite sync, composite blanking) and a 6 bit binary coded input giving colour information. The inputs are organized as 2 bits per primary colour and gamma correction is applied to the resultant luminance and chrominance levels. Each of the equally spaced intensity levels (for each primary colour) is combined with those of the other primary colours. This produces 64 output colours comprising a wide range of saturated and desaturated colours, black, white and two levels of grey. The resultant output is a composite video signal compatible with the PAL and NTSC/M standards.

PIN DESCRIPTION

R0, R1, G0, G1, B0, B1, pins 18, 1,2,3,4 and 5.
These are the red, green and blue logic inputs. 2 bits per primary colour. These inputs are TTL compatible.

$\overline{\text{CSYNC}}$, pin 16.

Composite sync input requiring a negative logic signal, TTL compatible. For PAL operation the field sync must include line sync information.

XTALA, XTALB, pins 12 and 13.

Oscillator inputs. A crystal in series with a trimmer capacitor is connected between pins 12 and 13. The output of the oscillator is divided to provide the four subcarrier phases required in the encoder. The crystal frequencies are:

PAL mode 8,867238 MHz
NTSC mode 7,15909 MHz

LUMO, LUMI, pins 7 and 8.

Luminance output and input. The combined luminance and sync signal appearing at pin 7 must be d.c. coupled to pin 8 via an appropriate luminance delay line or resistor network. Resistors must have a tolerence of ± 5%, or better, as they affect the d.c. level at COVO, pin 6.

CHRBL, pin 10.

Chrominance filtering can be accomplished by connecting a chrominance frequency tuned filter (4,43 MHz or 3,57 MHZ), via a blocking capacitor to pin 10. This point is the chrominance summing junction and has a nominal internal impedance of 1,5 kΩ. If a filter is used at this point then the delay caused to the chrominance signal should be compensated by an appropriate luminance delay line.

COVO, pin 6.

Composite video output is internally buffered giving a nominal output voltage swing of 2 V sync-white and a nominal sync 5 V level.

PL/$\overline{\text{NT}}$, pin 14.

PAL/NTSC, select input selects PAL mode when HIGH and NTSC mode when LOW. This input is TTL compatable. An internal pull-up resistor selects PAL if the pin is not connected.

RAMP, pin 15.

Ramp timing component connection. A capacitor and resistor connected to pin 15 provide timing information for the colour burst and for PAL phase switching. Alternative components may be used to optimise for NTSC operation.

V_{CC}, pin 11.

12 volt supply.

GND, pin 9.

Ground connection, zero volts.

CBLNK, pin 17.

Blanking input when high, switches off colour inputs. CBLNK must be high during sync and colour burst unless colour inputs are all low at this time. This input is TTL compatible.

RATINGS

Limiting values in accordance with Absolute Maximum System (IEC 134)

Supply voltage V_{11-9}	max.	13,2	V
Voltages, pin 1,2,3,4,5,14,16,17,18	max.	V_{11-9}	V
Storage temperature		−20 to +125	°C
→ Operating ambient temperature		0 to + 70	°C

CHARACTERISTICS

V_{11-9} = 12 V; T_{amb} = 25 °C; measured in Fig. 3 unless otherwise specified

parameter	symbol	min.	typ.	max.	unit
Supply					
Supply voltage	V_{11-9}	10,8	12	13,2	V
Supply current V_{11-9} = 12 V	I_{11}	–	55	–	mA
Oscillator stability, pins 12 and 13					
Crystal type 4322 143 04051					
V_P = 10,8 to 12 V		–	+50	–	Hz
V_P = 12 to 13,2 V		–	–50	–	Hz
Digital inputs					
$\overline{CSYNC}$, CBLNK, PL/$\overline{NT}$ pins 16,17,14 R0,R1,G0,G1,B0,B1 pins 18,1,2,3,4,5					
V_{IN} (LOW)	V_{IL}	–0,5	–	0,8	V
V_{IN} (HIGH)	V_{IH}	2	–	V_{11-9}	V
Input capacitance	C_i	–	–	10	pF
Input rise and fall times	t_r,t_f	–	–	200	ns
$\overline{CSYNC}$, CBLNK, R0,R1,G0,G1,B0,B1 pins 16,17,18,1,2,3,4,5					
Input current d.c. for V_{IN} = 0 V	I_{IL}	–	–	–100	μA
Input current d.c. for V_{IN} = 2 V	I_{IH}	–	–	20	μA
PL/$\overline{NT}$, pin 14					
Input current d.c. for V_{IN} = 0 V	I_{IL}	–	–	–500	μA
Input current d.c. for V_{IN} = 2 V	I_{IH}	–	–	–200	μA
Composite video output, pin 6					
Output amplitude (sync tip-white)	$V_{6-9\ (p-p)}$	–	2	–	V
Sync tip level	V_{6-9}	–	5	–	V
Output load resistor	R_{6-9}	0,47	1	–	kΩ
Variation of output amplitude T_{amb} = 0 to +70 °C	$V_{(p-p)}$	–	–	tbf	%
Over supply range V_{11-9} = 10,8 to 13,2 V	ΔV	–	–	tbf	%
Output impedance (with 1 kΩ load)	R_L	–	15	–	Ω
Residual chrominance on white	ΔV_{rms}	–	30	–	mV
Tolerance on luminance amplitude	ΔV	–	10	–	%
Tolerance on chrominance amplitude	ΔV	–	10	–	%
Tolerance on chrominance phase	ΔΩ	–	tbf	–	%
Chrominance band limiting, pin 10					
Internal resistance	R_{10-11}	–	1,5	–	kΩ
Luminance delay, pins 7 and 8					
Nominal series resistor (±5%)	R_S	–	1,2	–	kΩ
Nominal load resistor at luminance input (±5%)	R_L	–	1	–	kΩ
Ramp timing, pin 15 (see Fig. 4)					
With external RC circuit R = 36 kΩ; C = 330 pF (note 1)					
Start of burst from line sync	t_b	–	5,7	–	μs
Burst width	t_w	–	2,5	–	μs
Threshold for seperation of equalizing pulses and sync pulses	t	36	44	56	μs

Note: 1. A figure of 5 pF is assumed for external capacitance. This figure includes temperature dependence of the components.

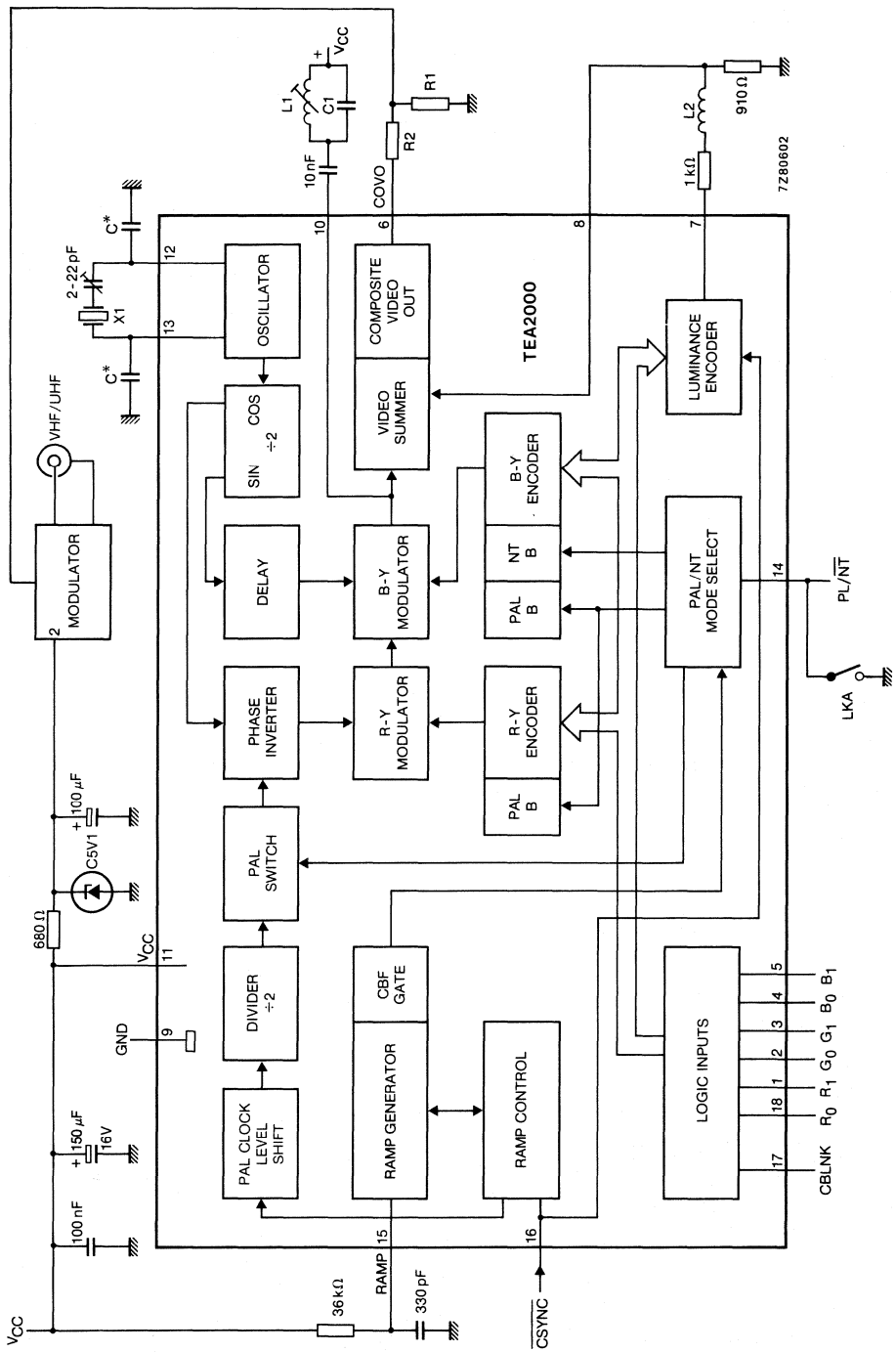


Fig. 2 Internal circuit details and typical external connections.

X1 (PAL) = 8,867238 MHz
X1 (NTSC) = 7,159100 MHz
C* = 5,6 pF only for mask version 1

COMPONENT	PAL	NTSC
L1	15 μH	18 μH
C1	82 pF	100 pF
L2	DL270	DL330
R1	430 Ω	510 Ω
R2	510 Ω	750 Ω
M1	UM1233	UM1622
LKA	o/c	made

Component list for Fig. 2.

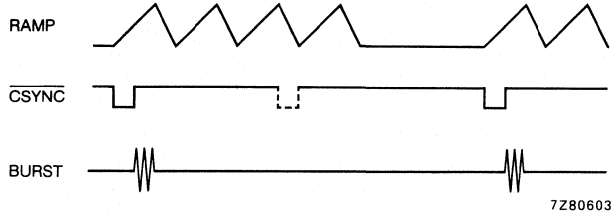


Fig. 3 Ramp timing.

SPEECH/SOUND SYNTHESIZERS

VOICE SYNTHESIZER

GENERAL DESCRIPTION

The MEA8000 is a 24-pin N-MOS integrated circuit for generating good quality speech from digital code with a programmable bit rate. The circuit is primarily intended for applications in microprocessor controlled systems, where the speech code is stored separately.

Features

- Interfaces easily with most popular microprocessors and microcomputer
- 8-bit wide data bus
- 32-bit wide data buffer holding speech frame codes
- Digital filter of 8th order with 3 programmable formant frequencies, one fixed formant frequency, and 4 programmable formant bandwidths
- Programmable amplitudes
- Programmable duration of each frame; 8, 16, 32 or 64 ms
- Synthesis occupies less than 1% of control processor time
- Capable of sophisticated unvoiced sound generation
- Crystal controlled oscillator or external (TTL) clock
- Minimal external audio filter requirement
- Single + 5 V power supply

QUICK REFERENCE DATA

parameter	condition	symbol	min.	typ.	max.	unit
Supply voltage	pin 13	V_{DD}	4,5	5,0	5,5	V
Supply current	no audio load	I_{DD}	–	30	50	mA
Inputs						
Input voltage	HIGH	V_{IH}	2,0	–	V_{DD}	V
Input voltage	LOW	V_{IL}	−0,5	–	0,8	V
Input capacitance		C_I	–	–	7	pF
Outputs						
Output voltage	$-I_{OH}$ = 100 μA	V_{OH}	2,4	–	–	V
Output voltage	I_{OL} = 1,6 mA	V_{OL}	–	–	0,4	V
Capacitance		C_L	–	–	30	pF
Operating ambient temperature range		T_{amb}	0	–	+ 70	°C

PACKAGE OUTLINE

24-lead DIL; plastic (SOT-101A).

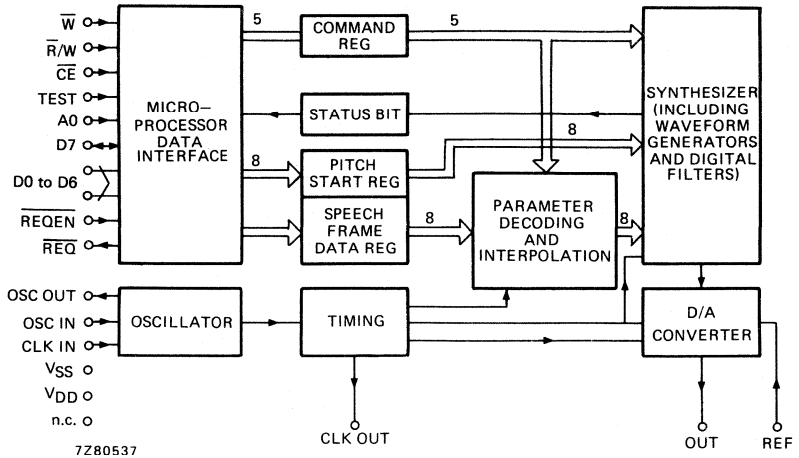


Fig. 1 Block diagram.

PINNING

1	V_{SS}	ground
2	$\overline{REQ}$	data request
3	D7	data bus
4	D6	
5	D5	
6	D4	
7	D3	
8	D2	
9	D1	
10	D0	
11	A0	data/control input
12	$\overline{CE}$	chip enable
13	V_{DD}	supply voltage
14	$\overline{REQEN}$	request enable input
15	N.C.	not connected
16	OSC IN	internal oscillator
17	OSC OUT	
18	CLK IN	clock input
19	REF	reference current
20	OUT	speech output
21	CLK OUT	internal clock output
22	$\overline{R}/W$	read/write
23	$\overline{W}$	write
24	TEST	test use only

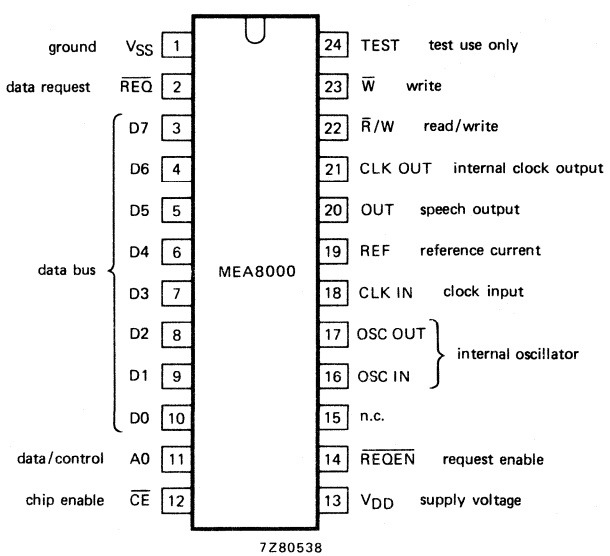


Fig. 2 Pinning diagram.

FUNCTIONAL DESCRIPTION (pin number)

Control

D0 to D7	(10 to 3)	Data bus to which command or speech can be written.
D7	(3)	Data port via which the status can be read.
$\overline{CE}$	(12)	Chip enable (chip select).
$\overline{W}$	(23)	Write.
$\overline{R}$/W	(22)	Read/Write The control signals $\overline{W}$ and $\overline{R}$/W allow connections to most microcomputers or microprocessors (see timing diagrams).
A0	(11)	Data/control input: discriminates between speech code input buffer (A0 = '0') and command register (A0 = '1') during a 'write' operation.
$\overline{REQ}$	(2)	Data request (open drain output); output signal which follows inverse of the status REQ bit, but only if enabled by either the ROE bit in the command register or the external $\overline{REQEN}$ pin.
$\overline{REQEN}$	(14)	Request enable input; $\overline{REQEN}$ = '0' enables the status REQ output, independent of the status of the command register.

Timing

OSC IN	(16)	Connections for internal clock oscillator; nominal crystal frequency 4 MHz.
OSC OUT	(17)	
CLK IN	(18)	Clock input for external clock, TTL compatible, 4 MHz.
CLK OUT	(21)	A buffered output for the internal clock cycle (which is equal to CLK divided by 3). May be used as a clock, for a microprocessor, for example.

Output

REF	(19)	Input pin for biasing the audio output level. This reference current can be derived from a resistor to the positive supply.
OUT	(20)	Speech output; this output is a 64 kHz pulse, modulated in both width and amplitude. It is configured as a current sink with a saturating voltage of about 3 V.

Supply

V_{DD}	(13)	Single supply voltage, nominally 5 V, but battery operation is possible.
V_{SS}	(1)	Ground.
TEST	(24)	Used for testing purposes. Changes other pin functions. Must be tied to ground for user operation.
NC	(15)	It is recommended to ground this pin.

HANDLING

Inputs and outputs are protected against electrostatic charge in normal handling. However, to be totally safe, it is desirable to take normal precautions appropriate to handling MOS devices (see 'Handling MOS Devices').

RATINGS

Limiting values in accordance with the Absolute Maximum System (IEC 134)

parameter	conditions	symbol	min.	max.	unit
Supply voltage range		V_{DD}	−0,5	+ 7	V
Voltage with respect to V_{SS}	on any pin	V_I	−0,5	+ 7	V
Output voltage	pins 2 and 20	V_{REQ}, V_{OUT}		15	V
Storage temperature range		T_{stg}	−20	+ 125	°C
Operating ambient temperature range		T_{amb}	0	+ 70	°C

CHARACTERISTICS

T_{amb} = 25 °C; V_{DD} = 5 V, unless otherwise specified; all voltages referenced to V_{SS}

parameter	conditions	symbol	min.	typ.	max.	unit
Symbol						
Supply voltage	note 1	V_{DD}	4,5	5,0	5,5	V
Supply current	no audio load	I_{DD}	–	30	50	mA
Inputs						
D0 to D7, A0, $\overline{CE}$, $\overline{W}$, $\overline{R}$/W, $\overline{REQEN}$, CLK IN						
Input voltage HIGH		V_{IH}	2,0	–	V_{DD}	V
Input voltage LOW		V_{IL}	–0,5	–	0,8	V
Input leakage current	note 2	I_{IR}	–	–	10	μA
Input capacitance		C_I	–	–	7	pF
Outputs						
D7 (I/O), CLK OUT						
Output voltage HIGH	$-I_{OH}$ = 100 μA	V_{OH}	2,4	–	–	V
Output voltage LOW	I_{OL} = 1,6 mA	V_{OL}	–	–	0,4	V
Output load capacitance		C_L	–	–	50	pF
$\overline{REQ}$						
Output voltage HIGH	open drain	V_{OH}	–	–	13,2	V
Output voltage LOW	I_{OL} = 1,6 mA	V_{OL}	–	–	0,4	V
Output load capacitance		C_L	–	–	50	pF
Audio output						
Reference current	pin 19; note 8	I_{REF}	–	–	0,3	mA
Output current	pin 20; peak value					
	I_{REF} = 0 mA	I_{OUT}	–	100	–	μA
	I_{REF} = 0,1 mA	I_{OUT}	–	1,7	–	mA
	I_{REF} = 0,3 mA	I_{OUT}	–	5	–	mA
Output voltage	pin 20; for linear operation; note 3;					
	I_{REF} = 0,1 mA	V_{OUT}	2,5	–	13,2	V
Oscillator						
Crystal frequency	internal	f_{XTAL}	–		4,00	MHz
Clock frequency	external	f_{CLK}	–		4,00	MHz

TIMING CHARACTERISTICS (note 4) (Figs 6 and 7)

parameter	condition	symbol	min.	typ.	max.	unit
Write enable		t_{WR}	200	–	–	ns
Address set-up		t_{AS}	30	–	–	ns
Address hold		t_{AH}	30	–	–	ns
Data set-up for write		t_{DS}	150	–	–	ns
Data hold for write		t_{DH}	30	–	–	ns
Request hold	note 5	t_{RH}	–	–	350	ns
Request next	note 6 clock frequency = 3,84 MHz	t_{RN}	–	–	3	μs
Read enable		t_{RD}	200	–	–	ns
Data delay for read	note 7	t_{DD}	–	–	150	ns
Data floating for read	note 7	t_{DF}	–	–	150	ns
Request valid before write		t_{RV}	0	–	–	ns
Request output enable response		t_{ROE}	–	–	750	ns
Control set-up		t_{CS}	20	–	–	ns
Control hold		t_{CH}	20	–	–	ns

Notes

1. The circuit will continue to operate from a supply of up to 6,5 V, but without necessarily meeting the specification.
2. This is also valid for V_{DD} = 0 V.
3. This permits the connection of the output load to a supply higher than that supplying the synthesizer.
4. Timing reference level is 1,5 V.
5. An external pull-up resistor is required, as this is an open drain output.
   The time (t_{RH}) to reach 2,0 V is specified at a load to 5 V of 3,3 kΩ and 50 pF.
6. Between two data write operations of one speech frame.
7. Levels greater than 2,0 V for a '1' or less than 0,8 V for a '0' are reached with a load of one TTL input and 50 pF.
8. Typical voltage level at the REF pin is 2,5 V.

OPERATION PRINCIPLE

The MEA8000 has been designed for a vocal tract modelling technique of voice synthesis. This method gives the lowest possible bit rate for speech quality which is acceptable for most industrial applications.

Figure 3 shows a simplified electronic model of the human vocal tract as a formant synthesizer. A combination of a periodic signal, representing the pitch of the original speech, and an aperiodic signal, representing the unvoiced sound in the speech. Both these signals are fed to a variable filter comprising four resonantors (via an amplifier which controls the amplitude of the synthesized sound). The resonators model the sound in accordance with the formats in the original speech. Each resonator is controlled by two parameters, one for the resonant frequency and one for the bandwidth.
The information required to control the synthesizer is:

- pitch
- amplitude
- voice/unvoiced source selector

} excitation source (vocal cords)

- filter control — spectrum shaping (vocal tract)

A good replica of the original speech is obtained by periodic updating of this control information.

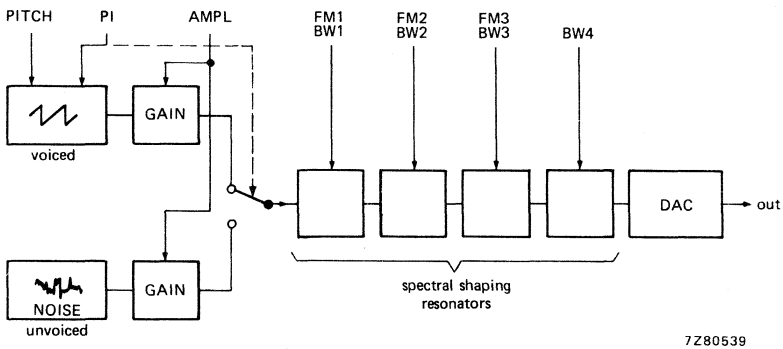


Fig. 3 Electronic model of human vocal tract.

OPERATION

Speech is generated by suitable filtering of a relatively low frequency sawtooth waveform for voiced sounds, or of random noise for unvoiced sounds. New parameters for both the digital waveform generator and the digital filter are supplied to the synthesizer in coded groups of 4 bytes via the data bus. The code group also contains the duration of the next speech frame to be produced (8, 16, 32 or 64 ms).

The output sample rate is 64 kHz or 8 times the internal sample rate with linear interpolation in between. This greatly reduces the need for an external analogue output filter.

Modes of operation

1. STOP mode: characterised by a silent output and the status $\overline{\text{REQ}}$ bit set to '1'. This mode is entered from power up or by STOP command. The mode is entered automatically if at the end of an active speech frame the next four parameter bytes are not yet received while the CONT bit in the command register is a '0'. In the latter case the final speech frame will be repeated once but with a decaying amplitude and the same pitch.
2. ACTIVE mode: a speech sample is being produced.
3. CONTINUOUS mode: entered if an active speech frame is finished and new data is not supplied in time while the CONT bit in the command register is a '1'. The synthesizer will repeat the last speech frame indefinitely until all four new data bytes are received, or a STOP command, or a reset of the CONT bit.

Speech code input buffer

Speech code is written to the synthesizer when $\overline{CE}$ and $\overline{W}$ are both '0', while $\overline{R}/W$ = '1' and A0 = '0'. Also the status $\overline{REQ}$ bit must read a '1', otherwise the synthesizer is still busy and will not react to a data write operation.
Starting from the STOP mode, the first data will be interpreted as a starting value for the PITCH. Thereafter every four successive data bytes are treated as a group of speech code. The coded speech frame format is shown in Fig. 4.

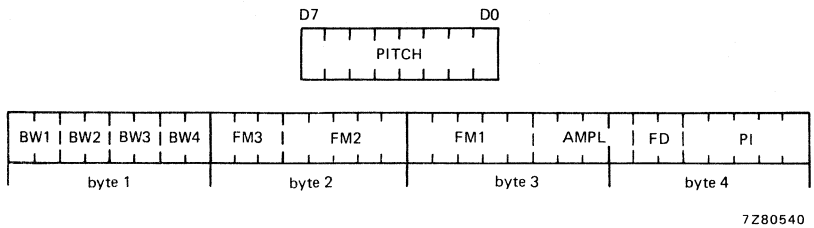


Fig. 4 Format of coded speech frame.

code	bits	parameter
PITCH	8	initial value for pitch
FD	2	speech frame duration
PI	5	pitch increment (rate of change) or noise selection
AMPL	4	amplitude
FM1	5	frequency of 1st formant
FM2	5	frequency of 2nd formant
FM3	3	frequency of 3rd formant
FM4	0	frequency of 4th formant (fixed)
BW1	2	bandwidth of 1st formant
BW2	2	bandwidth of 2nd formant
BW3	2	bandwidth of 3rd formant
BW4	2	bandwidth of 4th formant

During each data write operation, the status REQ bit will be cleared to '0'.
It appears within a few microseconds, requesting the next byte of the group.
The request for the first byte of the next group always appears shortly after the beginning of the current speech frame, and all four bytes must be provided before it finishes. This leaves the control circuit (i.e. microprocessor) enough time to use polling, instead of interrupts, as the minimum time of a speech is 8 ms.

When in the STOP mode the synthesizer will commence producing sound after receipt of 1 + 4 bytes.

Status bit

The status bit is accessed at $\overline{CE}$ = $\overline{R}/W$ = '0'.
The status of $\overline{W}$ and A0 are arbitrary.
Pin D7 reveals the request for a (next) speech code byte: '0' = busy, '1' = request for data.

Command register

A command is written to the synthesizer at $\overline{CE}$ = $\overline{W}$ = '0' while A0 = $\overline{R}$/W = '1'.

D7	D6	D5	D4	D3	D2	D1	D0
			STOP	CONT enable	CONT	ROE enable	ROF
NOT USED			'0' = INVALID '1' = STOP	00 = INVALID 01 = INVALID 10 = SLOW STOP 11 = CONTINUE		00 = INVALID 01 = INVALID 10 = DISABLE $\overline{REQ}$ OUTPUT 11 = ENABLE $\overline{REQ}$ OUTPUT	

STOP — Stop mode. This results in an immediate reset of the synthesizer to the STOP mode. The ROE and CONT are not affected by this command.

CONT — Continuous mode. This bit can be set or cleared only if the corresponding CONT enable bit is programmed as a '1'. In the continuous mode the synthesizer will not revert to the STOP mode if all four parameters are not received before the end of the current speech frame, but repeat it indefinitely.

If CONT = '1' the last frame will be repeated once with decaying amplitude and the same pitch before the stop mode is entered.

ROE — Request Output Enable. This can be set or cleared only if the corresponding ROE enable bit is a '1'. ROE determines whether the request in the status bit appears on the $\overline{REQ}$ pin.

Note: the same can be achieved by connecting the $\overline{REQEN}$ pin (request enable) to a '0'.

After power on, the command register bits CONT and ROE will both be zero. Thus power on equals the command 00011010 = 1 A (hexadecimal).

Control signals

With the three control signals $\overline{CE}$, $\overline{W}$ and $\overline{R}$/W the synthesizer is made compatible with most microprocessors and microcomputers.

$\overline{CR}$	$\overline{W}$	$\overline{R}$/W	A0		Operation
0	0	1	0		WRITE DATA
0	0	1	1		WRITE COMMAND
0	X	0	X		READ STATUS
0	1	1	X		3-STATE DATA BUS
1	X	X	X		3-STATE DATA BUS

Power supply

During (slow) power up or power down the circuit will not produce any spurious sound. As soon as the supply is high enough for reliable operation, the circuit will be in the STOP mode with ROE = CONT = '0'.

Timing diagrams

The control signals $\overline{CE}$, $\overline{R}/W$ and $\overline{W}$ have been specified to enable easy interface to most microprocessors and microcomputers. For instance, with connection to an MAB8048 microcomputer the $\overline{R}/W$ and $\overline{W}$ inputs can be used as the $\overline{RD}$ and $\overline{WR}$ strobe inputs.

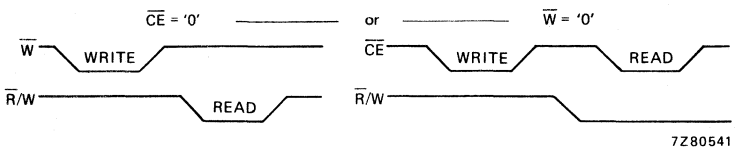


Fig. 5 Typical waveforms of the control signals.

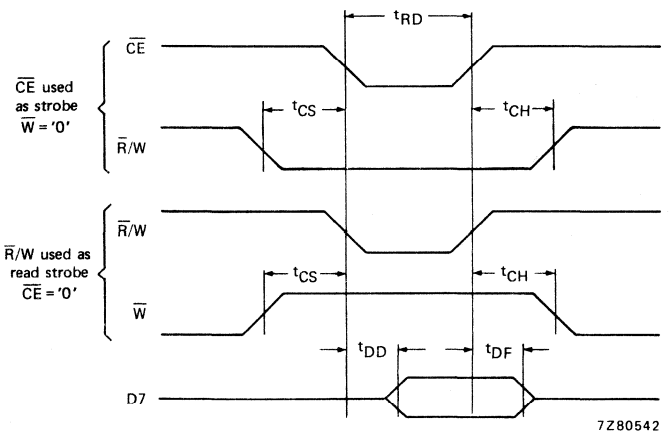


Fig. 6 Read timing.

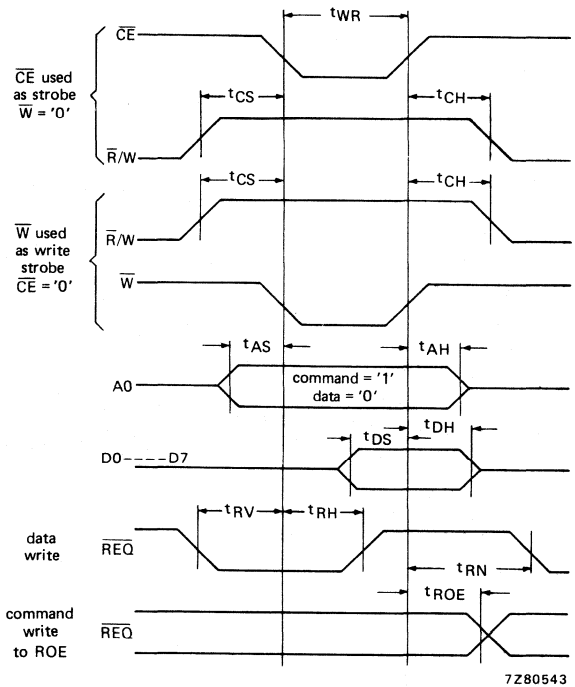
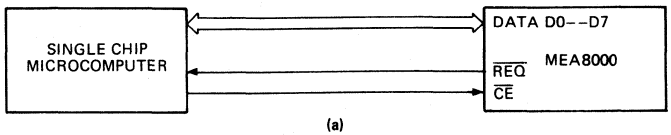
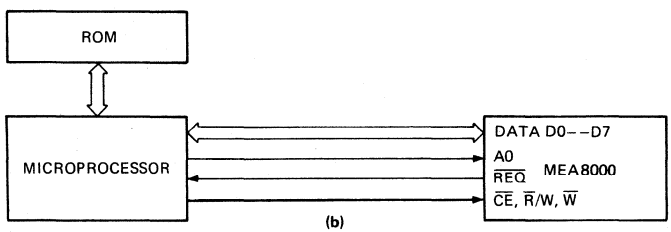


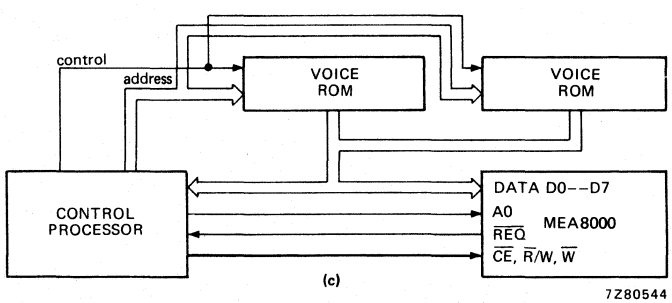
Fig. 7 Write timing.



(a) Minimum system of single chip microcomputer with voice ROM on board.



(b) MEA8000 as a microprocessor peripheral.



(c) Applications using separate voice ROMs.

Fig. 8 Typical applications.

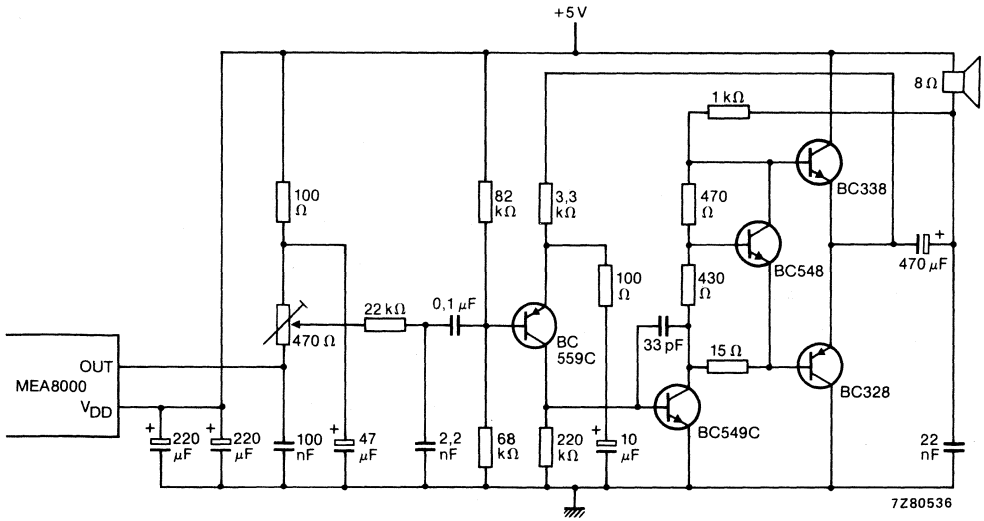


Fig. 9 Typical output applications.

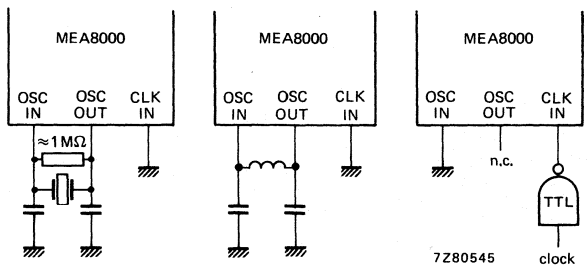


Fig. 10 Oscillator/clock configurations.

LOW COST SPEECH DEMONSTRATION BOARD

GENERAL DESCRIPTION

The low cost speech demonstration board is designed to add voice output to existing card based electronic equipment with the minimum of additional effort and components. The majority of components used are of the CMOS type with low power consumption making the board suitable for battery operation.

Applications include speech evaluation and speech demonstration.

FEATURES

- PCF8200 speech synthesizer
  - Male and female speech of very high quality
  - CMOS technology
  - Extended operating temperature range
  - Programmable speaking speed
- Low current consumption
  - All major components use CMOS technology (PCF8200, 80C39 and 27C64)
- Very large vocabulary up to 12 minutes
  - 4 EPROM sockets
  - EPROM selection for 27C16 to 27C256
  - Low data rates for synthesizer (average 1500 bits per second)
- Easy interfacing
  - 8-bit parallel data bus/key switch input
  - Volume control, speaker connection
  - Control signals (e.g. RESET, BUSY etc etc)
- Simple operating modes
  - ROM selection
  - Word sequence within a ROM
  - Repeat last utterence
  - Control software is readily customizeable
  - To implement parameter download from external source
- Single Eurocard size PC board
- Single + 5 V supply
- Low cost

APPLICATIONS

- OEM design-in
- May be simply used with many card systems for speech evaluation
- Speech demonstration
  - Particularly simple when used with the OM8201 (Speech Demonstration Box)

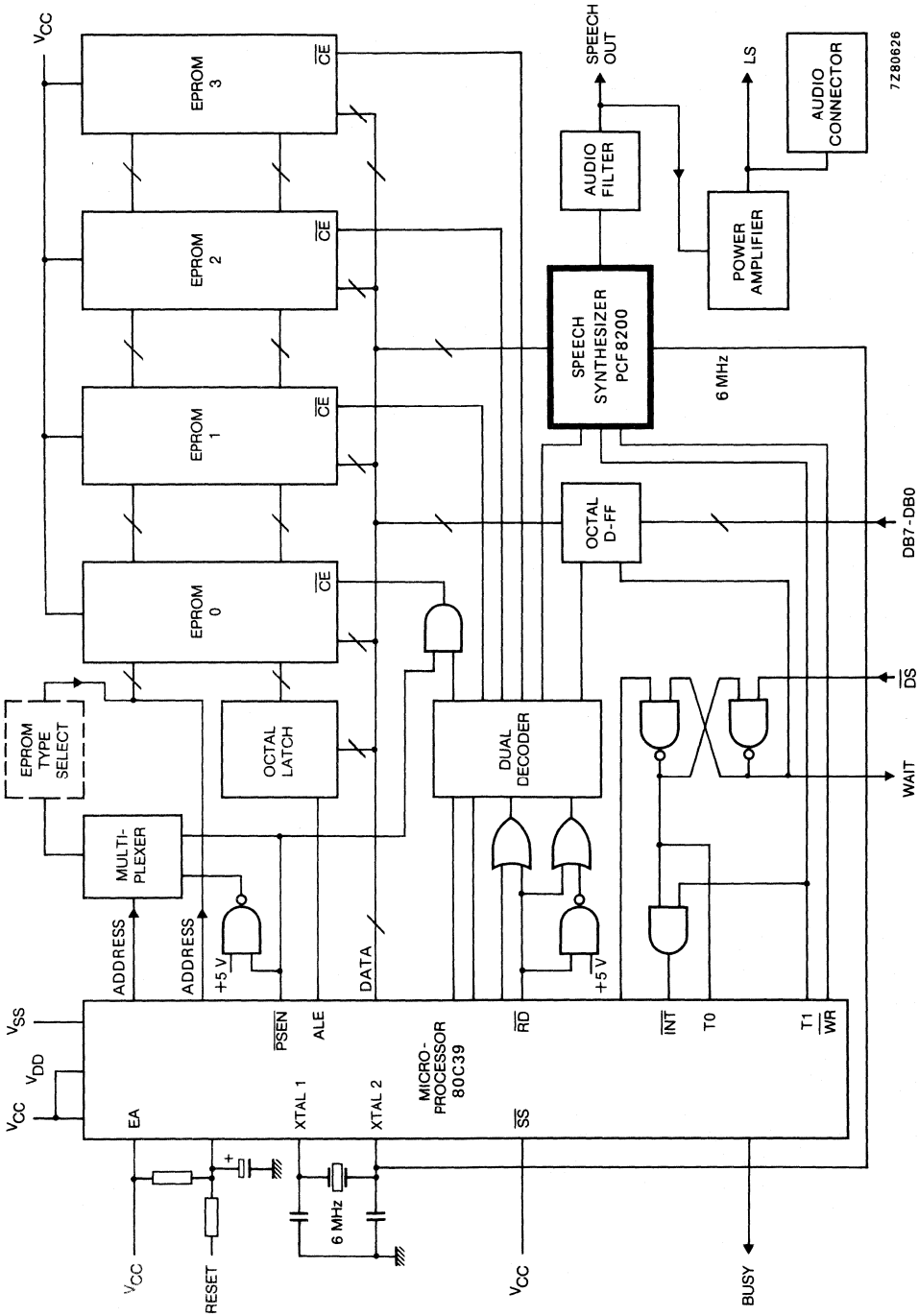


Fig. 1 Block diagram.

OPERATION

HARDWARE DESCRIPTION

The main controlling microprocessor is an 80C39 running at 6 MHz. This device supplies all of the main controlling signals for the board operation and the interfacing to any external system.
Four sockets are provided for EPROMS which contain speech coding. These may be 27C16 types, through to 27C256 types; the sockets will be a low insertion force type to allow for easy customizing.
The board will be supplied with one socket occupied by a 27C64 which will contain the control program and some speech examples. All four EPROM sockets must contain the same EPROM type.

The speech synthesizer PCF8200 converts the coding into a speech output. This synthesizer has been designed to simulate the human vocal tract using five formants for male and four formants for female speech. Periodic updating of the parameters for these formants can produce very high quality speech.

The output of the synthesizer can be fed into an audio amplifier, TDA7050, via a resistor-capacitor filter network which provides a frequency cut-off above 5 kHz of about 25 dB. The configuration of the audio amplifier used on this board gives an output of 140 mW peak power into a 25 Ω speaker from a 5 V supply.

Connections are made to the board via a standard DIN/IEC connector. This allows access to the 8-bit parallel data bus so that speech coding from an external source may be used, if implemented, and allows the selection of speech phrases by an external system, such as a microcomputer or even a bank of switches. The same connector also permits the addition of a volume control, loudspeaker, a high impedance audio output, and power supply. The control signals RESET, BUSY, WAIT and DS are also taken to the outside of the board. There is also a loudspeaker plug on the board.

All components are contained on a standard single Eurocard, and therefore suitable for rack mounted equipment.

SOFTWARE DESCRIPTION

All the software required to operate the board is contained in the only EPROM supplied. The software is written in modular from so that it is possible for a customer to alter or add to any particular function which suits his applications. An industrial standard microprocessor was chosen so that readily available development systems could be used to facilitate this modification.

There are four main modes of operation:

- ROM Selection
- Word Sequence
- Repeat Word
- Speaking Speed Selection

These modes are all controlled by software.

ROM Selection mode permits access to an individual EPROM and pronounces the first utterence from that EPROM.

Word Sequence gives the next word (activated by repeated access to the same EPROM) and if continually exercised will keep looping on the words in that EPROM.

The Repeat Word command allows indefinate repetition of the last utterance pronounced.

The Speaking Speed Selection allows the utterence to be pronounced at a different speed.

The software also controls the address sequencing within the utterance and ensures that the required data is supplied to the synthesizer.

There are also some examples of words/utterences encoded in the remainder of the supplied EPROM. These words are intended for demonstration purposes and will show the features of the synthesizer when selected. The main features being illustrated are:

- Male speech in several languages
- Female speech in several languages
- Programmable speaking speed

ORDERING INFORMATION

Product name: Low Cost Speech Demonstration Board

Type number: OM8200

Ordering code: 9337 541 30000

Orders should be placed with your local Philips/Signetics agency.

SPEECH DEMONSTRATION BOX

GENERAL DESCRIPTION

Speech demonstration box OM8201 is designed to be used in conjunction with the low cost speech demonstration board OM8200. The box contains all the necessary components to drive the board. The combination of these two components make an extremely attractive demonstration unit.

FEATURES

- Low cost
- Can use unmodified OM8200 board which allows access to all features of the OM8200
- Single + 9 V supply
  - Low power consumption therefore permits battery operation
  - External power supplies may also be used
  - Voltage is regulated and dropped to a standard + 5 V for the OM8200 board
- Simple mechanical construction
  - Allows easy access to the OM8200 for changing EPROMS
- Contains all peripherals needed to drive the OM8200

HARDWARE DESCRIPTION

The box contains a set of eight keypad switches which are connected to the data bus. Four switches can select which EPROM your speech data is derived from. Repeated pressing of an EPROM switch increments the expression number which will be uttered. To repeat the last expression, a separate switch must be activated.

It is possible in the PCF8200 to change the rate of speaking to 73%, 123% or 145% of the normal speed. A switch has been included on the box which will sequence through the speed options making the same utterance every time.

One of the two remaining switches is the master reset for the program and the other is for future enhancements of the box.

Included in the box are, the volume control for the amplifier, the loudspeaker, and a high impedance audio output.

The final piece of electronics is the power supply. This can be supplied from a + 9 V internal battery or from a + 9 V external supply. The + 9 V is regulated to a + 5 V supply which is then fed to other parts of the box and to the OM8200.

The box is of simple construction and allows easy access to the OM8200 for changing of EPROMS.

SOFTWARE DESCRIPTION

There is no software in the OM8201. The software of the OM8200 may be used in an unmodified form without any problems. However, if changes have been made to the control program of the OM8200 then different functions for the switches of the box can be achieved.

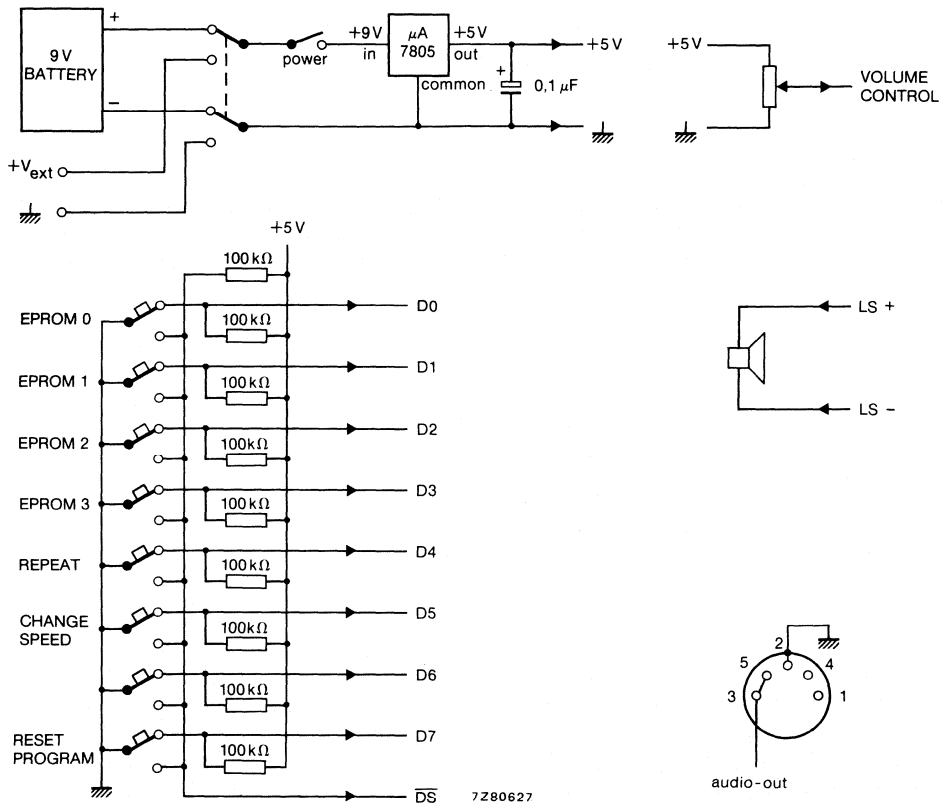


Fig. 1 Schematic diagram.

ORDERING INFORMATION

Product name: Speech Demonstration Box

Type number: OM8201

Ordering code: 9337 541 40000

N.B. OM8200 must be ordered as well if this box is to be used in demonstration mode. The order number for the OM8200 is 9337 541 30000.

Orders should be placed with your local Philips/Signetics agent.

SPEECH ANALYSIS/EDITING SYSTEM

GENERAL DESCRIPTION

The OM8210 is a speech analysing/editing system, and comprises of a speech adapter box and associated software. The system uses either the HP9816S or IBM-PC personal computer.
The OM8210 and the computer function together to produce speech coding for the PCF8200.
The system has many commands available, mostly single key operations, which gives it flexability.

FEATURES

- Input sampling of analogue speech signals
- Speech analysis
- Graphic parameter representation
- Parameter editing screen
- Conversion of parameters to PCF8200 synthesizer
- EPROM programming
- Parameter storage on floppy disc
- Speech output via PCF8200 voice synthesizer

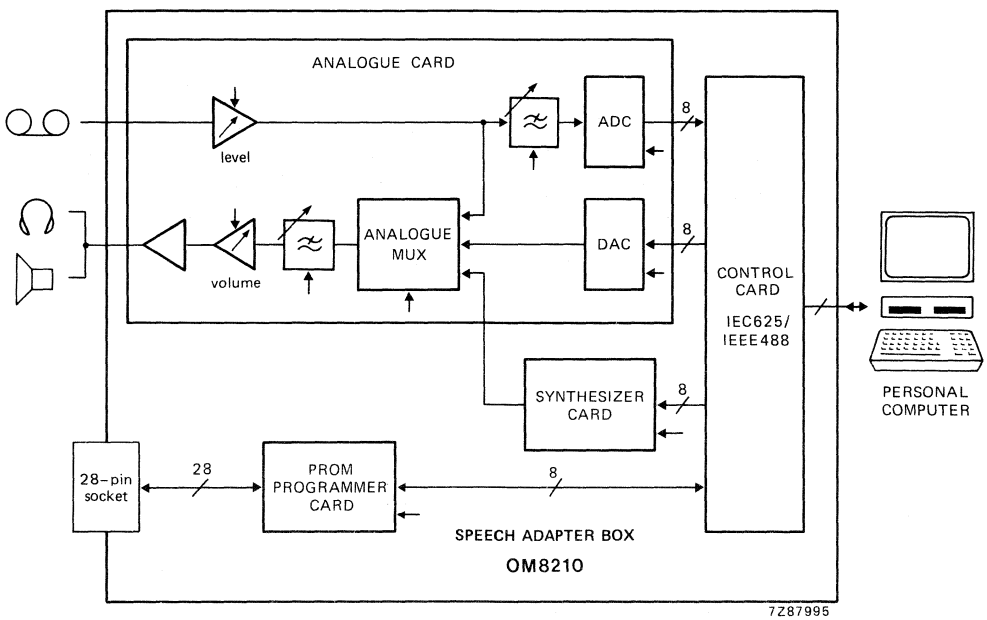


Fig. 1 Block diagram.

HARDWARE DESCRIPTION

The hardware for the OM8210 is contained in an attractive box with access to all the interconnections (IEC 625, interface loudspeaker, headphones, tape input, and EPROM socket), from the front panel. There are four single Eurocards and a power supply forming the speech adapter box.
These cards are:

- Analogue Card
- Synthesizer Card
- EPROM Card
- Control Card

Analogue Card

On this card, the level of the recorded audio input signal is adjusted by an electronic potentiometer. Before the audio is sampled, frequencies higher than half the sampling frequency are removed by a switched capacitor filter of the type normally used for codecs. A 12-bit analogue-to-digital converter (ADC) produces the digital samples that are sent to the control card. An 8-bit digital-to-analogue converter (DAC) on the analogue card allows the sampled speech to be output. The audio input signal, the sampled speech and the synthesized speech are selected by an analogue multiplexer, filtered, and adjusted for volume before reproduction by a loudspeaker.

The use of integrated electronic potentiometers and codec filters substantially reduces the number of components required while maintaining high performance.

Synthesizer Card

This card accommodates the PCF8200 voice synthesizer and a small amount of peripheral components and a socket for the MEA8000 voice synthesizer.

EPROM Programmer Card

This card allows four different types of EPROM (2716, 2732, 2732A and 2764) to be programmed under software control. All the hardware to generate the programming voltages and the programming waveforms are on this card.

Control Card

This card performs three functions:

- IEC 625/IEEE 488 interface
- Control sequencer
- Clock generator

The IEC/IEEE interface is a simple talker/listener implementation with a HEF4738 circuit.

An FPLA control sequencer provides the handshake signals for IEC/IEEE interface and the chip enable signals for the rest of the system (the ADC, the DAC, the synthesizer and control circuits).

The filter sampling frequency is generated with a software programmable PLL frequency synthesizer. The speech sampling frequency is derived from the filter sampling frequency by frequency division. Hence, the filter frequency cut-off and the sample rate of the ADC and the DAC are automatically linked.

The hardware includes all the necessary cables, adapter plug, loudspeaker, headphone and power supply.

SOFTWARE DESCRIPTION

The software for this speech coding system has been developed and arranged for optimum user convenience. There are eight modes available.

Each mode and each command in the mode is selected by single key entries. Commands that can destroy data have to be confirmed before they are executed. More than 100 commands are available. The modes are:

Sample Mode	Samples and digitizes the recorded speech, the amplitude can be checked and speech segments selected. The sampled speech is stored in a memory and can be displayed or made audible.
Analysis Mode	Generates speech parameters from samples. The analysis selects the voiced/unvoiced sections, extracts the formants (5 for male and 4 for female), amplitude, and the pitch, and quantisizes the speech parameters.
Parameter Edit Mode	Speech parameters are displayed graphically on the VDU and can be edited to correct errors in the analysis, improve speech quality by altering contours, or amplitudes, concatenate sounds and optimize data rate by editing the frame duration.
Code Mode	Generates PCF8200 code and permits the arrangement of utterences in the optimum order of application. This mode also generates the address map at the head of the EPROM.
EPROM Mode	Used to program/read EPROMS with data for the code memory also possible is a new check, bit check and verification commands.
File Mode	Stores speech parameters or codes on disc, can also assemble code speech segment from an already existing library.
Media Mode	For diskette initialization and making back-up copies.
Option Mode	Allows the system configuration to be read or changed.

The software is supplied on two diskettes, one labelled 'BOOT' which wakes up the system and also contains the system library routines. The other diskette labelled 'SPEECH' contains the speech program, the disc initialization and the file handler programs. The 'BOOT' disc is not required during operation, giving a free disc drive with the system for a diskette to store speech parameter files.

DEVELOPMENT DATA

Computer System

The following equipment is required to make a complete Hewlett Packard based editing system:
- HP9816S-630 (optimum computer type) or HP9817
- HP9121D (dual floppy disc)
- Additional memory card for the HP9816S (512 K bytes total required)

The following equipment is required to make a complete IBM based editing system:
- IBM-PC or PC-XT or Philips P3100
- Additional memory (512 K recommended)
- Display graphics card (Hercules monochrome)
- IEEE488 card (Tecmar Rev. D.)

ORDERING INFORMATION

Product name:	Speech Analysis/Editing System
Type number:	OM8210
Ordering code:	9337 561 50112

The computer system should be purchased from your local agents.
The OM8210 should be ordered through your local Philips/Signetics agent.

VOICE SYNTHESIZER

GENERAL DESCRIPTION

The PCF8200 is a CMOS integrated circuit for generating good quality speech from digital code with a programmable bit rate. The circuit is primarily intended for applications in microprocessor controlled systems, where the speech code is stored separately.
Applications include automotive, telephony, personal computers, annunciators, aids for the handicapped, and general industrial devices.

Features

- Male and female speech with good quality
- Speech-band from 0 to 5 kHz
- Bit-rate between 455 bits/second and 4545 bits/second
- Programmable frame duration
- Programmable speaking speed
- CMOS technology
- Operating temperature range −40 to + 85 °C
- Single 5 V supply with low power consumption and power-down stand-by mode
- Interfaces easily with most popular microcomputers and microprocessors through 8 bit parallel bus or I^2C bus
- Software readable status word (parallel bus or I^2C bus)
- BUSY-signal and $\overline{REQ}$-signal hardware readable
- Internal low-pass filter and 11-bit D/A converter

QUICK REFERENCE DATA

parameter	symbol	min.	typ.	max.	unit
Supply voltage	V_{DD}	–	5	–	V
Supply current	I_{DD}	–	12	#	mA
Supply current (stand-by)	$I_{DD(SB)}$	–	1	–	μA
Inputs					
Input voltage	V_{IH}	2,0	–	V_{DD}	V
Input voltage	V_{IL}	0	–	0,8	V
Input capacitance	C_I	–	7	–	pF
Outputs (D5 to D7)					
Output voltage high	V_{OH}	3,5	–	V_{DD}	V
Output voltage low	V_{OL}	0	–	0,4	V
Load capacitance	C_L	–	–	80	pF
Operating ambient temperature range	T_{amb}	−40	–	+ 85	°C

Value not yet available.

PACKAGE OUTLINE

24-lead DIL; plastic (SOT-101A).

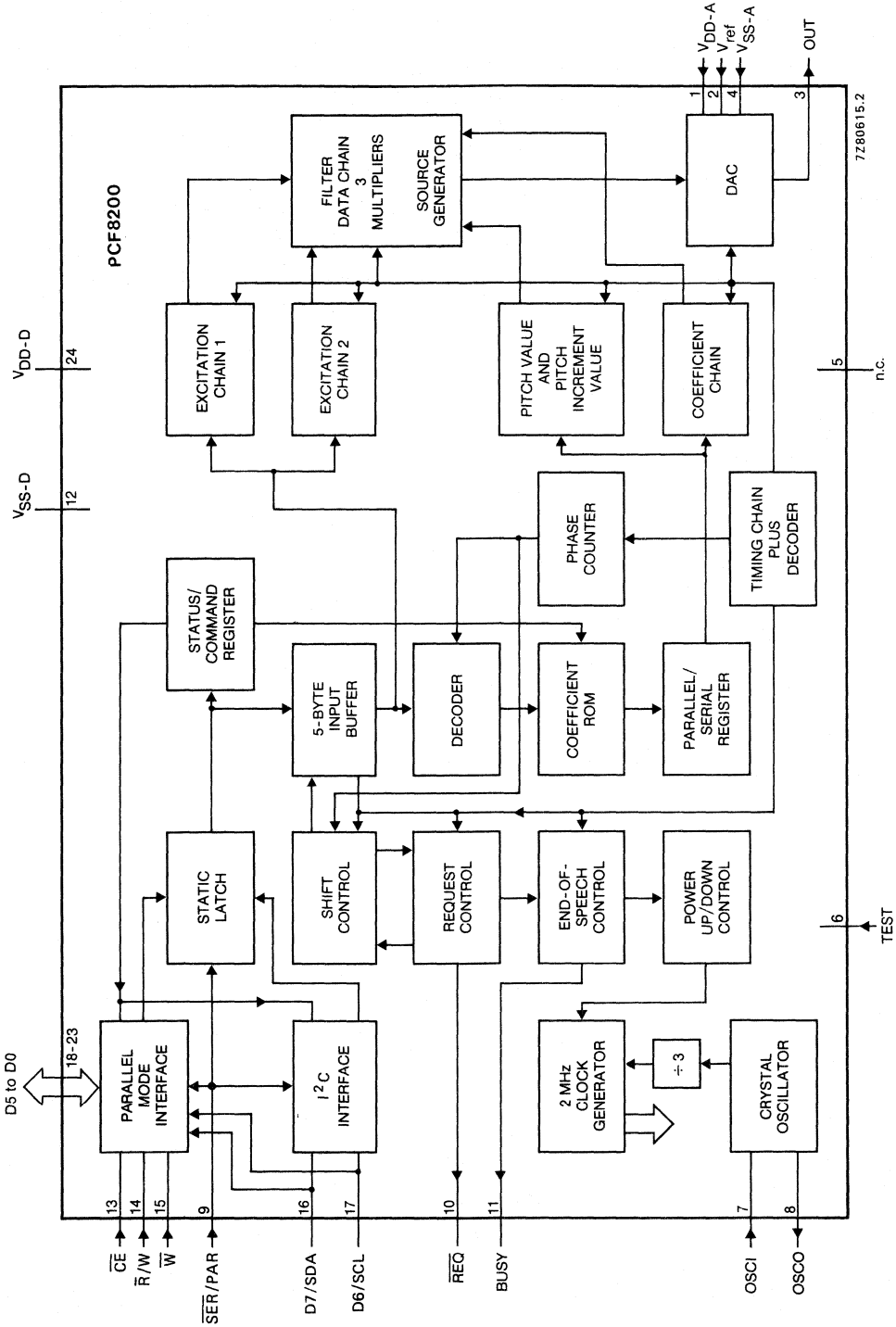


Fig. 1 Block diagram.

PINNING

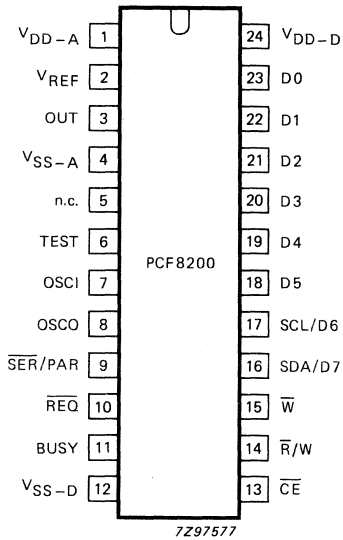


Fig. 2 Pinning diagram.

DEVELOPMENT DATA

1	V_{DD-A}	positive supply voltage for DAC output stage
2	V_{REF}	DAC reference voltage input
3	OUT	speech output
4	V_{SS-A}	negative supply voltage for DAC stage
5	n.c.	not connected
6	TEST	for normal operation this pin must be grounded (V_{SS})
7	OSCI	oscillator input
8	OSCO	oscillator output
9	$\overline{SER}$/PAR	for parallel data bus operation this pin is hard-wired to V_{DD}, or to V_{SS} to enable the I^2C bus
10	$\overline{REQ}$	status bit indicating request for data
11	BUSY	status indicating synthesizer busy
12	V_{SS-D}	negative supply voltage for digital circuits
13	$\overline{CE}$	chip-enable input
14	$\overline{R}$/W	read/write control input
15	$\overline{W}$	write input
16	SDA/D7	I^2C bus serial data input/output (serial mode) or parallel data input/output D7 (parallel mode)
17	SCL/D6	I^2C bus serial clock input/output (serial mode) or parallel data input/output D6 (parallel mode)
18	D5	parallel data input/outputs
19	D4	
20	D3	
21	D2	
22	D1	
23	D0	
24	V_{DD-D}	positive supply voltage for digital circuits

FUNCTIONAL DESCRIPTION

The synthesizer has been designed for a vocal tract modelling technique of voice synthesis. An excitation signal is fed to a series of resonators. Each resonator simulates one of the formants in the original speech. It is controlled by two parameters, one for the resonant frequency and one for the bandwidth. Five formants are needed for male speech and four for female speech. The output of this system is defined by the excitation signal, the amplitude values and the resonator settings. By periodic updating of all parameters very high quality speech can be produced.

OPERATION

Speech characteristics change quite slowly, therefore the control parameters for the speech synthesizer can be adequately updated every few tens of milliseconds with interpolation during the interval to ensure a smooth changeover from one parameter value to the next. In the PCF8200 the standard-frame duration can be set to 8,8 , 10,4, 12,8 or 17,6 milliseconds with the speed-option, speaking speed, in the command-register.
The duration of each individual speech frame is programmable to be 1, 2, 3 or 5 times the standard-frame duration.

FD1, FD0 \ FS0, FS1	10	01	00	11	
00	8,8	10,4	12,8	17,6	ms
01	17,6	20,8	25,6	35,2	ms
10	26,4	31,2	38,4	52,8	ms
11	44,0	52,0	64,0	88,0	ms

Table 1. Frame duration as a function of speed-option (FS1, FS0) and frame-duration (FD1, FD0).

The excitation signal is a random noise source for unvoiced sounds and a programmable pulse generator for voiced sounds. Both sources have an amplitude modulator which is updated 8 times in one speech-frame by linear interpolation. The pitch is updated every 1/8 of a standard frame.

The excitation signal is filtered with a five formant filter for male speech and a four formant filter for female speech. The formant filter is a cascade of all second-order sections. The control parameters, formant-frequency and formant-bandwidth, are updated eight times per speech frame by linear interpolation.
A block diagram of the formant synthesizer is shown in Fig. 3.

The filter output is upsampled to 80 kHz and filtered with a digital low-pass filter. Before the signal is digital to analogue converted (DAC), with an 11-bit switched capacitor DAC, the signal is multiplied with a DAC-amplitude factor. The use of a digital filter means that no external audio filtering is required for low-medium applications and minimal filtering is required for those applications requiring very high quality speech.

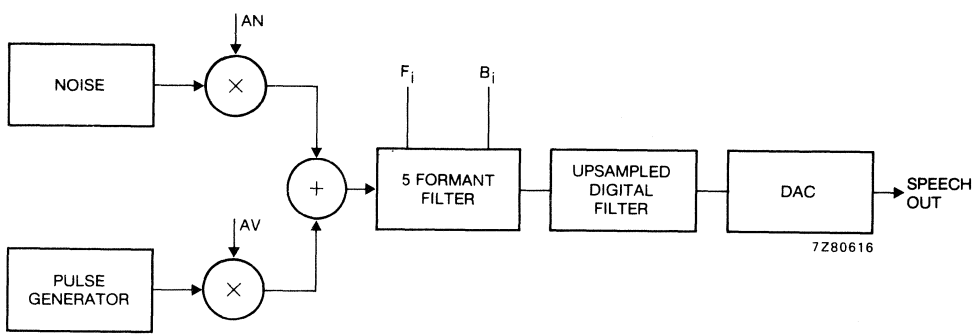


Fig. 3 Block diagram of formant synthesizer.

DATA FORMAT

Three types of format are used for data transfer to the synthesizer.

DAC-amplitude factor

The DAC-amplitude factor is one byte, which is used to optimize the digital speech signal to the 11-bit DAC. It is the first byte after a STOP or a BADSTOP or V_{DD} on. Table 2 indicates the amplitude factor.

byte	factor	dB
01110000	3,5	10,88
10110000	3,25	10,24
00110000	3,0	9,54
11010000	2,75	8,97
01010000	2,5	7,96
10010000	2,25	7,04
00010000	2,0	6,02
11100000	1,75	4,86
01100000	1,5	3,52
10100000	1,25	1,94
00100000	1,0	0,00
11000000	0,75	−2,50
01000000	0,5	−6,02
10000000	0,25	−12,04
00000000	0,0	
11110000	HEX code F0 is not allowed as a DAC amplitude	

Table 2 DAC amplitude factor.

Start pitch

The second byte after a STOP or BADSTOP, or V_{DD} on is the start pitch. It is a one byte start value for the on-chip pitch-period generator.

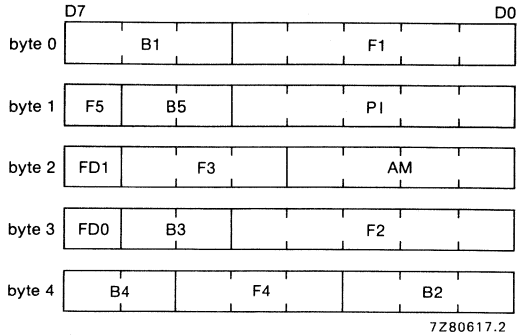
Frame Data

The frame data is a five byte block which contains the filter and source information:

pitch increment/decrement value	5 bits
amplitude	4 bits
frame duration	2 bits
frequency of 1st formant	5 bits
frequency of 2nd formant	5 bits
frequency of 3rd formant	3 bits
frequency of 4th formant	3 bits
frequency of 5th formant	1 bit
bandwidth of 1st formant	3 bits
bandwidth of 2nd formant	3 bits
bandwidth of 3rd formant	2 bits
bandwidth of 4th formant	2 bits
bandwidth of 5th formant	2 bits
	40 bits = 5 bytes

The frame-data bits are organized as shown in Fig. 4.

DEVELOPMENT DATA



It is not allowed to set byte 0 to the hexadecimal value 00.

Fig. 4 Format of frame-date.

CONTROL FORMAT

Command Write

A command write consists of two bytes, and it may occur before a data block. The four bits which can be written are shown in Fig. 5.

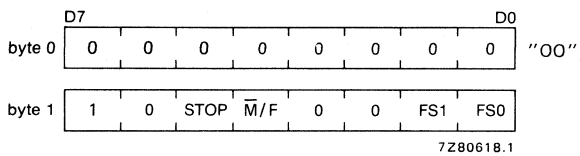


Fig. 5 Control write: first byte fixed, second byte control.

FS0, FS1 speed option

FS1	FS0	speech speed	standard-frame duration
0	0	100%	12,8 ms
0	1	145%	8,8 ms
1	0	123%	10,4 ms
1	1	73%	17,6 ms

$\overline{M}$/F, male/female option

$\overline{M}$/F = 0 male quantization table
= 1 female quantization table

STOP

STOP = 1 stop; repeat last complete frame with amplitude = 0 (no excitation signal)
= 0 if the frame data is not sent within the duration of a half frame, there will be a BADSTOP:

1. $\overline{REQ}$ = 1 STOP = 0
2. Repeat last frame with amplitude = 0
3. BUSY = 0

Status Read

Three status bits can be read out at any time without a preceding byte (00). This is shown in Fig. 6.

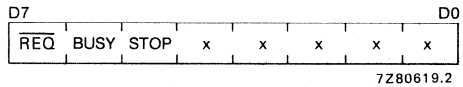


Fig. 6 Status read.

$\overline{REQ}$	= 1	No data required
	= 0	Synthesizer requesting for new data
BUSY	= 1	Busy (an utterance is pronounced)
	= 0	Idle, $\overline{REQ}$ will set to 1; the synthesizer is in STOP or BADSTOP mode
STOP		The STOP bit is the same as the stop bit written to the synthesizer during a command write. STOP = 1, BUSY = 0 stopped by the user. STOP = 0, BUSY = 0 BADSTOP because the data was not sent in time.

After initial power-up the status/command register is set to the following status:

FS0, FS1	= 0	Standard-frame duration of 12,8 ms
$\overline{M}$/F	= 0	Male quantization table
STOP	= 0	
BUSY	= 0	Idle
$\overline{REQ}$	= 1	No data required

DEVELOPMENT DATA

INTERFACE PROTOCOL

Data can be written to the synthesizer when $\overline{REQ}$ = 0 or, when $\overline{REQ}$ = 1 and BUSY = 0. Figure 7 shows the interface protocol of the synthesizer.

In parallel mode the synthesizer is activated by sending the DAC-amplitude factor. In serial mode the DAC-amplitude factor can be sent as soon as the synthesizer is powered-up.

The I^2C transmitter/receiver will then acknowledge. When the request for the pitch-byte occurs the byte must be provided within the duration of a half standard frame. If the byte is not provided in time a BADSTOP will be generated.

During each data write operation, the status bit $\overline{REQ}$ will be set to '1'.
Within a frame data block, it disappears within a few microseconds, asking for the next byte of that block. If the bytes of frame data are not provided within the time-duration of a half frame, a BADSTOP will be generated.

I^2C ADDRESS

On chip there is a I^2C slave receiver/transmitter with the address:

7 6 5 4 3 2 1 0
0 0 1 0 0 0 0 R/W

POWER UP

The synthesizer will be set to power-up on a parallel-write sequence.

PAR-mode: The input-latches are active so they can receive the first byte

SER-mode: The I^2C transmitter/receiver will not acknowledge until the synthesizer has powered-up. To power up the synthesizer a parallel write sequence (Fig. 9) must be made to the synthesizer by using external logic for the control lines; at least one line must be toggled, $\overline{CE}$, while $\overline{W} = 0$ and $\overline{R}/W = 1$.
The synthesizer can be set to permanent power-up by hard-wired control pins ($\overline{CE} = 0$, $\overline{R}/W = 1$, $\overline{W} = 0$).

POWER DOWN MODE

When BUSY = 0 the synthesizer will be set to power-down. In the power-down mode the status/command register will be retained.
In power-down mode the clock-oscillator is switched off. After initial V_{DD} the synthesizer is in power-down mode.

HANDLING

All inputs and outputs are protected against electrostatic charge under normal handling conditions.

RATINGS

Limiting values in accordance with the Absolute Maximum System (IEC 134)

parameter	conditions	symbol	min.	max.	unit
Supply voltage	any pin with respect to V_{SS}	V_{DD}	−0,3	7,5	V
Input voltage	any pin with respect to V_{SS}	V_I	−0,3	7,5	V
Output voltage	any pin with respect to V_{SS}	V_O	−0,3	7,5	V
D.C. input diode current	$V_I < V_{SS}$	$-I_{IK}$	–	20	mA
	$V_I > V_{DD}$	I_{IK}	–	20	mA
D.C. output diode current	$V_O < V_{SS}$	$-I_{OK}$	–	20	mA
	$V_O > V_{DD}$	I_{OK}	–	20	mA
Operating ambient temperature range		T_{amb}	−40	85	°C
Storage temperature range		T_{stg}	−55	125	°C

CHARACTERISTICS

T_{amb} = −45 to + 85 °C; supply voltage (V_{DD} to V_{SS}) = 4,5 to 5,5 V with respect to V_{SS}, unless otherwise specified

parameter	symbol	min.	typ.	max.	unit
Supply					
Supply voltage	V_{DD}	4,5	5,0	5,5	V
Supply current	I_{DD}	–	10	–	mA
Standby current	$I_{DD(SB)}$	–	200	–	μA
Inputs					
$\overline{CE}$, $\overline{R}$/W, $\overline{W}$					
Input voltage HIGH	V_{IH}	2,0	–	V_{DD}	V
Input voltage LOW	V_{IL}	0	–	0,8	V
Input leakage current V_{in} = 0 to 5,5 V	I_{IR}	−10	–	10	μA
Rise and fall times (note 2)	t_{rf}	–	–	50	ns
Input capacitance	C_I	–	–	7	pF
OSCI					
Input voltage HIGH	V_{IH}	2,2	–	V_{DD}	V
Input voltage LOW	V_{IL}	0	–	0,8	V
Input leakage current V_{in} = 0 to 5,5 V	I_{IR}	−10	–	10	μA
Rise and fall times (note 2)	t_{rf}	–	–	50	ns
Input capacitance	C_I	–	–	7	pF
PARALLEL MODE					
Input Characteristics (D0 to D7)					
Input voltage HIGH	V_{IH}	2,0	–	V_{DD}	V
Input voltage LOW	V_{IL}	0	–	0,8	V
Input leakage current (V_{in} = 0 to 5,5 V, output off)	I_{IR}	−10	–	10	μA
Input capacitance	C_I	–	–	7	pF
Output Characteristics (D5 to D7 only)					
Output voltage HIGH (I_{OH} = −100 μA)	V_{OH}	3,5	–	V_{DD}	V
Output voltage LOW (I_{OL} = 3,2 mA)	V_{OL}	0	–	0,4	V
Load capacitance	C_L	–	–	80	pF
Rise and fall times (note 3)	t_{rf}	–	–	50	ns
SERIAL MODE					
Input characteristics (SDA and SDL)					
Input voltage HIGH	V_{IH}	3,0	–	V_{DD}	V
Input voltage LOW	V_{IL}	0	–	1,5	V
Input leakage current (V_{in} = 0 to 5,5 V, output off)	I_{IR}	−10	–	10	μA
Input capacitance	C_I	–	–	10	pF

DEVELOPMENT DATA

parameter	symbol	min.	typ.	max.	unit
Output Characteristics (SDA only, open drain)					
Output voltage LOW (I_{OL} = 3 mA)	V_{OL}	0	–	0,4	V
OSCILLATOR					
Crystal frequency	f_{XTAL}	–	6	6,1	MHz
V_{REF}					
Reference voltage	V_{REF}	1,9	–	$\frac{V_{DD}-1,5}{1,25}$	V
Input leakage current (active)	I_{IR}	–	5	–	μA
Outputs					
$\overline{REQ}$, BUSY					
Output voltage HIGH (I_{OH} = 100 μA)	V_{OH}	3,5	–	V_{DD}	V
Output voltage LOW (I_{OL} = 3,2 mA)	V_{OL}	0	–	0,4	V
Load capacitance	C_L	–	–	80	pF
Rise and fall times (note 3)	t_{rf}	–	–	50	ns
OUT					
Output voltage	V_{OUT}	0,66 x V_{REF}		1,34 x V_{REF}	V
Minimum external load		600	–	–	Ω
Timing characteristics (note 1) (Figs 8 and 9)					
Write enable	t_{WR}	200	–	–	ns
Data set-up for write	t_{DS}	150	–	–	ns
Data hold for write	t_{DH}	30	–	–	ns
Read enable	t_{RD}	200	–	–	ns
Data delay for read (note 2)	t_{DD}	–	–	150	ns
Data floating for read (note 2)	t_{DF}	–	–	150	ns
Control set-up	t_{CS}	0	–	–	ns
Control hold	t_{CH}	0	–	–	ns
REQ new (new byte of the same speech frame)	t_{RN}	–	# (≈ 3)		μs
REQ Valid	t_{RV}	0	–	–	ns
REQ Hold	t_{RH}	–	250	#	ns

NOTES TO THE CHARACTERISTICS

1. Timing reference level is 1,5 V; supply 5 V ± 10%; temperature range of −40 °C to 85 °C.
2. Levels greater than 2 V for a '1' or less than 0,8 V for a '0' are reached with a load of one TTL input and 50 pF.
3. Rise and fall times between 0,6 V and 2,2 V levels.

Values not yet available.

DEVELOPMENT DATA

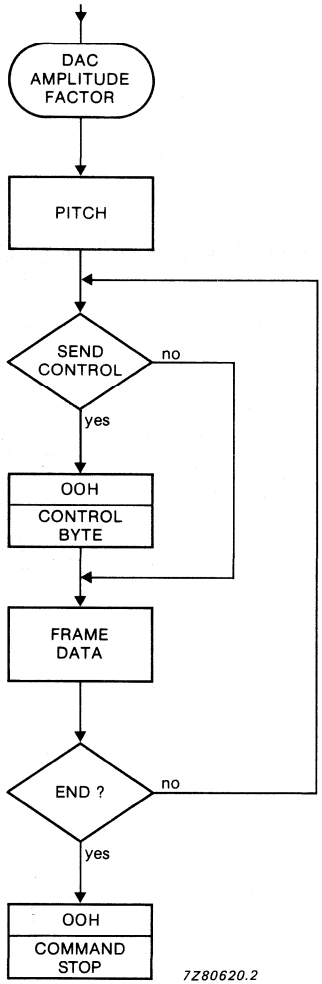
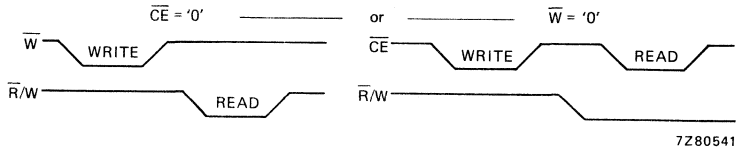


Fig. 7 Interface protocol.

Timing diagrams

The control signals $\overline{CE}$, $\overline{R}/W$ and $\overline{W}$ have been specified to enable easy interface to most microprocessors and microcomputers. For instance with connection to an MAB8048 microcomputer the $\overline{R}/W$ and $\overline{W}$ inputs can be used as the RD and WR strobe inputs.



Typical connection of control signals.

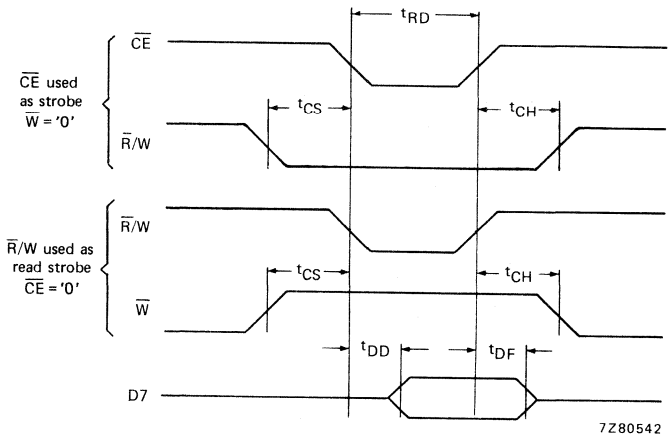


Fig. 8 Read timing.

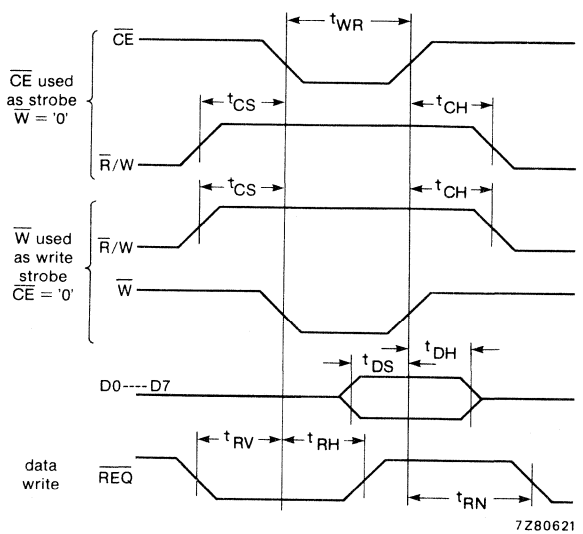


Fig. 9 Write timing.

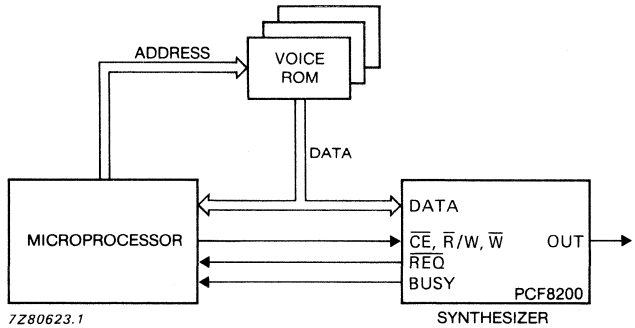


Fig. 10 Typical application configuration with parallel interface.

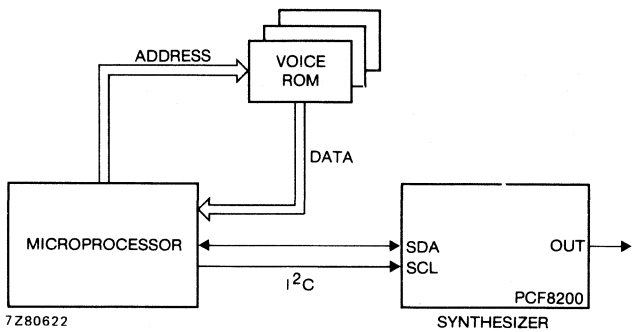


Fig. 11 Typical application configuration with series interface.

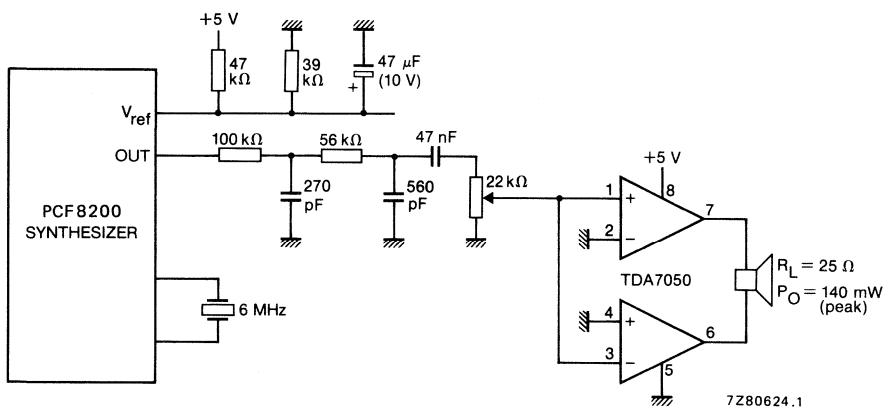


Fig. 12 An example of an output configuration.

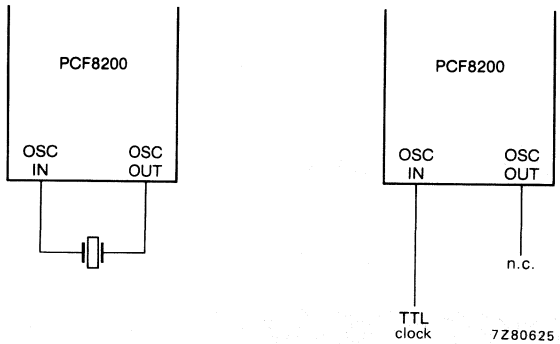


Fig. 13 Oscillator clock configurations.

MICROPROCESSOR CONTROLLED STEREO SOUND GENERATOR FOR SOUND EFFECTS AND MUSIC SYNTHESIS

GENERAL DESCRIPTION

The SAA1099 is a monolithic integrated circuit designed for generation of stereo sound effects and music synthesis.

Features

- Six frequency generators
  - eight octaves per generator
  - 256 tones per octave
- Two noise generators
- Six noise/frequency mixers
- Twelve amplitude controllers
- Two envelope controllers
- Two 6-channel mixers/current sink analogue output stages
- TTL input compatible
- Readily interfaces to 8-bit microcontroller
- Minimal peripheral components
- Simple output filtering

Applications

- Consumer games systems
- Home computers
- Electronic organs
- Arcade games
- Toys
- Chimes/alarm clocks

QUICK REFERENCE DATA

Supply voltage (pin 18)	V_{DD}	typ.	5 V
Supply current (pin 18)	I_{DD}	typ.	70 mA
Reference current (pin 6)	I_{ref}	typ.	250 μA
Total power dissipation	P_{tot}		500 mW
Operating ambient temperature range	T_{amb}		0 to + 70 °C

PACKAGE OUTLINE

18-lead DIL; plastic (SOT-102ME).

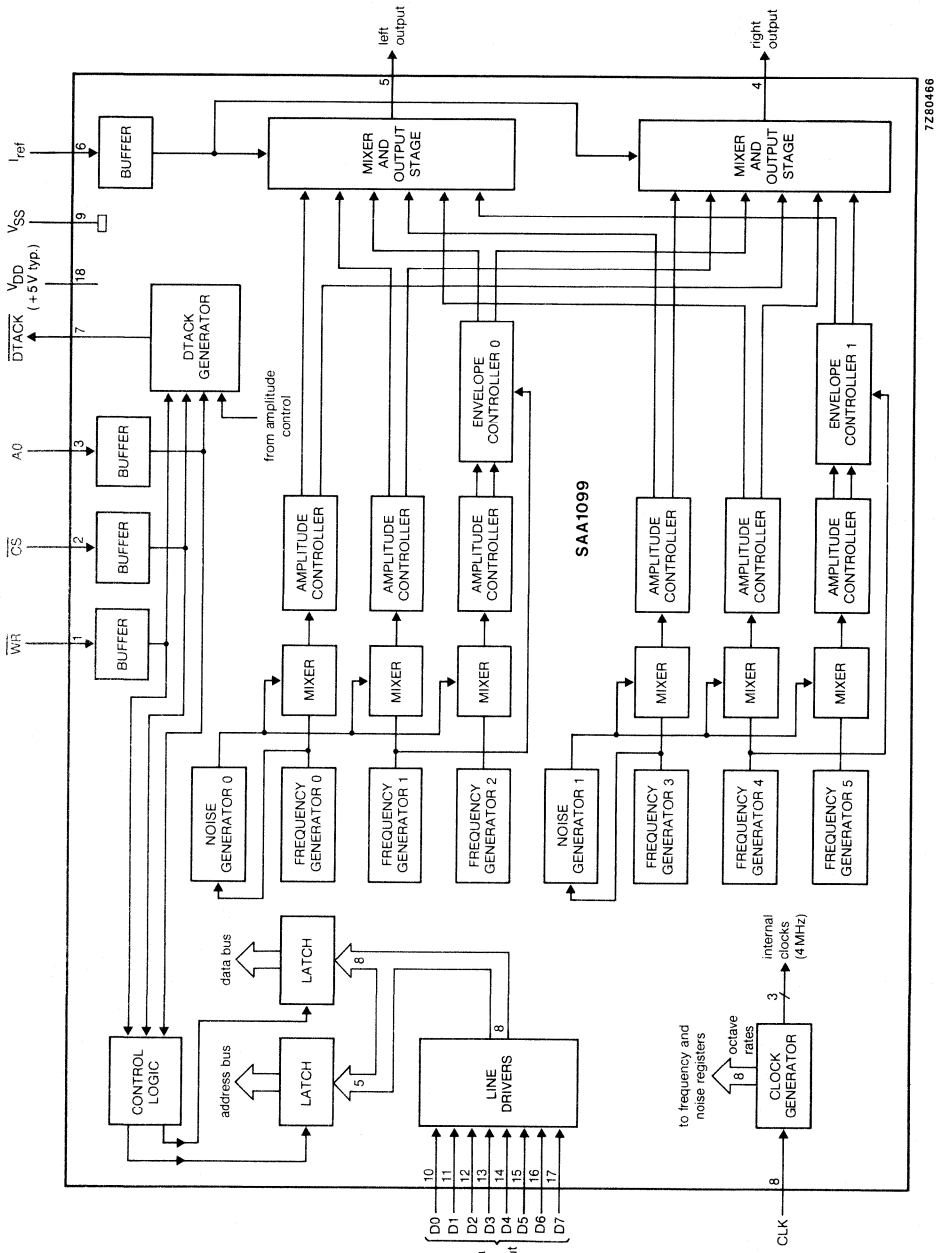


Fig. 1 Block diagram.

PINNING

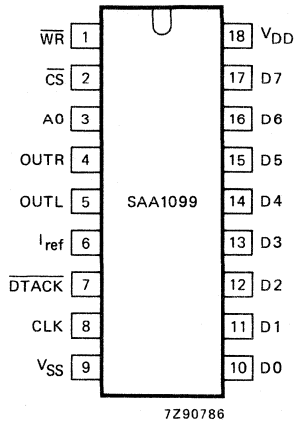


Fig. 2 Pinning diagram.

DEVELOPMENT DATA

PIN DESIGNATION

1	$\overline{WR}$	**Write Enable**: active LOW input which operates in conjunction with $\overline{CS}$ and A0 to allow writing to the internal registers.
2	$\overline{CS}$	**Chip Select**: active LOW input to identify valid $\overline{WR}$ inputs to the chip. This input also operates in conjunction with $\overline{WR}$ and A0 to allow writing to the internal registers.
3	A0	**Control/Address select**: input used in conjunction with $\overline{WR}$ and $\overline{CS}$ to load data to the control register (A0 = 0) or the address buffer (A0 = 1).
4	OUTR	**Right channel output**: a 7-level current sink analogue output for the 'right' component. This pin requires an external load resistor.
5	OUTL	**Left channel output**: a 7-level current sink analogue output for the 'left' component. This pin requires an external load resistor.
6	I_{ref}	**Reference current supply**: used to bias the current sink outputs.
7	$\overline{DTACK}$	**Data Transfer Acknowledge**: open drain output, active LOW to acknowledge successful data transfer. On completion of the cycle $\overline{DTACK}$ is set to inactive.
8	CLK	**Clock**: input for an externally generated clock at a nominal frequency of 8 MHz.
9	V_{SS}	**Ground**: 0 V.
10-17	D0-D7	**Data**: Data bus input.
18	V_{DD}	**Power supply**: + 5 V typical.

FUNCTIONAL DESCRIPTION

The following sections provide a detailed functional description of the SAA1099 as shown in the block diagram, Fig. 1.

Frequency generators

Six frequency generators can each select one of 8 octaves and one of 256 tones within an octave. A total frequency range of 31 Hz to 7,81 kHz is available. The outputs may also control noise or envelope generators. All frequency generators have an enable bit which switches them on and off, making it possible to preselect a tone and to make it inaudible when required. The frequency generators may be synchronized using the frequency reset bit.

The frequency ranges per octave are:

Octave	Frequency range
0	31 Hz to 61 Hz
1	61 Hz to 122 Hz
2	122 Hz to 244 Hz
3	245 Hz to 488 Hz
4	489 Hz to 977 Hz
5	978 Hz to 1,95 kHz
6	1,96 kHz to 3,91 kHz
7	3,91 kHz to 7,81 kHz

Noise generators

The two noise generators both have a programmable output. This may be a software controlled noise via one of the frequency controlled generators or one of three pre-defined noises. There is no tone produced by the frequency generator when it is controlling the noise generator. The noise produced is based on double the frequency generator output, i.e. a range of 61 Hz to 15,6 kHz.

In the event of a pre-defined noise being chosen, the output of noise generator 0 can be mixed with frequency generator 0, 1 and 2; and the output of noise generator 1 can be mixed with frequency generator 3, 4 and 5. In order to produce an equal level of noise and tone outputs (when both are mixed) the amplitude of the tone is increased. The three pre-defined noises are based on a clock frequency of 7,8 kHz, 15,6 kHz or 31,25 kHz.

Noise/frequency mixers

Six noise/frequency mixers each with four selections

- Channel off
- Frequency only
- Noise only
- Noise and frequency

Each mixer channel has one of the frequency generator outputs fed to it, three channels use noise generator 0 and the other three use noise generator 1.

Amplitude controllers

Each of the six channel outputs from the mixer is split up into a right and left component giving effectively twelve amplitude controllers. An amplitude of 16 possible levels is assigned to each of the twelve signals. With this configuration a stereo effect can be achieved by varying only the amplitude component. The moving of a sound from one channel to the other requires, per tone, only one update of the amplitude register contents.

When an envelope generator is used, the amplitude levels are restricted. The number of levels available is then reduced to eight. This is achieved by disabling the least significant bit (LSB) of the amplitude control.

Envelope controllers

Two of the six tone generators are under envelope control. This applies to both the left and right outputs from the tone generator.
The envelope has the following eight possible modes:

- Amplitude is zero
- Single attack
- Single decay
- Single attack-decay (triangular)
- Maximum amplitude
- Continuous attack
- Continuous decay
- Continuous attack-decay

The timing of the envelope controllers is programmable using one of the frequency generators (see Fig. 1). When the envelope mode is selected for a channel its control resolution is halved for that channel from 16 levels to 8 levels by rounding down to the nearest even level.
There is also the capability of controlling the 'right' component of the channel with inverse of the 'left' component, which remains as programmed.

A direct enable permits the start of an envelope to be defined, and also allows termination of an envelope at any time. The envelope rate may be controlled by a frequency channel (see Fig. 1), or by the microprocessor writing to the address buffer register. If the frequency channel controlled is OFF (NE = FE = 0) the envelope will appear at the output, which provides an alternative 'non-square' tone capability. In this event the frequency will be the envelope rate, which provided the rate is from the frequency channel, will be a maximum of 1 kHz. Higher frequencies of up to 2 kHz can be obtained by the envelope resolution being halved from 16 levels to 8 levels. Rates quoted are based on the input of a 8 MHz clock.

Six-channel mixers/current sink analogue output stages

Six channels are mixed together by the two mixers allowing each one to control one of six equally weighted current sinks, to provide a seven level analogue output.

Command/control select

In order to simplify the microprocessor interface the command and control information is multiplexed. To select a register in order to control frequencies, amplitudes, etc. the command-register has to be loaded. The contents of this register determines to which register the data is written in the next control-cycle. If a continuous update of the control-register is necessary, only the control-information has to be written (the command-information does not change).
If the command/control select (A0) is logic 0, the byte transfer is control; if A0 is logic 1, the byte transfer is command.

Interface to microprocessor

The SAA1099 is a data bus based I/O peripheral. Depending on the value of the command/control signal (A0) the $\overline{CS}$ and $\overline{WR}$ signals control the data transfer from the microprocessor to the SAA1099. The data-transfer-acknowledge ($\overline{DTACK}$) indicates that the data transfer is completed. When, during the write cycle, the microprocessor recognizes the $\overline{DTACK}$, the bus cycle will be completed by the processor.

RATINGS

Limiting values in accordance with the Absolute Maximum System (IEC 134)

Supply voltage (pin 18)	V_{DD}	−0,3 to + 7,5	V
Maximum input voltage	V_I	−0,3 to + 7,5	V
at V_{DD} = 4,5 to 5,5 V	V_I	−0,5 to + 7,5	V
Maximum output current	I_O	max. 10	mA
Total power dissipation	P_{tot}	500	mW
Storage temperature range	T_{stg}	−55 to + 125	°C
Operating ambient temperature range	T_{amb}	0 to + 70	°C
Electrostatic handling*	V_{es}	−1000 to + 1000	V

* Equivalent to discharging a 250 μF capacitor through a 1 kΩ series resistor.

DEVELOPMENT DATA

D.C. CHARACTERISTICS

V_{DD} = 5 V ± 10%; T_{amb} = 0 to 70 °C; unless otherwise specified

parameter	symbol	min.	typ.	max.	unit
Supply					
Supply voltage	V_{DD}	4,5	5,0	5,5	V
Supply current	I_{DD}	–	70	100	mA
Reference current (note 1)	I_{ref}	100	250	400	μA
INPUTS					
Input voltage HIGH	V_{IH}	2,0	–	6,0	V
Input voltage LOW	V_{IL}	−0,5	–	0,8	V
Input leakage current	$\pm I_{LI}$	–	–	10	μA
Input capacitance	C_I	–	–	10	pF
OUTPUTS					
$\overline{DTACK}$ (open drain; note 2)					
Output voltage LOW at I_{OL} = 3,2 mA	V_{OL}	0	–	0,4	V
Voltage on pin 7 (OFF state)	V_{7-9}	−0,3	–	6,0	V
Output capacitance (OFF state)	C_O	–	–	10	pF
Load capacitance	C_L	–	–	150	pF
Output leakage current (OFF state)	$-I_{LO}$	–	–	10	μA
Audio outputs (pins 4 and 5)					
With fixed I_{ref} (note 3)					
One channel on	I_{01}/I_{ref}	90	–	120	%
Six channels on	$I_{06}/6xI_{ref}$	85	–	110	%
With I_{ref} = 250 μA; R_L = 1,5 kΩ (± 5%)					
One channel on	I_{01}/I_{ref}	90	–	110	%
Six channels on	$I_{06}/6xI_{ref}$	85	–	105	%
Output current one channel on	I_{01}	225	–	275	μA
Output current six channels on	I_{06}	1,3	–	1,6	mA
With resistor supplying I_{ref} (note 4)					
Output current one channel on	I_{01}	150	–	350	μA
Output current six channels on	I_{06}	0,9	–	1,9	mA
Load resistance	R_L	600	–	–	Ω
D.C. leakage current all channels off	$-I_{LO}$	–	–	10	μA
Maximum current difference between left and right current sinks (note 5)	$\pm I_{Omax}$	–	–	15	%
Signal-to-noise ratio (note 6)	S/N	–	tbf	–	dB

A.C. CHARACTERISTICS

V_{DD} = 5 V ± 10%; T_{amb} = 0 to 70 °C; timing measurements taken at 2,0 V for a logic 1 and 0,8 V for a logic 0 unless otherwise specified (see waveforms Figs 3 and 4)

parameter	symbol	min.	typ.	max.	unit
Bus interface timing (see Fig. 3)					
A0 set-up time to $\overline{CS}$ fall	t_{ASC}	0	–	–	ns
$\overline{CS}$ LOW to $\overline{WR}$ fall	t_{CSW}	30	–	–	ns
A0 set-up time to $\overline{WR}$ fall	t_{ASW}	50	–	–	ns
$\overline{WR}$ LOW time	t_{WL}	100	–	–	ns
Data bus valid to $\overline{WR}$ rise	t_{BSW}	100	–	–	ns
$\overline{DTACK}$ fall delay from $\overline{WR}$ fall (note 7)	t_{DFW}	0	–	85	ns
A0 hold time from $\overline{WR}$ HIGH	t_{AHW}	0	–	–	ns
$\overline{CS}$ hold time from $\overline{WR}$ HIGH	t_{CHW}	0	–	–	ns
Data bus hold time from $\overline{WR}$ HIGH	t_{DHW}	0	–	–	ns
$\overline{DTACK}$ rise delay from $\overline{WR}$ HIGH	t_{DRW}	0	–	100	ns
Bus cycle time (note 8)	t_{CY}	$4t_{CLK}$	–	–	
Bus cycle time (note 9)	t_{CY}	$16t_{CLK}$	–	–	
Clock input timing (see Fig. 4)					
Clock period	t_{CLK}	120	125	255	ns
Clock LOW time	t_{LOW}	55	–	–	ns
Clock HIGH time	t_{HIGH}	55	–	–	ns

Notes to the characteristics

1. Using an external constant current generator to provide a nominal I_{ref} or external resistor connected to V_{DD}.
2. This output is short-circuit protected to V_{DD} and V_{SS}.
3. Measured with I_{ref} a constant value between 100 and 400 μA; load resistance (R_L) allowed to match E12 (5%) in all applications via:

$$R_L = 0{,}6\ [I_{ref}]^{-1} - 16\ [I_{ref}]^{-0{,}5} \pm 12\%$$

4. Measured with R_{ref} = 10 kΩ (± 5%) connected between I_{ref} and V_{DD}; R_L = 1,5 kΩ (± 5%); OUTR and OUTL short-circuit protected to V_{SS}.
5. Left and right outputs must be driven with identical configuration.
6. Sample tested value only.
7. This timing parameter only applies when no wait states are required; otherwise parameter is invalid.
8. The minimum bus cycle time of four clock periods is for loading all registers except the amplitude registers.
9. The minimum bus cycle time of 16 clock periods is for loading the amplitude registers. In a system using $\overline{DTACK}$ it is possible to achieve minimum times of 500 ns. Without $\overline{DTACK}$ the parameter given must be used.

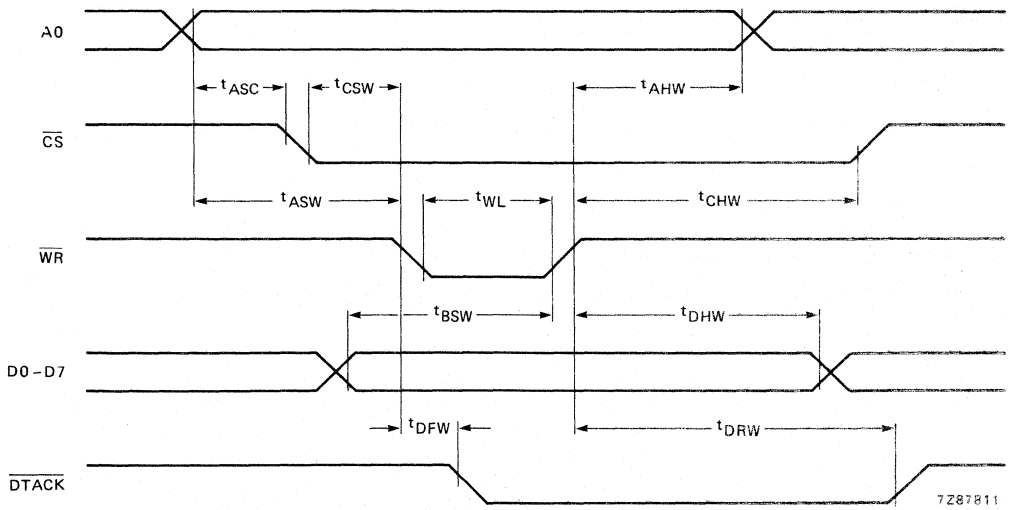


Fig. 3 Bus interface waveforms.

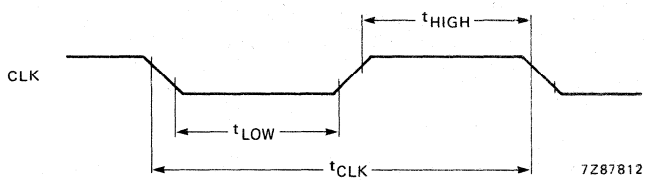


Fig. 4 Clock input waveform.

DEVELOPMENT DATA

APPLICATION INFORMATION

Device operation

The SAA1099 uses pulse width modulation to achieve amplitude and envelope levels. The twelve signals are mixed in an analogue format (6 'left' and 6 'right') before leaving the chip. The amplitude and envelope signals chop the output at a minimum rate of 62,5 kHz, compared with the highest tone output of 7,81 kHz. Simple external low-pass filtering is used to remove the high frequency components.

Rates quoted are based on the input of a 8 MHz clock.

A data bus based write only structure is used to load the on-board registers. The data bus is used to load the address for a register, and subsequently the data to that register. Once the address is loaded multiple data loads to that register can be performed.

The selection of address or data is made by the single address bit A0, as shown in register maps Table 1 and Table 2.

The bus control signals $\overline{WR}$ and $\overline{CS}$ are designed to be compatible with a wide range of microprocessors, a $\overline{DTACK}$ output is included to optimise the interface with an S68000 series microprocessor. In most bus cycles $\overline{DTACK}$ will be returned immediately, this applies to all register address load cycles and all except amplitude data load cycles. With respect to amplitude data, a number of wait cycles may need to be performed, depending on the time since the previous amplitude load. $\overline{DTACK}$ will indicate the number of required waits.

Register description (see Tables 2 and 3)

The amplitudes are assigned with 'left' and 'right' components in the same byte, on a channel by channel basis. The spare locations that are left between blocks of registers is to allow for future expansion, and should be written as zero's. The tone within an octave is defined by eight bits and the octave by three bits. Note that octaves are paired (0/1, 2/3 etc.). The frequency and noise enables are grouped together for ease of programming. The controls for noise 'colour' (clock rate) are grouped in one byte.

The envelope registers are positioned in adjacent locations. There are two types of envelope controls, direct acting controls and buffered controls. The direct acting controls always take immediate effect, and are:

- Envelope enable (reset)
- Envelope resolution (16/8 level)

The buffered controls are acted upon only at the times shown in Fig. 5 and control selection of:

- Envelope clock source
- Waveform type
- Inverted/non-inverted 'right' component

Table 1 External memory map

select	data bus inputs								operations
A0	D7	D6	D5	D4	D3	D2	D1	D0	
0	D7	D6	D5	D4	D3	D2	D1	D0	data for internal registers
1	X	X	X	A4	A3	A2	A1	A0	internal register address

Where X = don't care state.

DEVELOPMENT DATA

Table 2 Internal register map

register address	data bus inputs D7	D6	D5	D4	D3	D2	D1	D0	operations
00	AR03	AR02	AR01	AR00	AL03	AL02	AL01	AL00	amplitude 0 right channel; left channel
01	1	1	1	1	1	1	1	1	amplitude 1 right/left
02	2	2	2	2	2	2	2	2	amplitude 2 right/left
03	3	3	3	3	3	3	3	3	amplitude 3 right/left
04	4	4	4	4	4	4	4	4	amplitude 4 right/left
05	5	5	5	5	5	5	5	5	amplitude 5 right/left
06	X	X	X	X	X	X	X	X	
07	X	X	X	X	X	X	X	X	
08	F07	F06	F05	F04	F03	F02	F01	F00	frequency of tone 0
09	1	1	1	1	1	1	1	1	frequency of tone 1
0A	2	2	2	2	2	2	2	2	frequency of tone 2
0B	3	3	3	3	3	3	3	3	frequency of tone 3
0C	4	4	4	4	4	4	4	4	frequency of tone 4
0D	F57	F56	F55	F54	F53	F52	F51	F50	frequency of tone 5
0E	X	X	X	X	X	X	X	X	
0F	X	X	X	X	X	X	X	X	
10	X	012	011	010	X	002	001	000	octave 1; octave 0
11	X	032	031	030	X	022	021	020	octave 3; octave 2
12	X	052	051	050	X	042	041	040	octave 5; octave 4
13	X	X	X	X	X	X	X	X	
14	X	X	FE5	FE4	FE3	FE2	FE1	FE0	frequency enable
15	X	X	NE5	NE4	NE3	NE2	NE1	NE0	noise enable
16	X	X	N11	N10	X	X	N01	N00	noise generator 1; noise generator 0
17	X	X	X	X	X	X	X	X	
18	E07	X	E05	E04	E03	E02	E01	E00	envelope generator 0
19	E17	X	E15	E14	E13	E12	E11	E10	envelope generator 1
1A	X	X	X	X	X	X	X	X	
1B	X	X	X	X	X	X	X	X	
1C	X	X	X	X	X	X	RST	SE	frequency reset (all channels) sound enable (all channels)
1D	X	X	X	X	X	X	X	X	
1E	X	X	X	X	X	X	X	X	
1F	X	X	X	X	X	X	X	X	

Where:
All don't cares (X) should be written as zero's.
00 to 1F block of registers repeats eight times in the block between addresses 00 to FF (full internal memory map).

APPLICATION INFORMATION (continued)

Table 3 Register description

bit	description
ARn3; ARn2; ARn1; ARn0 (n = 0,5)	4 bits for amplitude control of right channel 0 0 0 0 minimum amplitude (off) 1 1 1 1 maximum amplitude
ALn3; ALn2; ALn1; ALn0 (n = 0,5)	4 bits for amplitude control of left channel 0 0 0 0 minimum amplitude (off) 1 1 1 1 maximum amplitude
Fn7 to Fn0 (n = 0,5)	8 bits for frequency control of the six frequency generators 0 0 0 0 0 0 0 0 lowest frequency 1 1 1 1 1 1 1 1 highest frequency
On2; On1; On0 (n = 0,5)	3 bits for octave control 0 0 0 lowest octave (31 Hz to 61 Hz) 0 0 1 (61 Hz to 122 Hz) 0 1 0 (122 Hz to 244 Hz) 0 1 1 (245 Hz to 488 Hz) 1 0 0 (489 Hz to 977 Hz) 1 0 1 (978 Hz to 1,95 kHz) 1 1 0 (1,96 kHz to 3,91 kHz) 1 1 1 highest octave (3,91 kHz to 7,81 kHz)
FEn (n = 0,5)	frequency enable bit (one tone per generator) FEn = 0 indicates that frequency 'n' is off
NEn (n = 0,5)	noise enable bit (one tone per generator) NEn = 0 indicates that noise 'n' is off
Nn1; Nn0 (n = 0,1)	2 bits for noise generator control. These bits select the noise generator rate (noise 'colour') Nn1 Nn0 clock frequency 0 0 31,3 kHz 0 1 15,6 kHz 1 0 7,6 kHz 1 1 61 Hz to 15,6 kHz (frequency generator 0/3)

bit	description
En7; En5 to En0 (n = 0,1)	7 bits for envelope control En0 0 left and right component have the same envelope 1 right component has inverse of envelope that is applied to left component En3 En2 En1 0 0 0 zero amplitude 0 0 1 maximum amplitude 0 1 0 single decay 0 1 1 repetitive decay 1 0 0 single triangular 1 0 1 repetitive triangular 1 1 0 single attack 1 1 1 repetitive attack En4 0 4 bits for envelope control (maximum frequency = 977 Hz) 1 3 bits for envelope control (maximum frequency = 1,95 kHz) En5 0 internal envelope clock (frequency generator 1 or 4) 1 external envelope clock (address write pulse) En7 0 reset (no envelope control) 1 envelope control enabled
SE	SE sound enable for all channels (reset on power-up to 0) 0 all channels disabled 1 all channels enabled
RST	Reset signal to all frequency generators 0 all generators enabled 1 all generators reset and synchronized

Note

All rates given are based on the input of a 8 MHz clock.

APPLICATION INFORMATION (continued)

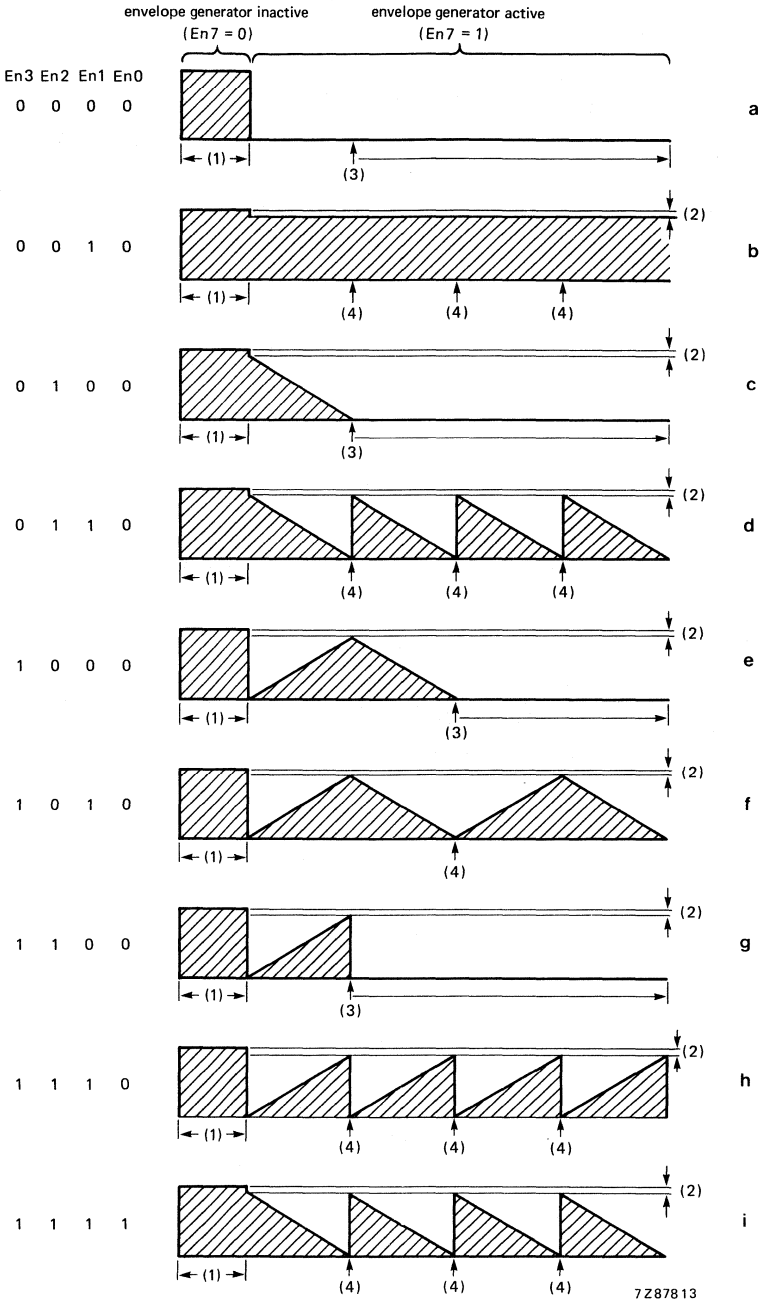


Fig. 5 Envelope waveforms.

Notes to Fig. 5

(1) The level at this time is under amplitude control only (En7 = 0; no envelope).

(2) When the generator is active (En7 = 1) the maximum level possible is 7/8ths of the amplitude level.

(3) After position (3) the buffered controls will be acted upon when loaded.

(4) At positions (4) the buffered controls will be acted upon if already loaded.

(5) Waveforms 'a' to 'h' show the left channel (En0 = 0; left and right components have the same envelope).
Waveform 'i' shows the right channel (En0 = 1; right component inverse of envelope applied to left).

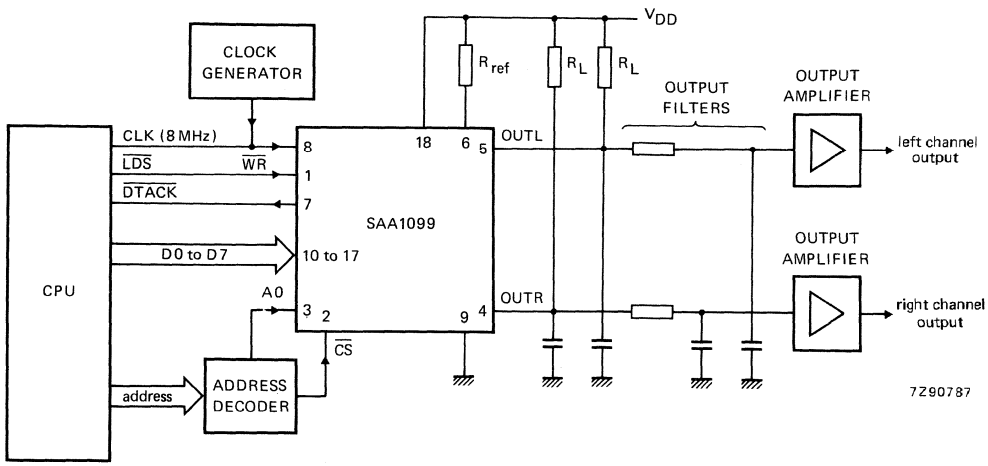
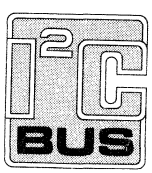


Fig. 6 Typical application circuit diagram.

I²C-BUS PERIPHERALS

User manual 1986 *
"Single-chip 8-bit Microcontrollers"
Available on request

* Includes I²C-bus specification



Purchase of Philips I²C components conveys a licence under the Philips I²C patent to use the components in the I²C system provided the system conforms to the I²C specification defined by Philips.

STATIC CMOS EEPROM (256 x 8 BIT)

GENERAL DESCRIPTION

The PCB8582 is a 2K-bit 5 V electrically erasable programmable read only memory (EEPROM) organized as 256 by 8 bits. It is designed in a floating gate CMOS technology.

As data bytes are received and transmitted via the serial I^2C bus, an eight pin DIL package is sufficient. Up to eight PCB8582 devices may be connected to the I^2C bus.

Chip select is accomplished by three address inputs.

Features

- Non-volatile storage of 2K-bit organized as 256 x 8
- Only one power supply required (5 V)
- On chip voltage multiplier for erase/write
- Serial input/output bus (I^2C)
- Automatic word address incrementing
- Low power consumption
- One point erase/write timer
- Power on reset
- 10 000 erase/write cycles per byte
- 10 years non-volatile data retention
- Infinite number of read cycles
- Pin and address compatible to PCF8570, PCF8571 and PCD8572

- A version for extended temperature range, −40 to + 85 °C, in preparation: PCF8582. ←

PACKAGE OUTLINE

PCB8582P: 8-lead DIL; plastic (SOT-97AE). ←

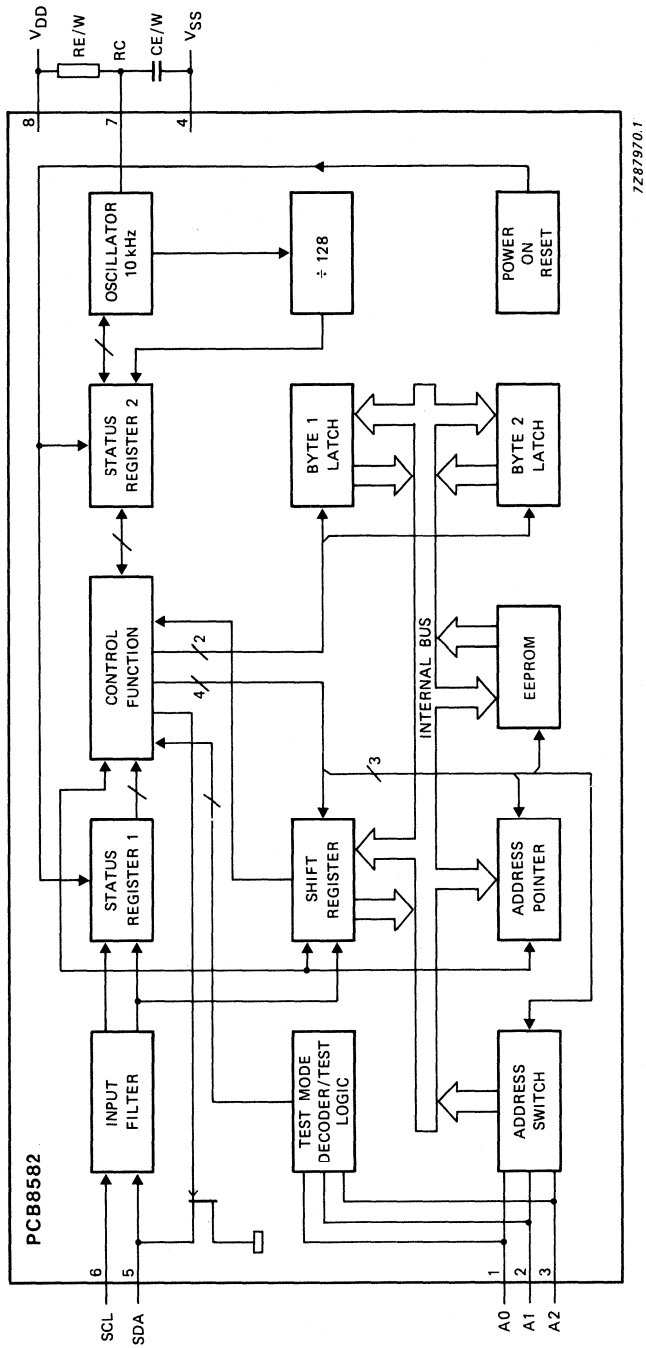


Fig. 1 Block diagram.

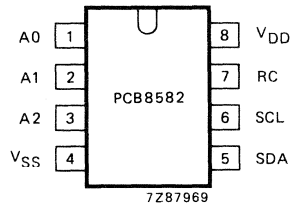


Fig. 2 Pinning diagram.

1 A0
2 A1 address inputs/test
3 A2 mode select
4 V_{SS} ground
5 SDA
6 SCL
I²C bus lines
7 RC input for timer constant
8 V_{DD} positive supply

FUNCTIONAL DESCRIPTION

Characteristics of the I²C bus

The I²C bus is intended for communication between different ICs. The serial bus consists of two bi-directional lines, one for data signals (SDA), and one for clock signals (SCL). Both the SDA and the SCL lines must be connected to a positive supply voltage via a pull-up resistor.

The following protocol has been defined:
Data transfer may be initiated only when the bus is not busy.
During data transfer, the data line must remain stable whenever the clock line is HIGH. Changes in the data line while the clock line is HIGH will be interpreted as control signals.

Accordingly, the following bus conditions have been defined:
Bus not busy: both data and clock lines remain HIGH.
Start data transfer: a change in the state of the data line, from HIGH to LOW, while the clock is HIGH defines the start condition.
Stop data transfer: a change in the state of the data line, from LOW to HIGH, while the clock is HIGH, defines the stop condition.
Data valid: the state of the data line represents valid data when, after a start condition, the data line is stable for the duration of the HIGH period of the clock signal. The data on the line may be changed during the LOW period of the clock signal. There is one clock pulse per bit of data.

Each data transfer is initiated with a start condition and terminated with a stop condition; the number of the data bytes, transfered between the start and stop conditions is limited to two bytes in the ERASE/WRITE mode and unlimited in the READ mode. The information is transmitted in bytes and each receiver acknowledges with a ninth bit.

Within the I²C bus specifications a low-speed mode (2 kHz clock rate) and a high-speed mode (100 kHz clock rate) are defined. The PCB8582 operates in both modes.
By definition a device that gives out a signal is called a "transmitter", and the device which receives the signal is called a "receiver". The device which controls the signal is called the "master". The devices that are controlled by the master are called "slaves".

Each word of eight bits is followed by one acknowledge bit. This acknowledge bit is a HIGH level put on the bus by the transmitter whereas the master generates an extra acknowledge related clock pulse.
A slave receiver which is addressed is obliged to generate an acknowledge after the reception of each byte.
Also, a master receiver must generate an acknowledge after the reception of each byte that has been clocked out of the slave transmitter.
The device that acknowledges has to pull down the SDA line during the acknowledge clock pulse in such a way that the SDA line is stable LOW during the high period of the acknowledge related clock pulse.
Set-up-and hold times must be taken into account. A master receiver must signal an end of data to the slave transmitter by not generating an acknowledge on the last byte that has been clocked out of the slave. In this case the transmitter must leave the data line HIGH to enable the master generation of the stop condition.

DEVELOPMENT DATA

Note

→ The general characteristics and detailed specification of the I²C bus is available on request.

I²C bus protocol

The I²C bus configuration for different READ and WRITE cycles of the PCB8582 are shown in Fig. 3.

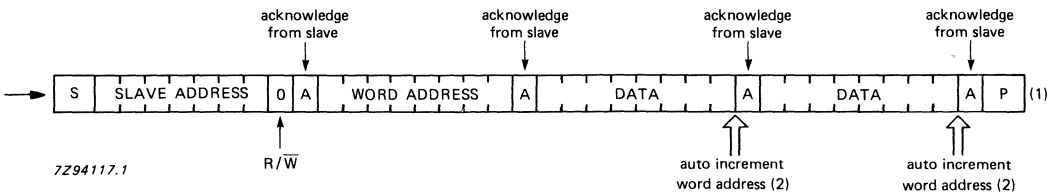


Fig. 3(a) Slave receiver ERASE/WRITE mode.

(1.)After this stop condition the erase/write cycle starts and the bus is free for another transmission; the duration of the erase/write cycle is approximatly 20 ms if only one byte is written, and 40 ms, if two bytes are written. During the erase/write cycle the slave receiver does not send an acknowledge bit if addressed via I²C bus.

→ (2.)The second data byte is voluntary. It is not allowed to erase/write more than two bytes.

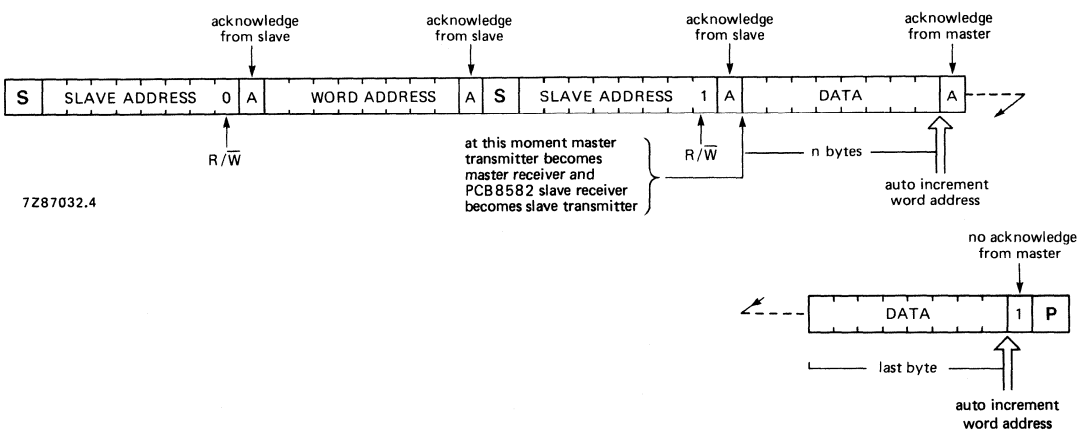


Fig. 3(b) Master reads PCB8582 slave after setting word address. (WRITE word address; READ data).

Note: The slave address is defined in accordance with the I²C bus specification as:

1	0	1	0	A2	A1	A0	R/$\overline{W}$

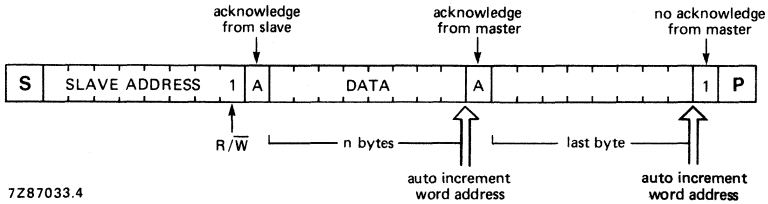


Fig. 3(c) Master reads PCB8582 slave immediately after first byte (READ mode).

I²C bus timing

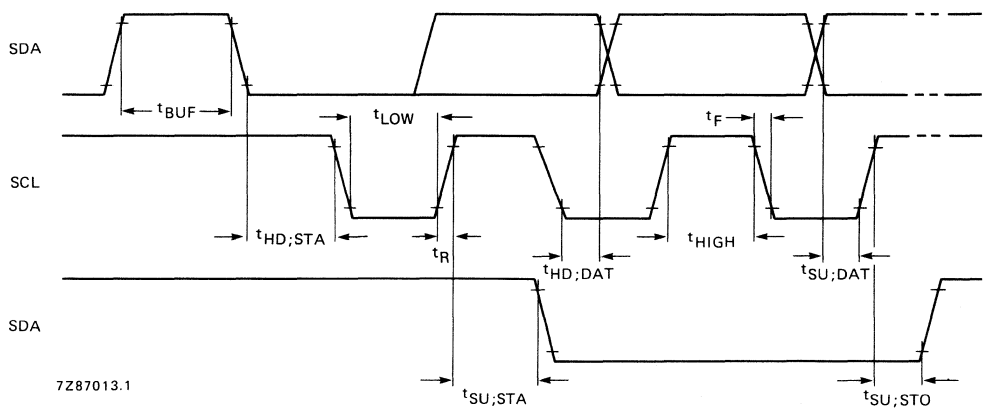


Fig. 4 I²C bus, timing.

RATINGS

Limiting values in accordance with the Absolute Maximum System (IEC 134)

Supply voltage	V_{DD}	−0,3 to 7 V
Voltage, on any input pin (input impedance 500 Ω)	V_I	V_{SS}−0,8 to V_{DD} +0,8 V
Operating temperature range	T_{amb}	0 to +70 °C
Storage temperature range	T_{stg}	−65 to +150 °C
Current into any input pin	I_I	1 mA
Output current	I_O	10 mA

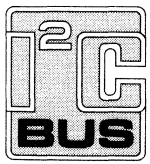
CHARACTERISTICS

V_{DD} = 5 V; V_{SS} = 0 V; T_{amb} = 0 to + 70 °C, unless otherwise specified.

parameter	symbol	min.	typ.	max.	unit
Operating supply voltage	V_{DD}	4,5	5	5,5	V
Operating supply current, READ (f_{SLC} = 100 kHz)	I_{DDR}	–	0,1	0,2	mA
Operating supply current, WRITE/ERASE	I_{DDW}	–	1	2	mA
Standby supply current (V_{DD} = 5 V)	I_{DDO}	–	5	10	μA
Input SCL and input/output SDA					
Input/output SDA:					
Input voltage LOW	V_{IL}	−0,3	–	1,5	V
Input voltage HIGH	V_{IH}	3	–	V_{DD}+0,8	V
Output voltage LOW (I_{OL} = 3 mA, V_{DD} = 4,5 V)	V_{OL}	–	–	0,4	V
Output leakage current HIGH (V_{OH} = V_{DD})	I_{OH}	–	–	1	μA
Input leakage current (A0,A1,A2, SCL), (note 1)	$\pm I_{IN}$	–	–	1	μA
Clock frequency	f_{SCL}	0	–	100	kHz
Input capacity (SCL,SDA)	C_I	–	–	7	pF
Noise suppression time constant at SCL and SDA input	t_I	0,25	0,5	1	μs
Time the bus must be free before a new transmission can start	t_{BUF}	4,7	–	–	μs
Hold time start condition. After this period the first clock pulse is generated	$t_{HD;STA}$	4	–	–	μs
The LOW period of the clock	t_{LOW}	4,7	–	–	μs
The HIGH period of the clock	t_{HIGH}	4	–	–	μs
Set-up time for start condition (only relevent for a repeated start condition)	$t_{SU;STA}$	4,7	–	–	μs
Hold time DATA for:					
CBUS compatible masters	$t_{HD;DAT}$	5	–	–	μs
I^2C devices (note 2)	$t_{HD;DAT}$	0	–	–	μs
Set-up time DATA	$t_{SU;DAT}$	250	–	–	ns
Rise time for both SDA and SCL lines	t_R	–	–	1	μs
Fall time for both SDA and SCL lines	t_F	–	–	300	ns
Set-up time for stop condition	$t_{SU;STO}$	4,7	–	–	μs
Erase/write timer constant (note 3)					
Erase/write cycle time	$t_{E/W}$	20	–	100	ms
Erase/write timing capacitor for erase/write cycle of 30 ms	$C_{E/W}$	–	3,3	–	nF
Erase/write timing resistor for erase/write cycle of 30 ms	$R_{E/W}$	–	56	–	kΩ
Data retention time	t_S	10	–	–	years

Notes to the characteristics

1. Selection of the chip address is done by connecting the A0, A1, and A2 inputs either to V_{SS} or V_{DD}.
2. A transmitter must internally provide a hold time to bridge the undefined region (maximum 300 ns) of the falling edge of SCL.
3. Endurance (number of erase/write cycles), NE/W, is 10^4 E/W cycles.



Purchase of Philips' I²C components conveys a license under the Philips' I²C patent to use the components in the I²C-system provided the system conforms to the I²C specifications defined by Philips.

DEVELOPMENT DATA

UNIVERSAL LCD DRIVER FOR LOW MULTIPLEX RATES

GENERAL DESCRIPTION

The PCF8566 is a peripheral device which interfaces to almost any liquid crystal display (LCD) having low multiplex rates. It generates the drive signals for any static or multiplexed LCD containing up to four backplanes and up to 24 segments and can easily be cascaded for larger LCD applications. The PCF8566 is compatible with most microprocessors/microcontrollers and communicates via a two-line bidirectional bus (I^2C). Communication overheads are minimized by a display RAM with auto-incremented addressing, by hardware subaddressing and by display memory switching (static and duplex drive modes).

Features

- Single-chip LCD controller/driver
- Selectable backplane drive configuration: static or 2/3/4 backplane multiplexing
- Selectable display bias configuration: static, 1/2 or 1/3
- Internal LCD bias generation with voltage-follower buffers
- 24 segment drives: up to twelve 8-segment numeric characters; up to six 15-segment alphanumeric characters; or any graphics of up to 96 elements
- 24 x 4-bit RAM for display data storage
- Auto-incremented display data loading across device subaddress boundaries
- Display memory bank switching in static and duplex drive modes
- Versatile blinking modes
- LCD and logic supplies may be separated
- 3 V to 6 V power supply range
- Low power consumption
- Power-saving mode for extremely low power consumption in battery-operated and telephone applications
- I^2C bus interface
- TTL/CMOS compatible
- Compatible with any 4-bit, 8-bit or 16-bit microprocessors/microcontrollers
- May be cascaded for large LCD applications (up to 1536 segments possible)
- Cascadable with the 40 segment LCD driver PFC8576
- Optimized pinning for single plane wiring in both single and multiple PCF8566 applications
- Space-saving 40-lead plastic mini-pack (VSO-40; SOT-158A)
- No external components required (even in multiple device applications)
- Manufactured in silicon gate CMOS process

Purchase of Philips' I^2C components conveys a license under the Philips' I^2C patent to use the components in the I^2C-system provided the system conforms to the I^2C specification defined by Philips.

PACKAGE OUTLINES

PCF8566T: 40-lead mini-pack; (VSO-40; SOT-158A).
PCF8566P: 40-lead DIL; plastic (SOT-129).

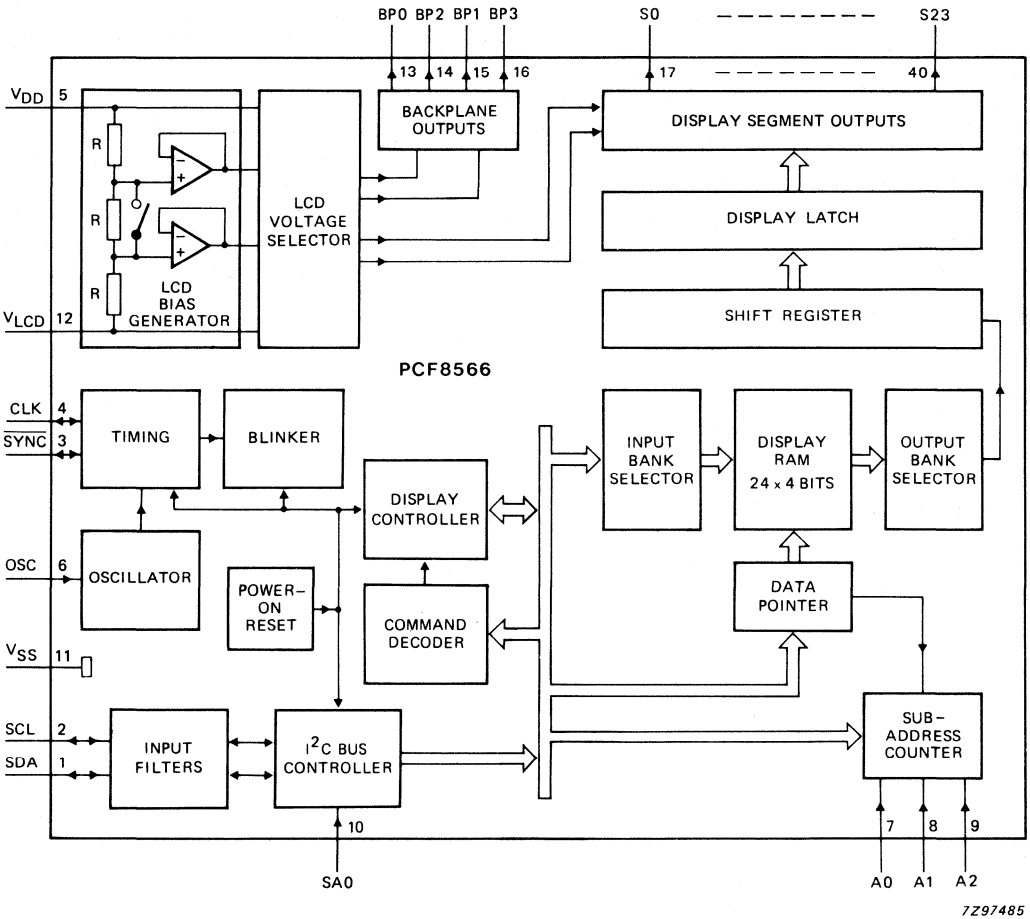


Fig. 1 Block diagram.

DEVELOPMENT DATA

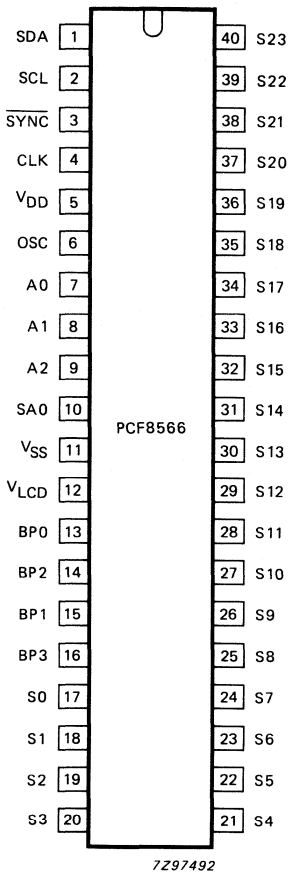


Fig. 2 Pinning diagram.

PINNING

1	SDA	I^2C bus data input/output
2	SCL	I^2C bus clock input/output
3	$\overline{SYNC}$	cascade synchronization input/output
4	CLK	external clock input/output
5	V_{DD}	positive supply voltage
6	OSC	oscillator input
7	A0	I^2C bus subaddress inputs
8	A1	
9	A2	
10	SA0	I^2C bus slave address bit 0 input
11	V_{SS}	logic ground
12	V_{LCD}	LCD supply voltage
13	BP0	LCD backplane outputs
14	BP2	
15	BP1	
16	BP3	
17 to 40	S0 to S23	LCD segment outputs

FUNCTIONAL DESCRIPTION

The PCF8566 is a versatile peripheral device designed to interface any microprocessor to a wide variety of LCDs. It can directly drive any static or multiplexed LCD containing up to four backplanes and up to 24 segments. The display configurations possible with the PCF8566 depend on the number of active backplane outputs required; a selection of display configurations is given in Table 1.

Table 1 Selection of display configurations

active back-plane outputs	no. of segments	7-segment numeric	14-segment alphanumeric	dot matrix
4	96	12 digits + 12 indicator symbols	6 characters + 12 indicator symbols	96 dots (4 x 24)
3	72	9 digits + 9 indicator symbols	4 characters + 16 indicator symbols	72 dots (3 x 24)
2	48	6 digits + 6 indicator symbols	3 characters + 6 indicator symbols	48 dots (2 x 24)
1	24	3 digits + 3 indicator symbols	1 characters + 10 indicator symbols	24 dots

All of the display configurations given in Table 1 can be implemented in the typical system shown in Fig. 3. The host microprocessor/microcontroller maintains the two-line I^2C bus communication channel with the PCF8566. The internal oscillator is selected by tying OSC (pin 6) to V_{SS}. The appropriate biasing voltages for the multiplexed LCD waveforms are generated internally. The only other connections required to complete the system are to the power supplies (V_{DD}, V_{SS} and V_{LCD}) and to the LCD panel chosen for the application.

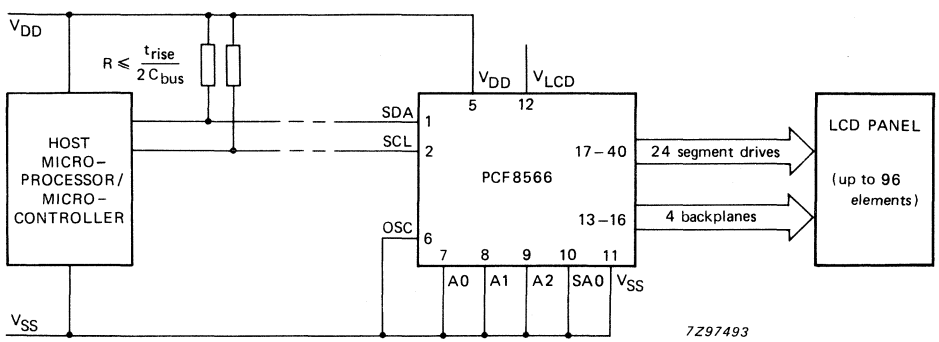


Fig. 3 Typical system configuration.

Power-on reset

At power-on the PCF8566 resets to a defined starting condition as follows:

1. All backplane outputs are set to V_{DD}.
2. All segment outputs are set to V_{DD}.
3. The drive mode '1 : 4 multiplex with 1/3 bias' is selected.
4. Blinking is switched off.
5. Input and output bank selectors are reset (as defined in Table 5).
6. The I^2C bus interface is initialized.
7. The data pointer and the subaddress counter are cleared.

Data transfers on the I^2C bus should be avoided for 1 ms following power-on to allow completion of the reset action.

LCD bias generation

The full-scale LCD voltage (V_{op}) is obtained from $V_{DD} - V_{LCD}$. The LCD voltage may be temperature compensated externally through the V_{LCD} supply to pin 12. Fractional LCD biasing voltages are obtained from an internal voltage divider of three series resistors connected between V_{DD} and V_{LCD}. The centre resistor can be switched out of circuit to provide a ½ bias voltage level for the 1 : 2 multiplex configuration.

LCD voltage selector

The LCD voltage selector coordinates the multiplexing of the LCD according to the selected LCD drive configuration. The operation of the voltage selector is controlled by MODE SET commands from the command decoder. The biasing configurations that apply to the preferred modes of operation, together with the biasing characteristics as functions of $V_{op} = V_{DD} - V_{LCD}$ and the resulting discrimination ratios (D), are given in Table 2.

Table 2 Preferred LCD drive modes: summary of characteristics

LCD drive mode	LCD bias configuration	$\frac{V_{off(rms)}}{V_{op}}$	$\frac{V_{on\ (rms)}}{V_{op}}$	$D = \frac{V_{on(rms)}}{V_{off(rms)}}$
static (1 BP)	static (2 levels)	0	1	∞
1 : 2 MUX (2 BP)	1/2 (3 levels)	$\sqrt{2}/4 = 0{,}354$	$\sqrt{10}/4 = 0{,}791$	$\sqrt{5} = 2{,}236$
1 : 2 MUX (2 BP)	1/3 (4 levels)	$1/3 = 0{,}333$	$\sqrt{5}/3 = 0{,}745$	$\sqrt{5} = 2{,}236$
1 : 3 MUX (3 BP)	1/3 (4 levels)	$1/3 = 0{,}333$	$\sqrt{33}/9 = 0{,}638$	$\sqrt{33}/3 = 1{,}915$
1 : 4 MUX (4 BP)	1/3 (4 levels)	$1/3 = 0{,}333$	$\sqrt{3}/3 = 0{,}577$	$\sqrt{3} = 1{,}732$

DEVELOPMENT DATA

LCD voltage selector (continued)

A practical value for V_{op} is determined by equating $V_{off(rms)}$ with a defined LCD threshold voltage (V_{th}), typically when the LCD exhibits approximately 10% contrast. In the static drive mode a suitable choice is $V_{op} \gtrapprox 3\ V_{th}$.
Multiplex drive ratios of 1 : 3 and 1 : 4 with 1/2 bias are possible but the discrimination and hence the contrast ratios are smaller ($\sqrt{3} = 1{,}732$ for 1 : 3 multiplex or $\sqrt{21}/3 = 1{,}528$ for 1 : 4 multiplex).
The advantage of these modes is a reduction of the LCD full scale voltage V_{op} as follows:

1 : 3 multiplex (1/2 bias) : $V_{op} = \sqrt{6}\ V_{off(rms)} = 2{,}449\ V_{off(rms)}$

1 : 4 multiplex (1/2 bias) : $V_{op} = 4\sqrt{3}/3\ V_{off(rms)} = 2{,}309\ V_{off(rms)}$

These compare with $V_{op} = 3\ V_{off(rms)}$ when 1/3 bias is used.

LCD drive mode waveforms

The static LCD drive mode is used when a single backplane is provided in the LCD. Backplane and segment drive waveforms for this mode are shown in Fig. 4.

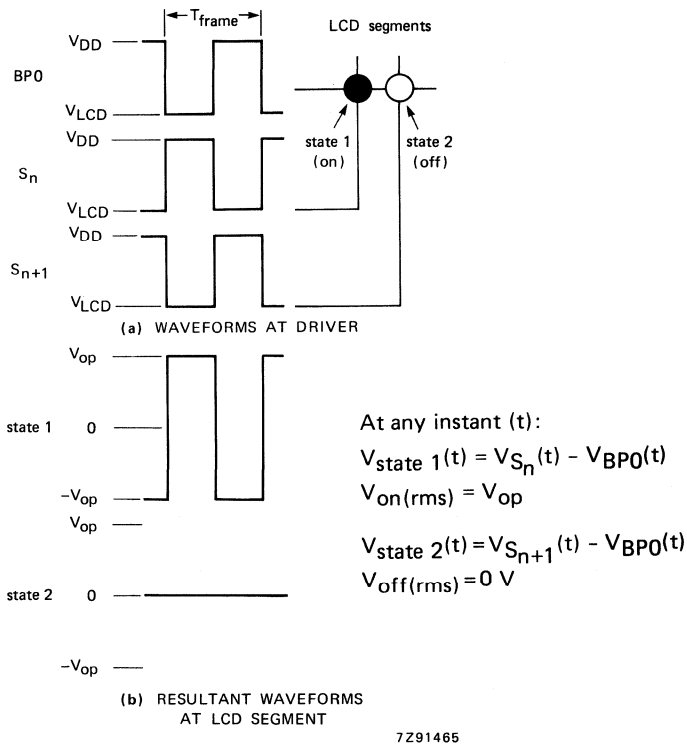


Fig. 4 Static drive mode waveforms: $V_{op} = V_{DD} - V_{LCD}$.

When two backplanes are provided in the LCD the 1 : 2 multiplex drive mode applies. The PCF8566 allows use of 1/2 or 1/3 bias in this mode as shown in Figs 5 and 6.

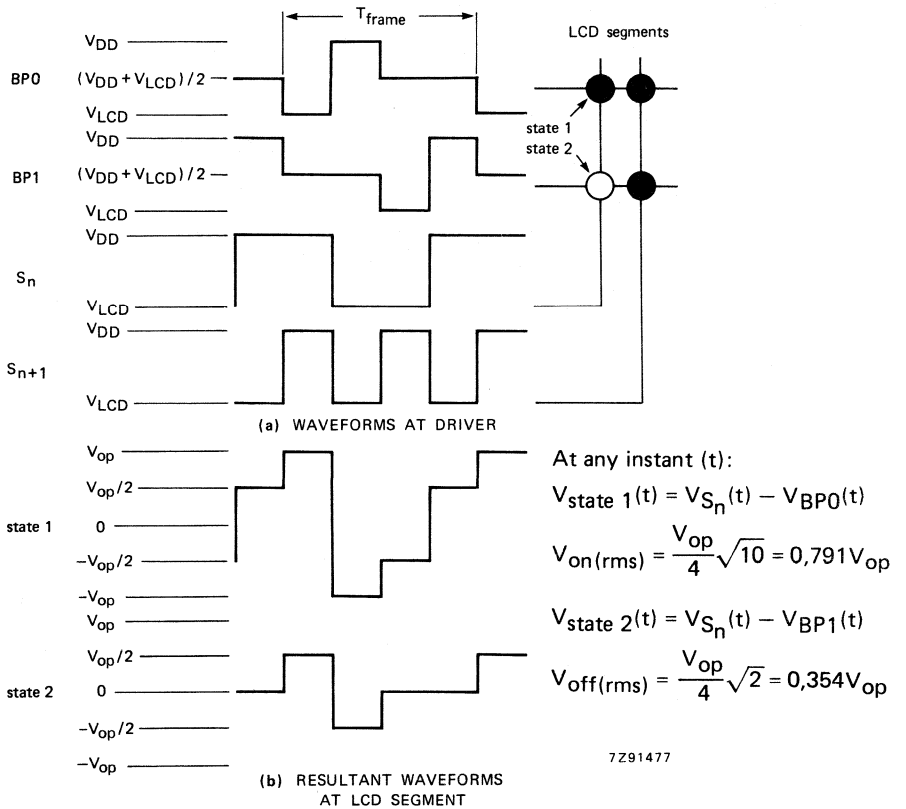


Fig. 5 Waveforms for 1 : 2 multiplex drive mode with 1/2 bias: $V_{op} = V_{DD} - V_{LCD}$.

DEVELOPMENT DATA

LCD drive mode waveforms (continued)

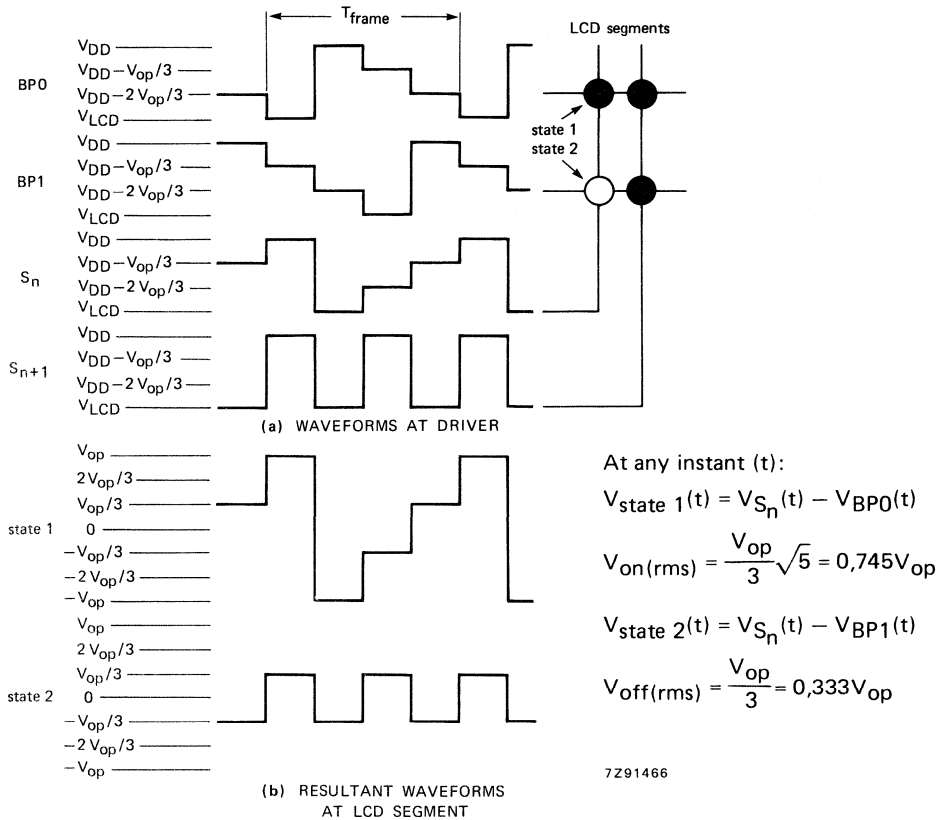


Fig. 6 Waveforms for 1 : 2 multiplex drive mode with 1/3 bias: $V_{op} = V_{DD} - V_{LCD}$.

The backplane and segment drive wavefront for the 1 : 3 multiplex drive mode (three LCD backplanes) and for the 1 : 4 multiplex drive mode (four LCD backplanes) are shown in Figs 7 and 8 respectively.

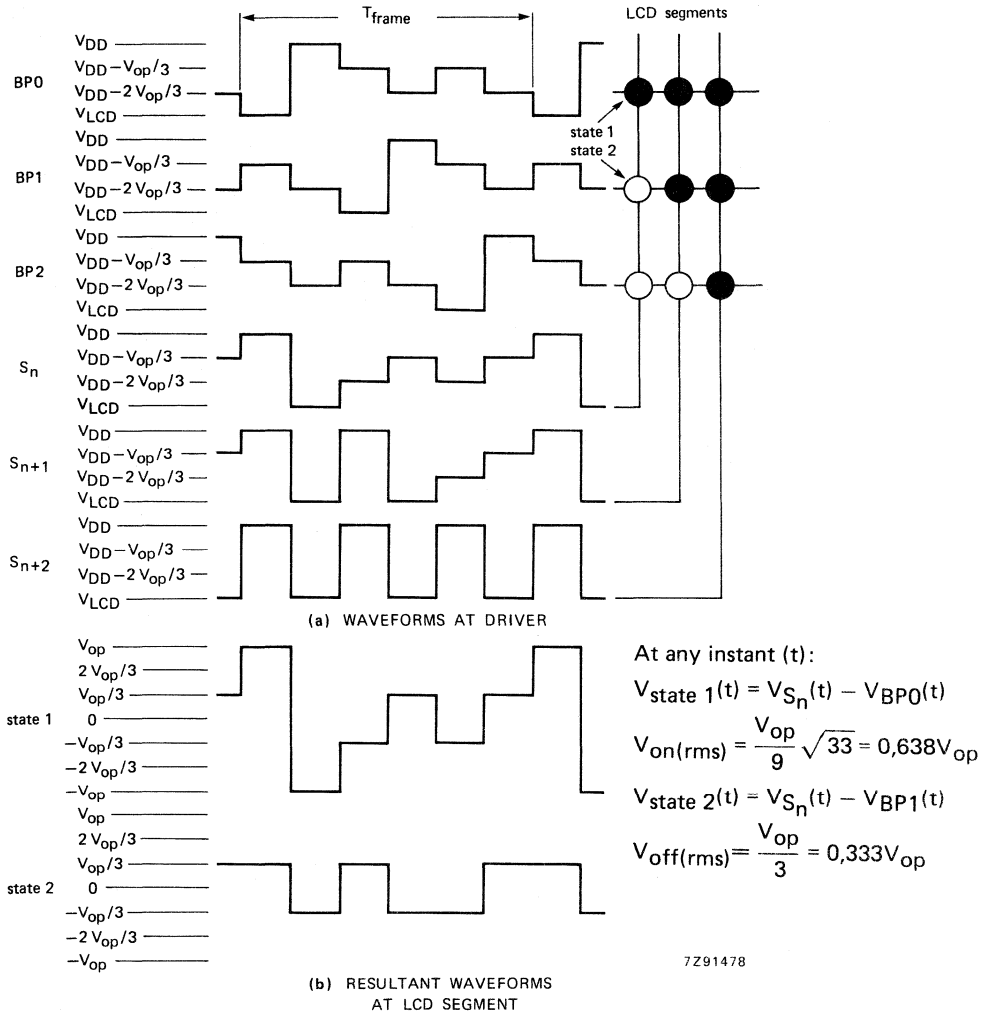


Fig. 7 Waveforms for 1 : 3 multiplex drive mode: $V_{op} = V_{DD} - V_{LCD}$.

LCD drive mode waveforms (continued)

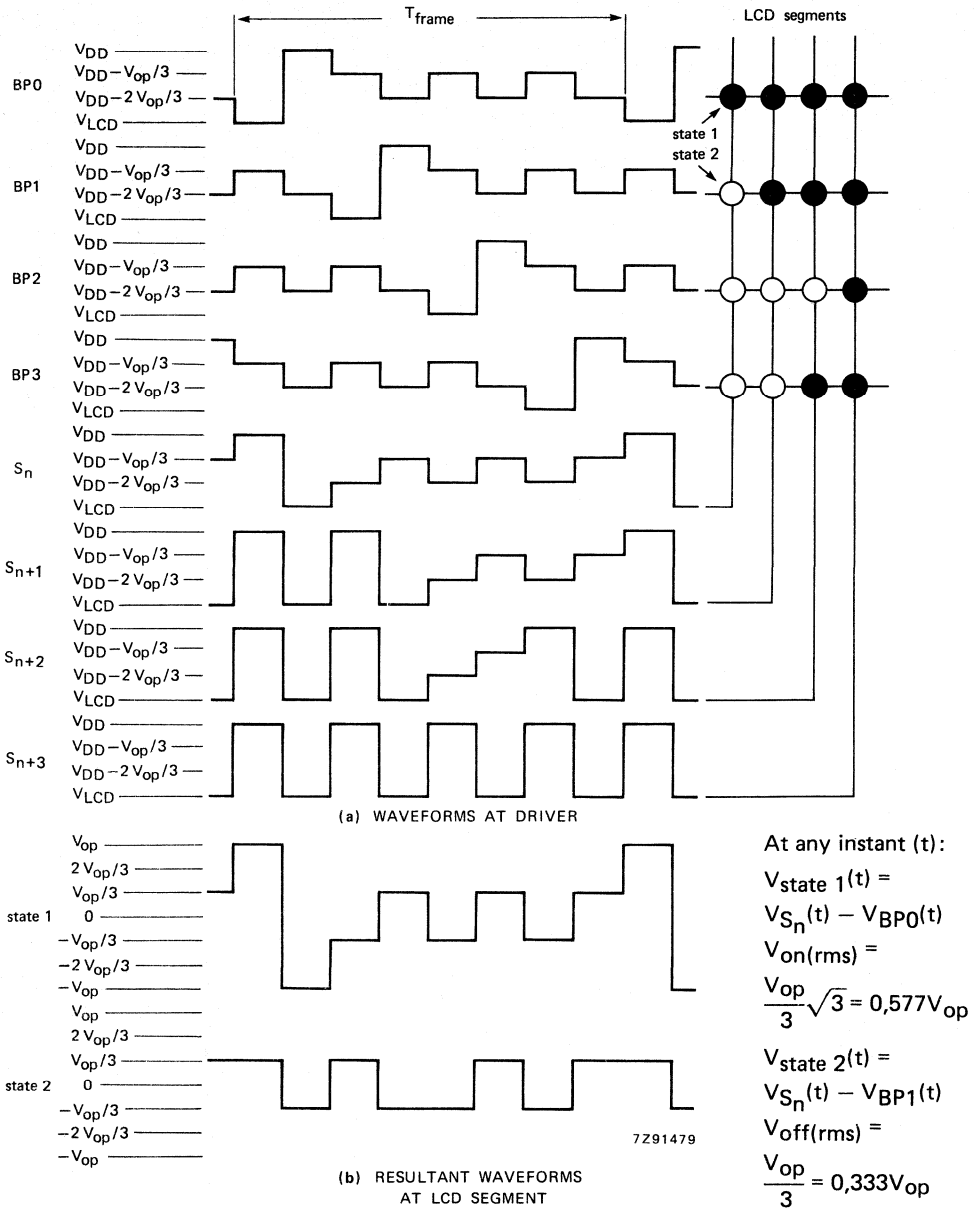


Fig. 8 Waveforms for 1 : 4 multiplex drive mode: $V_{op} = V_{DD} - V_{LCD}$.

Oscillator

The internal logic and the LCD drive signals of the PCF8566 or PCF8576 are timed either by the built-in oscillator or from an external clock.

The clock frequency (f_{CLK}) determines the LCD frame frequency and the maximum rate for data reception from the I^2C bus. To allow I^2C bus transmissions at their maximum data rate of 100 kHz, f_{CLK} should be chosen to be above 125 kHz.

A clock signal must always be supplied to the device; removing the clock may freeze the LCD in a d.c. state.

Internal clock

When the internal oscillator is used, OSC (pin 6) should be tied to V_{SS}. In this case, the output from CLK (pin 4) provides the clock signal for cascaded PCF8566s and PCF8576s in the system.

External clock

The condition for external clock is made by tying OSC (pin 6) to V_{DD}; CLK (pin 4) then becomes the external clock input.

Timing

The timing of the PCF8566 organizes the internal data flow of the device. This includes the transfer of display data from the display RAM to the display segment outputs. In cascaded applications, the synchronization signal $\overline{SYNC}$ maintains the correct timing relationship between the PCF8566s in the system. The timing also generates the LCD frame frequency which it derives as an integer multiple of the clock frequency (Table 3). The frame frequency is set by MODE SET commands when internal clock is used, or by the frequency applied to pin 4 when external clock is used.

Table 3 LCD frame frequencies

PCF8566 mode	f_{frame}	nominal f_{frame} (Hz)
normal mode	$f_{CLK}/2880$	64
power-saving mode	$f_{CLK}/480$	64

The ratio between the clock frequency and the LCD frame frequency depends on the mode in which the device is operating. In the power-saving mode the reduction ratio is six times smaller, this allows the clock frequency to be reduced by a factor of six. The reduced clock frequency results in a significant reduction in power dissipation. The lower clock frequency has the disadvantage of increasing the response time when large amounts of display data are transmitted on the I^2C bus. When a device is unable to 'digest' a display data byte before the next one arrives, it holds the SCL line low until the first display data byte is stored. This slows down the transmission rate of the I^2C bus but no data loss occurs.

Display latch

The display latch holds the display data while the corresponding multiplex signals are generated. There is a one-to-one relationship between the data in the display latch, the LCD segment outputs and one column of the display RAM.

Shift register

The shift register serves to transfer display information from the display RAM to the display latch while previous data are displayed.

Segment outputs

The LCD drive section includes 24 segment outputs S0 to S23 (pins 17 to 40) which should be connected directly to the LCD. The segment output signals are generated in accordance with the multiplexed backplane signals and with the data resident in the display latch. When less than 24 segment outputs are required the unused segment outputs should be left open.

Backplane outputs

The LCD drive section includes four backplane outputs BP0 to BP3 which should be connected directly to the LCD. The backplane output signals are generated in accordance with the selected LCD drive mode. If less than four backplane outputs are required the unused outputs can be left open. In the 1 : 3 multiplex drive mode BP3 carries the same signal as BP1, therefore these two adjacent outputs can be tied together to give enhanced drive capabilities. In the 1 : 2 multiplex drive mode BP0 and BP2, BP1 and BP3 respectively carry the same signals and may also be paired to increase the drive capabilities. In the static drive mode the same signal is carried by all four backplane outputs and they can be connected in parallel for very high drive requirements.

Display RAM

The display RAM is a static 24 x 4-bit RAM which stores LCD data. A logic 1 in the RAM bit-map indicates the 'on' state of the corresponding LCD segment; similarly, a logic 0 indicates the 'off' state. There is a one-to-one correspondence between the RAM addresses and the segment outputs, and between the individual bits of a RAM word and the backplane outputs. The first RAM column corresponds to the 24 segments operated with respect to backplane BP0 (Fig. 9). In multiplexed LCD applications the segment data of the second, third and fourth column of the display RAM are time-multiplexed with BP1, BP2 and BP3 respectively.

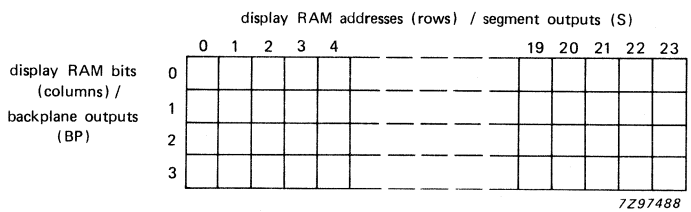


Fig. 9 Display RAM bit-map showing direct relationship between display RAM addresses and segment outputs, and between bits in a RAM word and backplane outputs.

When display data are transmitted to the PCF8566 the display bytes received are stored in the display RAM according to the selected LCD drive mode. To illustrate the filling order, an example of a 7-segment numeric display showing all drive modes is given in Fig. 10; the RAM filling organization depicted applies equally to other LCD types.

With reference to Fig. 10, in the static drive mode the eight transmitted data bits are placed in bit 0 of eight successive display RAM addresses. In the 1 : 2 multiplex drive mode the eight transmitted data bits are placed in bits 0 and 1 of four successive display RAM addresses. In the 1 : 3 multiplex drive mode these bits are placed in bits 0, 1 and 2 of three successive addresses, with bit 2 of the third address left unchanged. This last bit may, if necessary, be controlled by an additional transfer to this address but care should be taken to avoid overriding adjacent data because full bytes are always transmitted. In the 1 : 4 multiplex drive mode the eight transmitted data bits are placed in bits 0, 1, 2 and 3 of two successive display RAM addresses.

Data pointer

The addressing mechanism for the display RAM is realized using the data pointer. This allows the loading of an individual display data byte, or a series of display data bytes, into any location of the display RAM. The sequence commences with the initialization of the data pointer by the LOAD DATA POINTER command. Following this, an arriving data byte is stored starting at the display RAM address indicated by the data pointer thereby observing the filling order shown in Fig. 10. The data pointer is automatically incremented according to the LCD configuration chosen. That is, after each byte is stored, the contents of the data pointer are incremented by eight (static drive mode), by four (1 : 2 multiplex drive mode), by three (1 : 3 multiplex drive mode) or by two (1 : 4 multiplex drive mode).

Subaddress counter

The storage of display data is conditioned by the contents of the subaddress counter. Storage is allowed to take place only when the contents of the subaddress counter agree with the hardware subaddress applied to A0, A1 and A2 (pins 7, 8, and 9). A0, A1 and A2 should be tied to V_{SS} or V_{DD}. The subaddress counter value is defined by the DEVICE SELECT command. If the contents of the subaddress counter and the hardware subaddress do not agree then data storage is inhibited but the data pointer is incremented as if data storage had taken place. The subaddress counter is also incremented when the data pointer overflows.

The storage arrangements described lead to extremely efficient data loading in cascaded applications. When a series of display bytes are being sent to the display RAM, automatic wrap-over to the next PCF8566 occurs when the last RAM address is exceeded. Subaddressing across device boundaries is successful even if the change to the next device in the cascade occurs within a transmitted character.

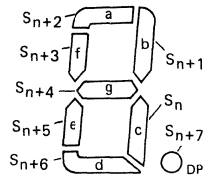
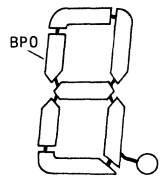
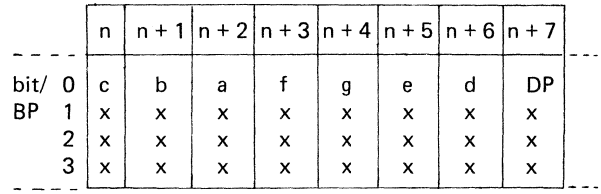
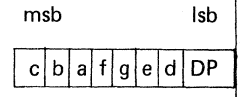
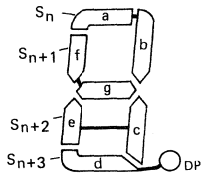
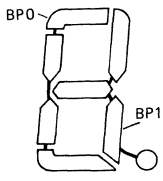
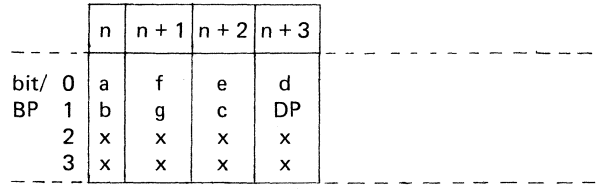
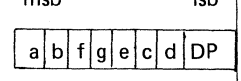
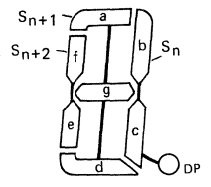
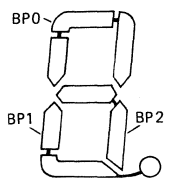
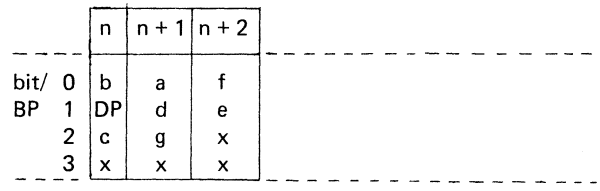
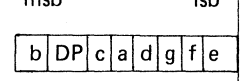
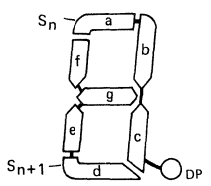
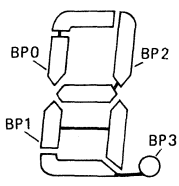
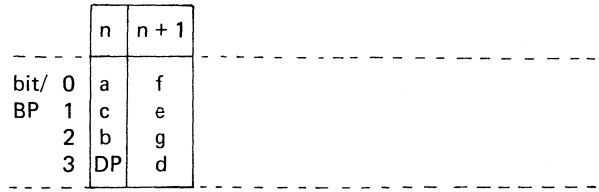
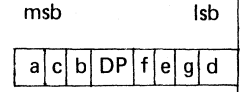
drive mode	LCD segments	LCD backplanes	display RAM filling order	transmitted display byte
static				
1 : 2 multiplex				
1 : 3 multiplex				
1 : 4 multiplex				

Fig. 10 Relationships between LCD layout, drive mode, display RAM filling order and display data transmitted over the I²C bus (x = data bit unchanged).

7Z91469

Output bank selector

This selects one of the four bits per display RAM address for transfer to the display latch. The actual bit chosen depends on the particular LCD drive mode in operation and on the instant in the multiplex sequence. In 1 : 4 multiplex, all RAM addresses of bit 0 are the first to be selected, these are followed by the contents of bit 1, bit 2 and then bit 3. Similarly in 1 : 3 multiplex, bits 0, 1 and 2 are selected sequentially. In 1 : 2 multiplex, bits 0 then 1 are selected and, in the static mode, bit 0 is selected.

The PCF8566 includes a RAM bank switching feature in the static and 1 : 2 multiplex drive modes. In the static drive mode, the BANK SELECT command may request the contents of bit 2 to be selected for display instead of bit 0 contents. In the 1 : 2 drive mode, the contents of bits 2 and 3 may be selected instead of bits 0 and 1. This gives the provision for preparing display information in an alternative bank and to be able to switch to it once it is assembled.

Input bank selector

The input bank selector loads display data into the display RAM according to the selected LCD drive configuration. Display data can be loaded in bit 2 in static drive mode or in bits 2 and 3 in 1 : 2 drive mode by using the BANK SELECT command. The input bank selector functions independently of the output bank selector.

Blinker

The display blinking capabilities of the PCF8566 are very versatile. The whole display can be blinked at frequencies selected by the BLINK command. The blinking frequencies are integer multiples of the clock frequency; the ratios between the clock and blinking frequencies depend on the mode in which the device is operating, as shown in Table 4.

An additional feature is for an arbitrary selection of LCD segments to be blinked. This applies to the static and 1 : 2 LCD drive modes and can be implemented without any communication overheads. By means of the output bank selector, the displayed RAM banks are exchanged with alternate RAM banks at the blinking frequency. This mode can also be specified by the BLINK command.

In the 1 : 3 and 1 : 4 multiplex modes, where no alternate RAM bank is available, groups of LCD segments can be blinked by selectively changing the display RAM data at fixed time intervals.

If the entire display is to be blinked at a frequency other than the nominal blinking frequency, this can be effectively performed by resetting and setting the display enable bit E at the required rate using the MODE SET command.

DEVELOPMENT DATA

Table 4 Blinking frequencies

blinking mode	normal operating mode ratio	power-saving mode ratio	nominal blinking frequency f_{blink} (Hz)
off	–	–	blinking off
2 Hz	$f_{CLK}/92160$	$f_{CLK}/15360$	2
1 Hz	$f_{CLK}/184320$	$f_{CLK}/30720$	1
0,5 Hz	$f_{CLK}/368640$	$f_{CLK}/61440$	0,5

CHARACTERISTICS OF THE I²C BUS

The I²C bus is for 2-way, 2-line communication between different ICs or modules. The two lines are a serial data line (SDA) and a serial clock line (SCL). Both lines must be connected to a positive supply via a pull-up resistor when connected to the output stages of a device. Data transfer may be initiated only when the bus is not busy.

Bit transfer

One data bit is transferred during each clock pulse. The data on the SDA line must remain stable during the HIGH period of the clock pulse as changes in the data line at this time will be interpreted as control signals.

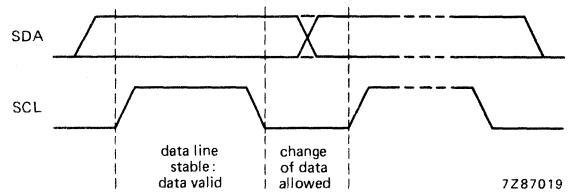


Fig. 11 Bit transfer.

Start and stop conditions

Both data and clock lines remain HIGH when the bus is not busy. A HIGH-to-LOW transition of the data line while the clock is HIGH is defined as the start condition (S). A LOW-to-HIGH transition of the data line while the clock is HIGH is defined as the stop condition (P).

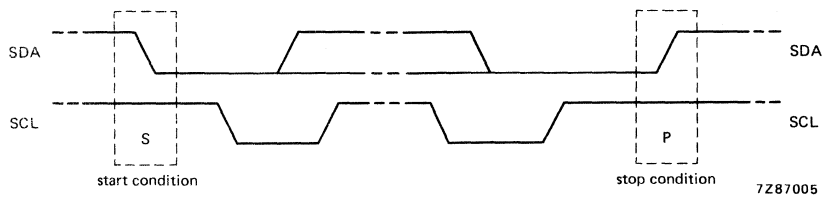


Fig. 12 Definition of start and stop conditions.

System configuration

A device generating a message is a "transmitter", a device receiving a message is a "receiver". The device that controls the message is the "master" and the devices which are controlled by the master are the "slaves".

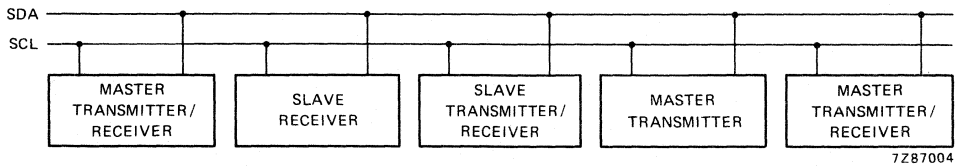


Fig. 13 System configuration.

Acknowledge

The number of data bytes transferred between the start and stop conditions from transmitter to receiver is not limited. Each byte is followed by one acknowledge bit. The acknowledge bit is a HIGH level put on the bus by the transmitter whereas the master generates an extra acknowledge related clock pulse. A slave receiver which is addressed must generate an acknowledge after the reception of each byte. Also a master must generate an acknowledge after the reception of each byte that has been clocked out of the slave transmitter. The device that acknowledges has to pull down the SDA line during the acknowledge clock pulse, so that the SDA line is stable LOW during the HIGH period of the acknowledge related clock pulse, set up and hold times must be taken into account. A master receiver must signal an end of data to the transmitter by *not* generating an acknowledge on the last byte that has been clocked out of the slave. In this event the transmitter must leave the data line HIGH to enable the master to generate a stop condition.

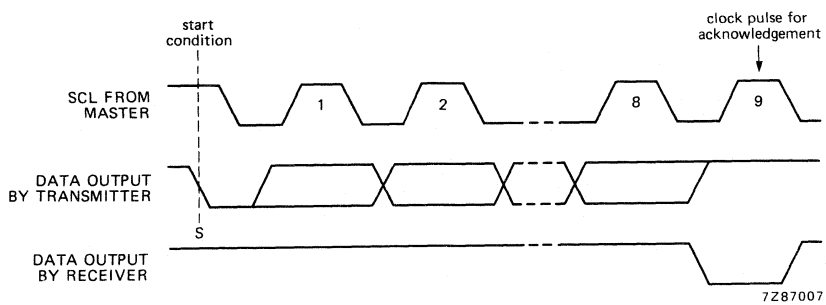


Fig. 14 Acknowledgement on the I^2C bus.

Note

The general characteristics and detailed specification of the I^2C bus are described in a separate data sheet (serial data buses) in handbook: ICs for digital systems in radio, audio and video equipment.

PCF8566 I²C bus controller

The PCF8566 acts as an I²C slave receiver. It does not initiate I²C bus transfers or transmit data to an I²C master receiver. The only data output from the PCF8566 are the acknowledge signals of the selected devices. Device selection depends on the I²C bus slave address, on the transferred command data and on the hardware subaddress.

In single device applications, the hardware subaddress inputs A0, A1 and A2 are normally left open or tied to V_{SS} which defines the hardware subaddress 0. In multiple device applications A0, A1 and A2 are tied to V_{SS} or V_{DD} according to a binary coding scheme such that no two devices with a common I²C slave address have the same hardware subaddress.

In the power-saving mode it is possible that the PCF8566 is not able to keep up with the highest transmission rates when large amounts of display data are transmitted. If this situation occurs, the PCF8566 forces the SCL line LOW until its internal operations are completed. This is known as the 'clock synchronization feature' of the I²C bus and serves to slow down fast transmitters. Data loss does not occur.

Input filters

To enhance noise immunity in electrically adverse environments, RC low-pass filters are provided on the SDA and SCL lines.

I²C bus protocol

Two I²C bus slave addresses (0111110 and 0111111) are reserved for PCF8566. The least-significant bit of the slave address that a PCF8566 will respond to is defined by the level tied at its input SA0 (pin 10). Therefore, two types of PCF8566 can be distinguished on the same I²C bus which allows:

(a) up to 16 PCF8566s on the same I²C bus for very large LCD applications;

(b) the use of two types of LCD multiplex on the same I²C bus.

The I²C bus protocol is shown in Fig. 15. The sequence is initiated with a start condition (S) from the I²C bus master which is followed by one of the two PCF8566 slave addresses available. All PCF8566s with the corresponding SA0 level acknowledge in parallel the slave address but all PCF8566s with the alternative SA0 level ignore the whole I²C bus transfer. After acknowledgement, one or more command bytes (m) follow which define the status of the addressed PCF8566s. The last command byte is tagged with a cleared most-significant bit, the continuation bit C. The command bytes are also acknowledged by all addressed PCF8566s on the bus.

After the last command byte, a series of display data bytes (n) may follow. These display data bytes are stored in the display RAM at the address specified by the data pointer and the subaddress counter. Both data pointer and subaddress counter are automatically updated and the data are directed to the intended PCF8566 device. The acknowledgement after each byte is made only by the (A0, A1, A2) addressed PCF8566. After the last display byte, the I²C bus master issues a stop condition (P).

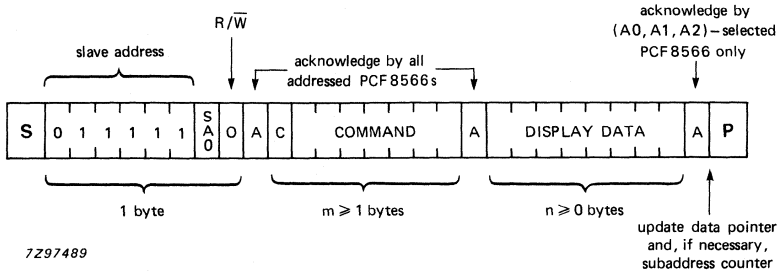


Fig. 15 I²C bus protocol.

Command decoder

The command decoder identifies command bytes that arrive on the I²C bus. All available commands carry a continuation bit C in their most-significant bit position (Fig. 16). When this bit is set, it indicates that the next byte of the transfer to arrive will also represent a command. If the bit is reset, it indicates the last command byte of the transfer. Further bytes will be regarded as display data.

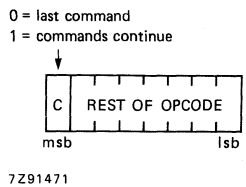


Fig. 16 General format of command byte.

The five commands available to the PCF8566 are defined in Table 5.

Command decoder (continued)

Table 5 Definition of PCF8566 commands

command/opcode	options	description
MODE SET C 1 0 LP E B M1 M0	LCD drive mode: bits M1 M0 static (1 BP): 0 1 1 : 2 MUX (2 BP): 1 0 1 : 3 MUX (3 BP): 1 1 1 : 4 MUX (4 BP): 0 0	Defines LCD drive mode
	LCD bias: bit B 1/3 bias: 0 1/2 bias: 1	Defines LCD bias configuration
	display status: bit E disabled (blank): 0 enabled: 1	Defines display status The possibility to disable the display allows implementation of blinking under external control
	mode: bit LP normal mode: 0 power-saving mode: 1	Defines power dissipation mode
LOAD DATA POINTER C 0 0 P4 P3 P2 P1 P0	bits P4 P3 P2 P1 P0 5-bit binary value of 0 to 23	Five bits of immediate data, bits P4 to P0, are transferred to the data pointer to define one of twenty-four display RAM addresses
DEVICE SELECT C 1 1 0 0 A2 A1 A0	bits A0 A1 A2 3-bit binary value of 0 to 7	Three bits of immediate data, bits A0 to A2, are transferred to the subaddress counter to define one of eight hardware subaddresses

DEVELOPMENT DATA

command/opcode	options	description
BANK SELECT C 1 1 1 1 0 I O	static / 1 : 2 MUX / bit I RAM bit 0 / RAM bits 0, 1 / 0 RAM bit 2 / RAM bits 2, 3 / 1 static / 1 : 2 MUX / bit O RAM bit 0 / RAM bits 0, 1 / 0 RAM bit 2 / RAM bits 2, 3 / 1	Defines input bank selection (storage of arriving display data) Defines output bank selection (retrieval of LCD display data) The BANK SELECT command has no effect in 1 : 3 and 1 : 4 multiplex drive modes
BLINK C 1 1 1 0 A BF1 BF0	blink frequency / bits BF1 / BF0 off / 0 / 0 2 Hz / 0 / 1 1 Hz / 1 / 0 0,5 Hz / 1 / 1 blink mode / bit A normal blinking / 0 alternation blinking / 1	Defines the blinking frequency Selects the blinking mode; normal operation with frequency set by bits BF1, BF0, or blinking by alternation of display RAM banks. Alternation blinking does not apply in 1 : 3 and 1 : 4 multiplex drive modes

Display controller

The display controller executes the commands identified by the command decoder. It contains the status registers of the PCF8566 and coordinates their effects. The controller is also responsible for loading display data into the display RAM as required by the filling order.

Cascaded operation

In large display configurations, up to 16 PCF8566s can be distinguished on the same I^2C bus by using the 3-bit hardware subaddress (A0, A1, A2) and the programmable I^2C slave address (SA0). It is also possible to cascade up to 16 PCF8566s. When cascaded, several PCF8566s are synchronized so that they can share the backplane signals from one of the devices in the cascade. Such an arrangement is cost-effective in large LCD applications since the backplane outputs of only one device need to be through-plated to the backplane electrodes of the display. The other PCF8566s of the cascade contribute additional segment outputs but their backplane outputs are left open (Fig. 17).

The $\overline{SYNC}$ line is provided to maintain the correct synchronization between all cascaded PCF8566s. This synchronization is guaranteed after the power-on reset. The only time that $\overline{SYNC}$ is likely to be needed is if synchronization is accidently lost (e.g. by noise in adverse electrical environments; or by the definition of a multiplex mode when PCF8566s with differing SA0 levels are cascaded). $\overline{SYNC}$ is organized as an input/output pin; the output section being realized as an open-drain driver with an internal pull-up resistor. A PCF8566 asserts the $\overline{SYNC}$ line at the onset of its last active backplane signal and monitors the $\overline{SYNC}$ line at all other times. Should synchronization in the cascade be lost, it will be restored by the first PCF8566 to assert $\overline{SYNC}$. The timing relationships between the backplane waveforms and the $\overline{SYNC}$ signal for the various drive modes of the PCF8576 are shown in Fig. 18. The waveforms are identical with the parent device PCF8576. Casadability between PCF8566s and PCF8576s is possible, giving cost effective LCD applications.

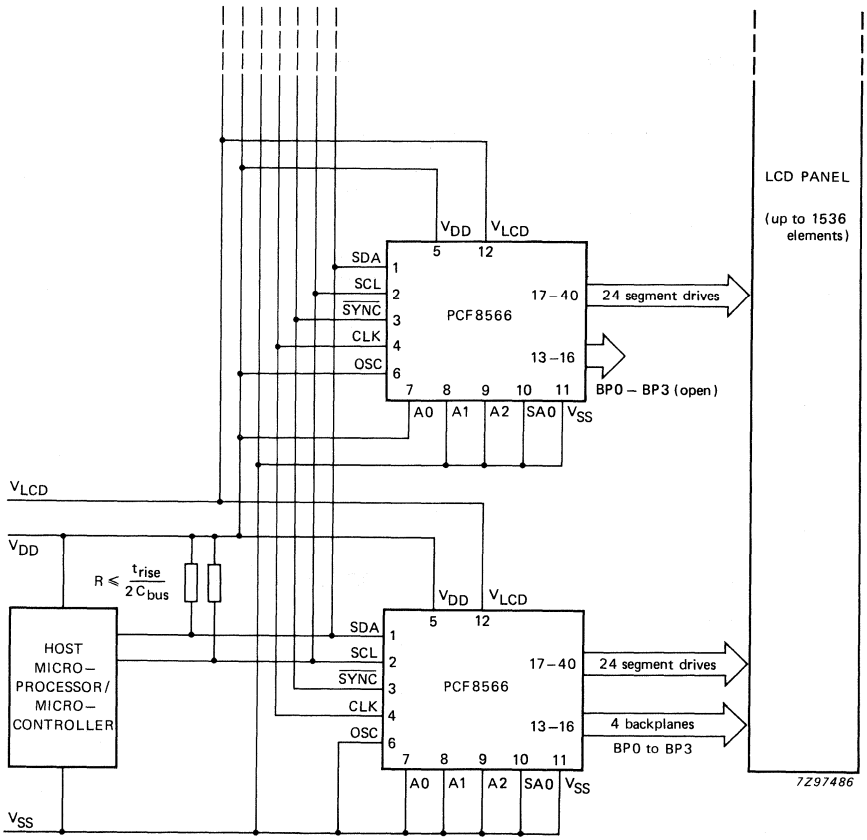


Fig. 17 Cascaded PCF8566 configuration.

DEVELOPMENT DATA

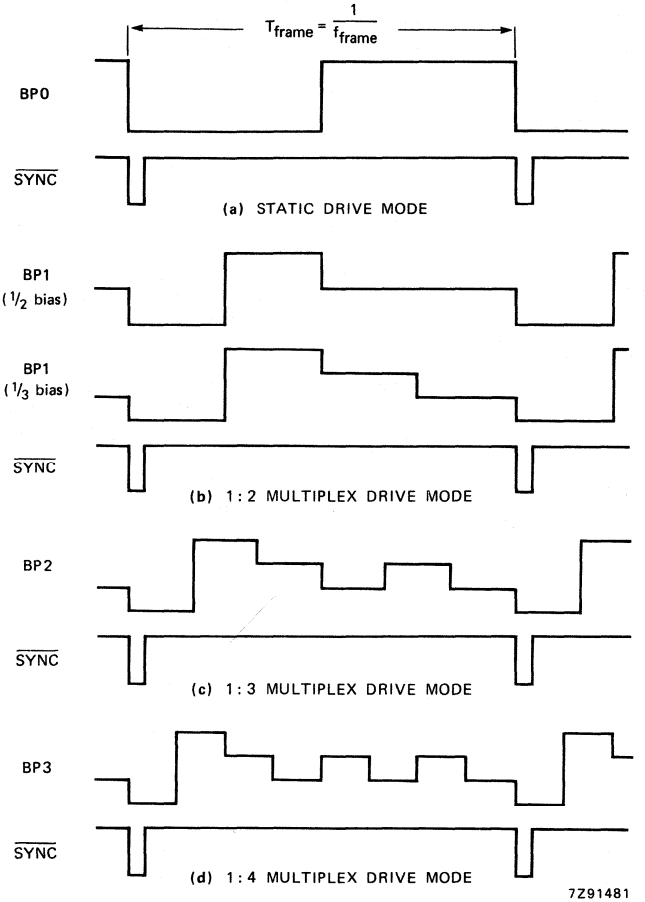


Fig. 18 Synchronization of the cascade for the various PCF8566 drive modes.

For single plane wiring of PCF8566s, see section "APPLICATION INFORMATION".

RATINGS

Limiting values in accordance with the Absolute Maximum System (IEC 134)

Supply voltage range; see note	V_{DD}	$-0,5$ to $+7$	V
LCD supply voltage range	V_{LCD}	$V_{DD} -7$ to V_{DD}	V
Input voltage range (SCL; SDA; A0 to A2; OSC; CLK; $\overline{SYNC}$; SA0)	V_I	$V_{SS} -0,5$ to $V_{DD} + 0,5$	V
Output voltage range (S0 to S23; BP0 to BP3)	V_O	$V_{LCD} -0,5$ to $V_{DD} + 0,5$	V
D.C. input current	$\pm I_I$	max. 20	mA
D.C. output current	$\pm I_O$	max. 25	mA
V_{DD}, V_{SS} or V_{LCD} current	$\pm I_{DD}$, $\pm I_{SS}$, $\pm I_{LCD}$	max. 50	mA
Power dissipation per package	P_{tot}	max. 400	mW
Power dissipation per output	P_O	max. 100	mW
Storage temperature range	T_{stg}	-65 to $+150$	°C

Note

Inputs and outputs are protected against electrostatic discharges in normal handling. However, to be totally safe, it is advised to take handling precautions appropriate to handling MOS devices (see 'Handling MOS devices').

D.C. CHARACTERISTICS

$V_{SS} = 0$ V; $V_{DD} = 3$ to 6 V; $V_{LCD} = V_{DD} -3$ to $V_{DD} -6$ V;
$T_{amb} = -40$ to $+85$ °C; unless otherwise specified

parameter	symbol	min.	typ.	max.	unit
Operating supply voltage	V_{DD}	3	–	6	V
LCD supply voltage	V_{LCD}	$V_{DD} -6$	–	$V_{DD} -3$	V
Operating supply current (normal mode) at f_{CLK} = 200 kHz (note 1)	I_{DD}	–	–	180	μA
Power-saving mode supply current at V_{DD} = 3,5 V; V_{LCD} = 0 V; f_{CLK} = 35 kHz; A0, A1 and A2 tied to V_{SS} (note 1)	I_{LP}	–	–	60	μA

DEVELOPMENT DATA

parameter	symbol	min.	typ.	max.	unit
Logic					
Input voltage LOW	V_{IL}	V_{SS}	–	0,3 V_{DD}	V
Input voltage HIGH	V_{IH}	0,7 V_{DD}	–	V_{DD}	V
Output voltage LOW at I_O = 0 mA	V_{OL}	–	–	0,05	V
Output voltage HIGH at I_O = 0 mA	V_{OH}	V_{DD} −0,05	–	–	V
Output current LOW (CLK, $\overline{SYNC}$) at V_{OL} = 1,0 V; V_{DD} = 5 V	I_{OL1}	1	–	–	mA
Output current HIGH (CLK) at V_{OH} = 4,0 V; V_{DD} = 5 V	I_{OH}	–	–	−1	mA
Output current LOW (SDA; SCL) at V_{OL} = 0,4 V; V_{DD} = 5 V	I_{OL2}	3	–	–	mA
Leakage current (SAO, CLK, OSC, A0, A1, A2, SCL, SDA) at $V_I = V_{SS}$ or V_{DD}	$\pm I_L$	–	–	1	μA
Pull-up resistor ($\overline{SYNC}$)	R_{SYNC}	15	25	60	kΩ
Power-on reset level (note 2)	V_{REF}	–	1,3	1,8	V
Tolerable spike width on bus	t_{sw}	–	–	100	ns
Input capacitance (note 3)	C_I	–	–	7	pF
LCD outputs					
D.C. voltage component (BP0 to BP3) at C_{BP} = 35 nF	$\pm V_{BP}$	–	20	–	mV
D.C. voltage component (S0 to S23) at C_S = 5 nF	$\pm V_S$	–	20	–	mV
Output impedance (BP0 to BP3) at $V_{LCD} = V_{DD}$ −5 V (note 4)	R_{BP}	–	–	5	kΩ
Output impedance (S0 to S23) at $V_{LCD} = V_{DD}$ −5 V (note 4)	R_S	–	–	7,0	kΩ

A.C. CHARACTERISTICS (note 5)

V_{SS} = 0 V; V_{DD} = 3 to 6 V; V_{LCD} = V_{DD} −3 to V_{DD} −6 V;

T_{amb} = −40 to +85 °C; unless otherwise specified

parameter	symbol	min.	typ.	max.	unit
Oscillator frequency (normal mode) (note 6)	f_{CLK}	125	200	315	kHz
Oscillator frequency (power-saving mode) at V_{DD} = 3,5 V	f_{CLKLP}	21	31	48	kHz
CLK HIGH time	t_{CLKH}	1	–	–	μs
CLK LOW time	t_{CLKL}	1	–	–	μs
$\overline{SYNC}$ propagation delay	t_{PSYNC}	–	–	400	ns
$\overline{SYNC}$ LOW time	t_{SYNCL}	1	–	–	μs
Driver delays with test loads at V_{LCD} = V_{DD} −5 V	t_{PLCD}	–	–	30	μs
I²C bus					
Bus free time	t_{BUF}	4,7	–	–	μs
Start condition hold time	$t_{HD;STA}$	4	–	–	μs
SCL LOW time	t_{LOW}	4,7	–	–	μs
SCL HIGH time	t_{HIGH}	4	–	–	μs
Start condition set-up time (repeated start code only)	$t_{SU;STA}$	4,7	–	–	μs
Data hold time	$t_{HD;DAT}$	0	–	–	μs
Data set-up time	$t_{SU;DAT}$	250	–	–	ns
Rise time	t_R	–	–	1	μs
Fall time	t_F	–	–	300	ns
Stop condition set-up time	$t_{SU;STO}$	4,7	–	–	μs

Notes to characteristics

1. Outputs open; inputs at V_{SS} or V_{DD}; external clock with 50% duty factor; I²C bus inactive.
2. Resets all logic when $V_{DD} < V_{REF}$.
3. Periodically sampled, not 100% tested.
4. Outputs measured one at a time.
5. All timing values referred to V_{IH} and V_{IL} levels with an input voltage swing of V_{SS} to V_{DD}.
6. At $f_{CLK} < 125$ kHz, I²C bus maximum transmission speed is derated.

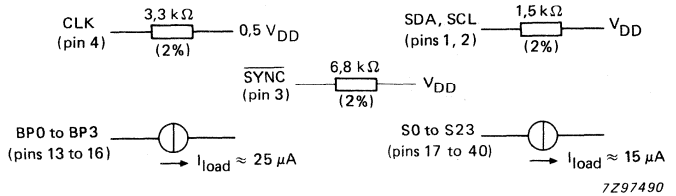


Fig. 19 Test loads.

DEVELOPMENT DATA

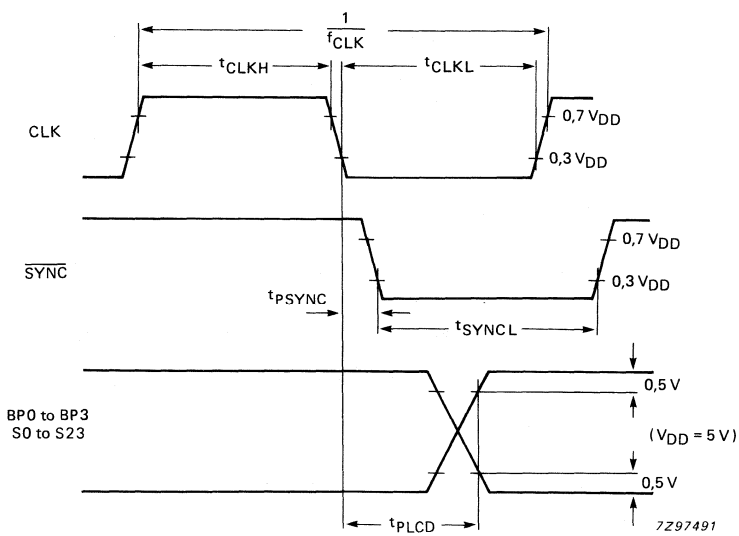


Fig. 20 Driver timing waveforms.

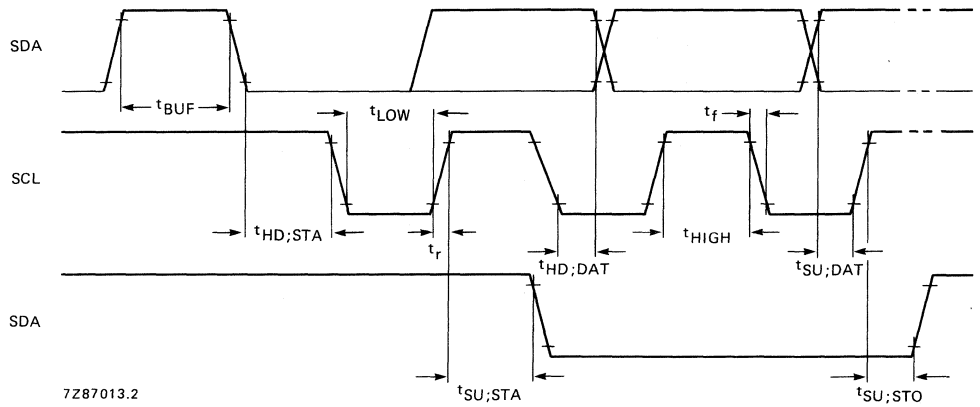
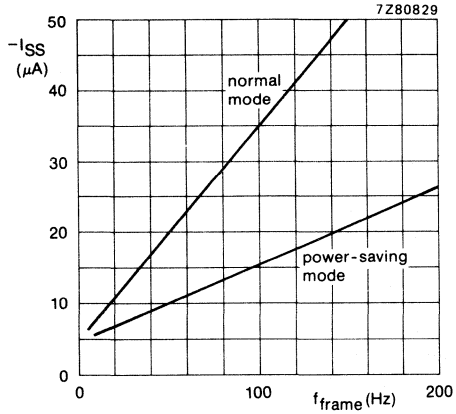
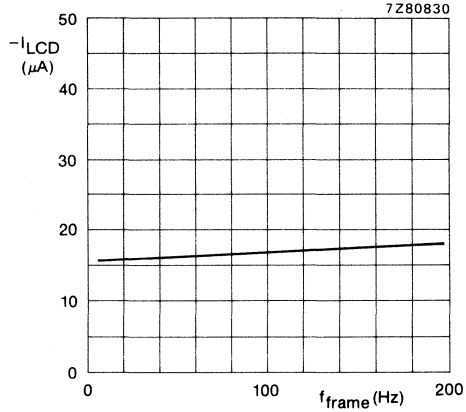


Fig. 21 I^2C bus timing waveforms.

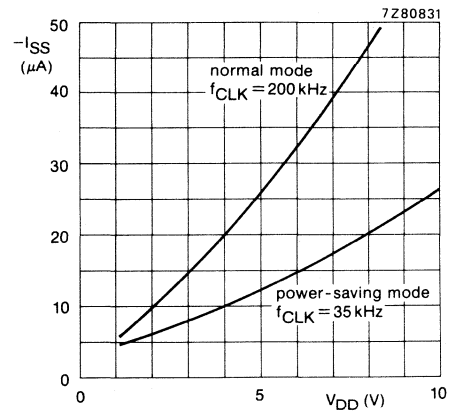
DEVELOPMENT DATA



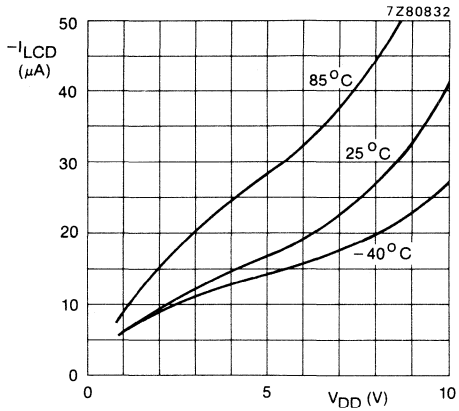
(a) V_{DD} = 5 V; V_{LCD} = 0 V; T_{amb} = 25 °C.



(b) V_{DD} = 5 V; V_{LCD} = 0 V; T_{amb} = 25 °C.

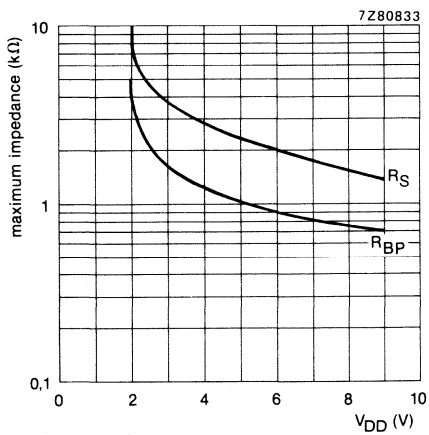


(c) V_{LCD} = 0 V; external clock; T_{amb} = −40 to + 85 °C.

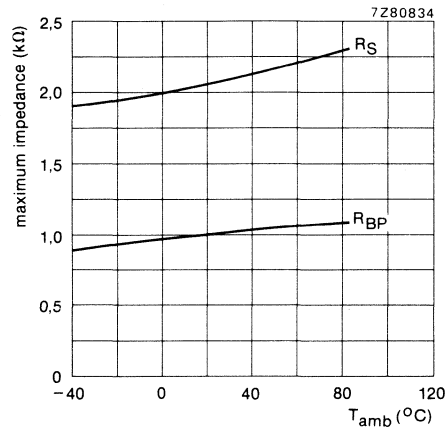


(d) V_{LCD} = 0 V; external clock; f_{CLK} = nominal frequency.

Fig. 22 Typical supply current characteristics.



(a) V_{LCD} = 0 V; T_{amb} = 25 °C.



(b) V_{DD} = 5 V; V_{LCD} = 0 V.

Fig. 23 Typical characteristics of LCD outputs.

APPLICATION INFORMATION

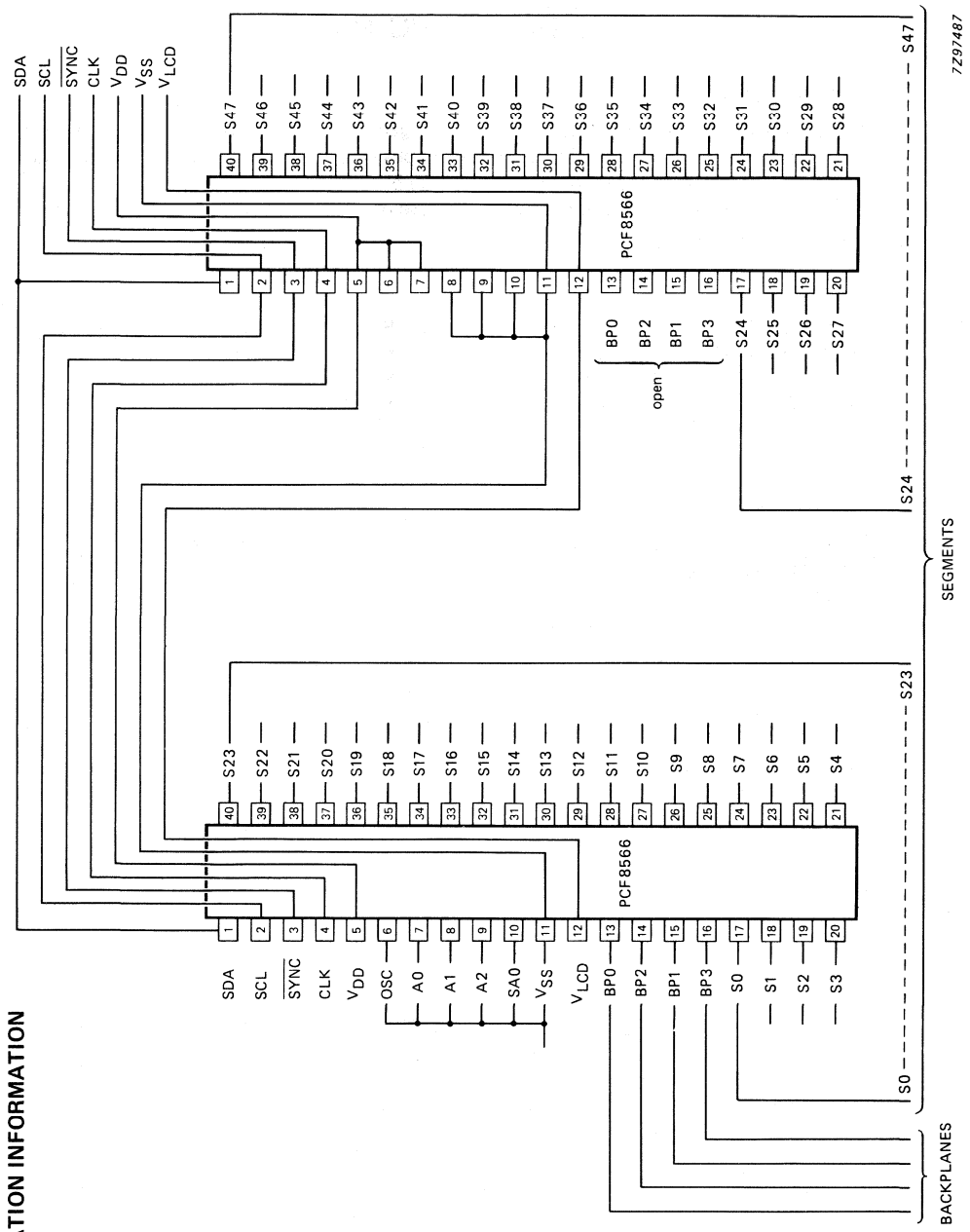


Fig. 24 Single plane wiring of packaged PCF8566s.

256 x 8-BIT STATIC RAM

GENERAL DESCRIPTION

The PCF8570 is a low power 2048-bit static CMOS RAM organized as 256 words by 8-bits. Addresses and data are transferred serially via a two-line bidirectional bus (I^2C). The built-in word address register is incremented automatically after each written or read data byte. Three address pins A0, A1, A2 are used for programming the hardware address, allowing the use of up to eight devices connected to the bus without additional hardware.

Features

- Operating supply voltage 2,5 V to 6 V
- Low data retention voltage min. 1,0 V
- Low standby current max. 15 μA
- Power saving mode typ. 50 nA
- Serial input/output bus (I^2C)
- Address by 3 hardware address pins
- Automatic word address incrementing
- 8-lead DIL package

Applications

- Telephony
- Radio and television
- Video cassette recorder
- General purpose

RAM expansion for stored numbers in repertory dialling (e.g. PCD3343 applications)
channel presets

RAM expansion for the microcontroller families MAB8400 and PCF84C00

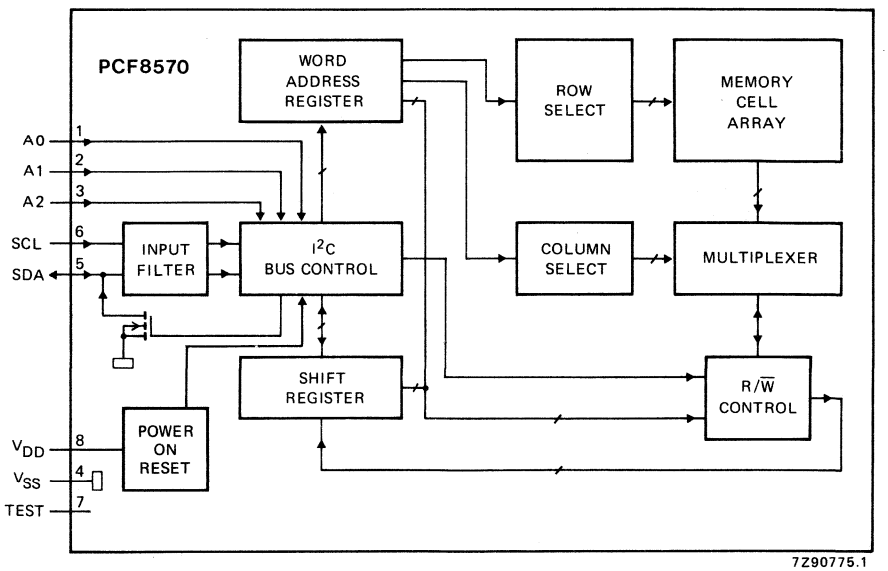


Fig. 1 Block diagram.

PACKAGE OUTLINES

PCF8570P: 8-lead DIL; plastic (SOT-97AE).
PCF8570T: 8-lead mini-pack plastic (SO-8L; SOT-176).

PINNING

1 to 3	A0 to A2	address inputs
4	V_{SS}	negative supply
5	SDA	serial data line (I²C bus)
6	SCL	serial clock line (I²C bus)
7	TEST	test input for test speed-up; must be connected to V_{SS} when not in use (power saving mode, see Figs 14 and 15)
8	V_{DD}	positive supply

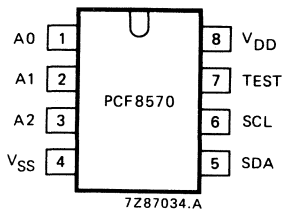


Fig. 2 Pinning diagram.

RATINGS

Limiting values in accordance with the Absolute Maximum System (IEC 134)

Supply voltage range (pin 8)	V_{DD}		−0,8 to + 8,0	V
Voltage range on any input	V_I		−0,8 to V_{DD} + 0,8	V
D.C. input current (any input)	$\pm I_I$	max.	10	mA
D.C. output current (any output)	$\pm I_O$	max.	10	mA
Supply current (pin 4 or pin 8)	$\pm I_{DD}$; I_{SS}	max.	50	mA
Power dissipation per package	P_{tot}	max.	300	mW
Power dissipation per output	P	max.	50	mW
Storage temperature range	T_{stg}		−65 to + 150	°C
Operating ambient temperature range	T_{amb}		−40 to + 85	°C

CHARACTERISTICS

V_{DD} = 2,5 to 6 V; V_{SS} = 0 V; T_{amb} = −40 to + 85 °C unless otherwise specified

parameter	symbol	min.	typ.	max.	unit
Supply					
Supply voltage	V_{DD}	2,5	–	6	V
Supply current at $V_I = V_{SS}$ or V_{DD}					
operating at f_{SCL} = 100 kHz	I_{DD}	–	–	200	μA
standby at f_{SCL} = 0 Hz	I_{DDO}	–	–	15	μA
standby at T_{amb} = −25 to + 70 °C	I_{DDO}	–	–	5	μA
Power-on reset voltage level*	V_{POR}	1,5	1,9	2,3	V
Inputs; input/output SDA					
Input voltage LOW**	V_{IL}	−0,8	–	0,3 x V_{DD}	V
Input voltage HIGH**	V_{IH}	0,7 x V_{DD}	–	V_{DD} + 0,8	V
Output current LOW at V_{OL} = 0,4 V	I_{OL}	3	–	–	mA
Output leakage current HIGH at $V_{OH} = V_{DD}$	I_{OH}	–	–	250	nA
Input leakage current at $V_I = V_{DD}$ or V_{SS}	$\pm I_I$	–	–	250	nA
Clock frequency (Fig. 7)	f_{SCL}	0	–	100	kHz
Input capacitance (SCL, SDA) at $V_I = V_{SS}$	C_I	–	–	7	pF
Tolerable spike width on bus	t_{SW}	–	–	100	ns
LOW V_{DD} data retention					
Supply voltage for data retention	V_{DDR}	1	–	6	V
Supply current at V_{DDR} = 1 V	I_{DDR}	–	–	5	μA
Supply current at V_{DDR} = 1 V; T_{amb} = −25 to + 70 °C	I_{DDR}	–	–	2	μA
Power saving mode (Figs 14 and 15)					
Supply current at T_{amb} = 25 °C; TEST = V_{DDR}	I_{DDR}	–	50	400	nA

* The power-on reset circuit resets the I²C bus logic when $V_{DD} < V_{POR}$.

** If the input voltages are a diode voltage above or below the supply voltage V_{DD} or V_{SS} an input current will flow; this current must not exceed ± 0,5 mA.

CHARACTERISTICS OF THE I²C BUS

The I²C bus is for 2-way, 2-line communication between different ICs or modules. The two lines are a serial data line (SDA) and a serial clock line (SCL). Both lines must be connected to a positive supply via a pull-up resistor when connected to the output stages of a device. Data transfer may be initiated only when the bus is not busy.

Bit transfer

One data bit is transferred during each clock pulse. The data on the SDA line must remain stable during the HIGH period of the clock pulse as changes in the data line at this time will be interpreted as control signals.

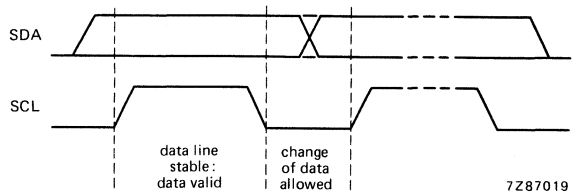


Fig. 3 Bit transfer.

Start and stop conditions

Both data and clock lines remain HIGH when the bus is not busy. A HIGH-to-LOW transition of the data line, while the clock is HIGH is defined as the start condition (S). A LOW-to-HIGH transition of the data line while the clock is HIGH is defined as the stop condition (P).

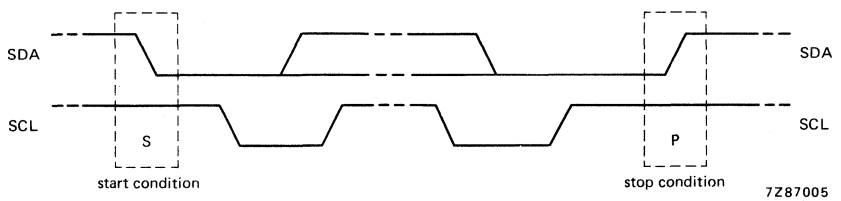


Fig. 4 Definition of start and stop conditions.

System configuration

A device generating a message is a "transmitter", a device receiving a message is the "receiver". The device that controls the message is the "master" and the devices which are controlled by the master are the "slaves".

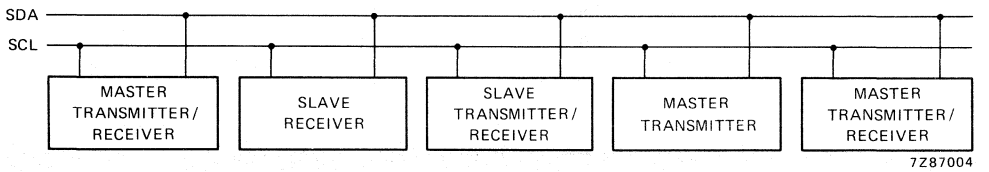


Fig. 5 System configuration.

Acknowledge

The number of data bytes transferred between the start and stop conditions from transmitter to receiver is not limited. Each byte of eight bits is followed by one acknowledge bit. The acknowledge bit is a HIGH level put on the bus by the transmitter whereas the master generates an extra acknowledge related clock pulse. A slave receiver which is addressed must generate an acknowledge after the reception of each byte. Also a master must generate an acknowledge after the reception of each byte that has been clocked out of the slave transmitter. The device that acknowledges has to pull down the SDA line during the acknowledge clock pulse, so that the SDA line is stable LOW during the HIGH period of the acknowledge related clock pulse, set-up and hold times must be taken into account. A master receiver must signal an end of data to the transmitter by *not* generating an acknowledge on the last byte that has been clocked out of the slave. In this event the transmitter must leave the data line HIGH to enable the master to generate a stop condition.

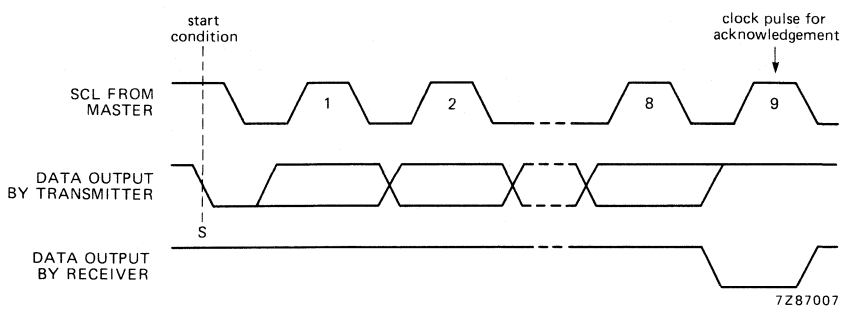


Fig. 6 Acknowledgement on the I^2C bus.

Timing specifications

Within the I^2C bus specifications a high-speed mode and a low-speed mode are defined. The device operates in both modes and the timing requirements are as follows:

High-speed mode

Masters generate a bus clock with a maximum frequency of 100 kHz. Detailed timing is shown in Fig. 7.

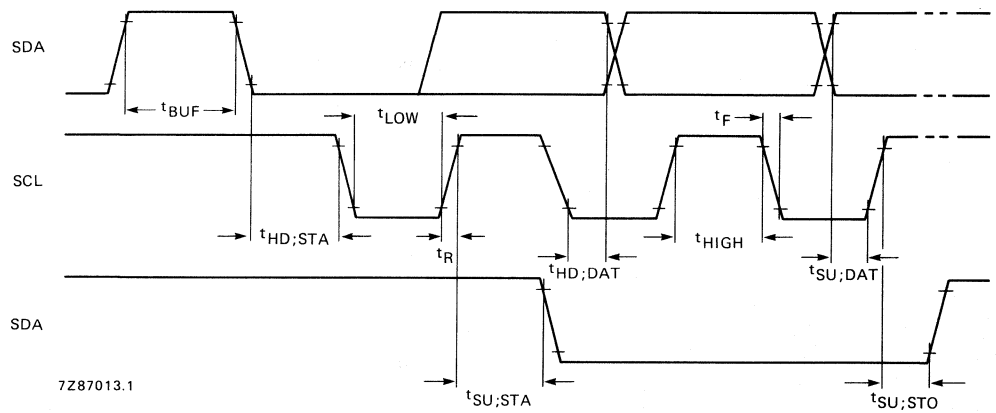


Fig. 7 Timing of the high-speed mode.

Where:

t_{BUF}	$t \geqslant t_{LOWmin}$	The minimum time the bus must be free before a new transmission can start
$t_{HD;STA}$	$t \geqslant t_{HIGHmin}$	Start condition hold time
t_{LOWmin}	4,7 μs	Clock LOW period
$t_{HIGHmin}$	4 μs	Clock HIGH period
$t_{SU;STA}$	$t \geqslant t_{LOWmin}$	Start condition set-up time, only valid for repeated start code
$t_{HD;DAT}$	$t \geqslant 0$ μs	Data hold time
$t_{SU;DAT}$	$t \geqslant 250$ ns	Data set-up time
t_R	$t \leqslant 1$ μs	Rise time of both the SDA and SCL line
t_F	$t \leqslant 300$ ns	Fall time of both the SDA and SCL line
$t_{SU;STO}$	$t \geqslant t_{LOWmin}$	Stop condition set-up time

Note

All the timing values refer to V_{IH} and V_{IL} levels with a voltage swing of V_{SS} to V_{DD}.

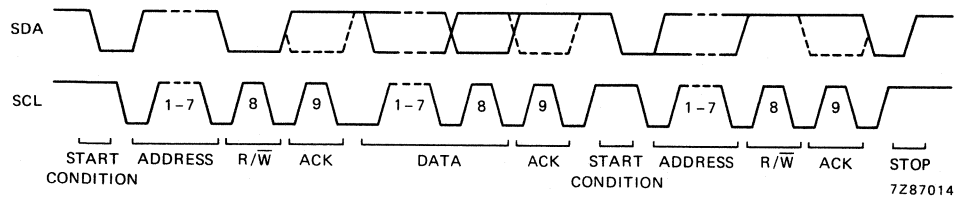


Fig. 8 Complete data transfer in the high-speed mode.

Where:

Clock t_{LOWmin}	4,7 μs
$t_{HIGHmin}$	4 μs

The dashed line is the acknowledgement of the receiver

Mark-to-space ratio	1 : 1 (LOW-to-HIGH)
Max. number of bytes	unrestricted
Premature termination of transfer	allowed by generation of STOP condition
Acknowledge clock bit	must be provided by the master

Low-speed mode

Masters generate a bus clock with a maximum frequency of 2 kHz; a minimum LOW period of 105 μs and a minimum HIGH period of 365 μs. The mark-to-space ratio is 1 : 3 LOW-to-HIGH. Detailed timing is shown in Fig. 9.

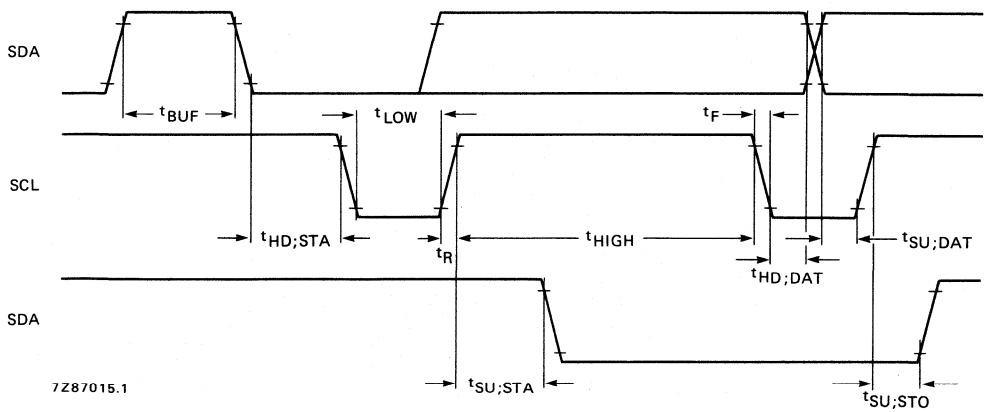


Fig. 9 Timing of the low-speed mode.

Timing specifications (continued)

Where:

t_{BUF}	$t \geqslant 105\ \mu s$ (t_{LOWmin})
$t_{HD;STA}$	$t \geqslant 365\ \mu s$ ($t_{HIGHmin}$)
t_{LOW}	$130\ \mu s \pm 25\ \mu s$
t_{HIGH}	$390\ \mu s \pm 25\ \mu s$
$t_{SU;STA}$	$130\ \mu s \pm 25\ \mu s$ *
$t_{HD;DAT}$	$t \geqslant 0\ \mu s$
$t_{SU;DAT}$	$t \geqslant 250$ ns
t_R	$t \leqslant 1\ \mu s$
t_F	$t \leqslant 300$ ns
$t_{SU;STO}$	$130\ \mu s \pm 25\ \mu s$

Note

All the timing values refer to V_{IH} and V_{IL} levels with a voltage swing of V_{SS} to V_{DD}. For definitions see high-speed mode.

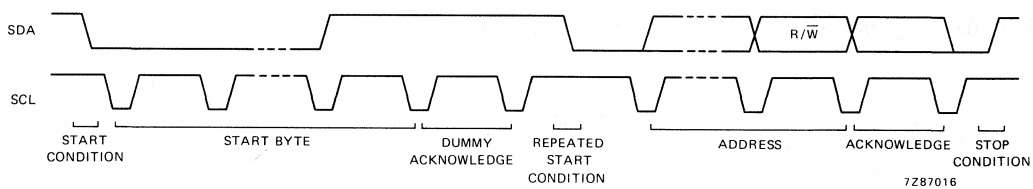


Fig. 10 Complete data transfer in the low-speed mode.

Where:

Clock t_{LOWmin}	$130\ \mu s \pm 25\ \mu s$
$t_{HIGHmin}$	$390\ \mu s \pm 25\ \mu s$
Mark-to-space ratio	1 : 3 (LOW-to-HIGH)
Start byte	0000 0001
Max. number of bytes	6
Premature termination of transfer	not allowed
Acknowledge clock bit	must be provided by master

Note

The general characteristics and detailed specification of the I^2C bus are described in a separate data sheet (serial data buses) in handbook "ICs for digital systems in radio, audio and video equipment".

* Only valid for repeated start code.

Bus protocol

Before any data is transmitted on the I²C bus, the device which should respond is addressed first. The addressing is always done with the first byte transmitted after the start procedure. The I²C bus configuration for different PCF8570 READ and WRITE cycles is shown in Fig. 11.

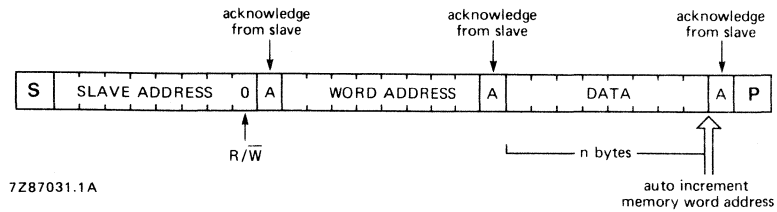


Fig. 11(a) Master transmits to slave receiver (WRITE mode).

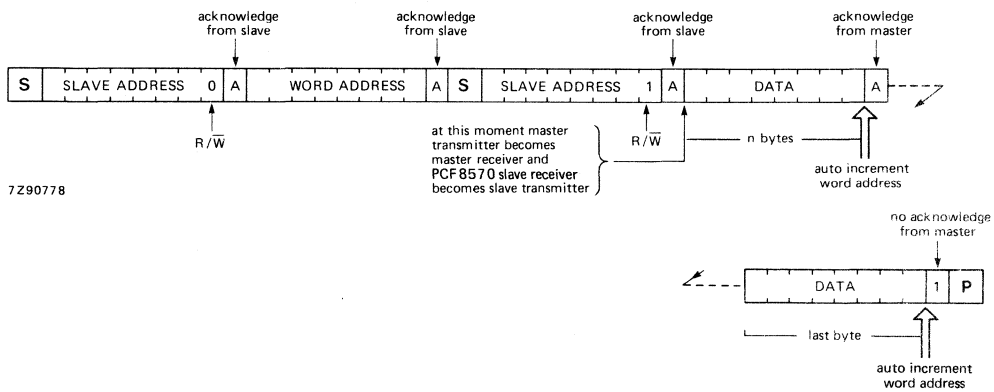


Fig. 11(b) Master reads after setting word address (WRITE word address; READ data).

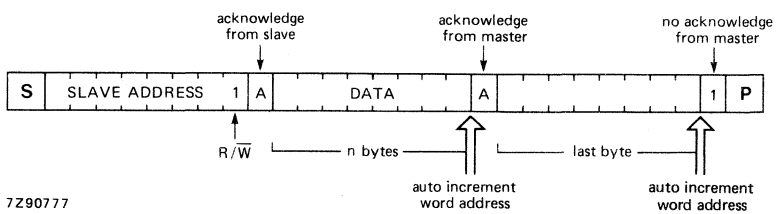


Fig. 11(c) Master reads slave immediately after first byte (READ mode).

APPLICATION INFORMATION

The PCF8570 slave address has a fixed combination 1010 as group 1, while group 2 is fully programmable (see Fig. 12).

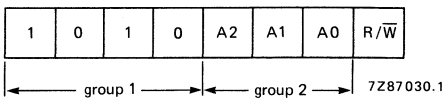


Fig. 12 PCF8570 address.

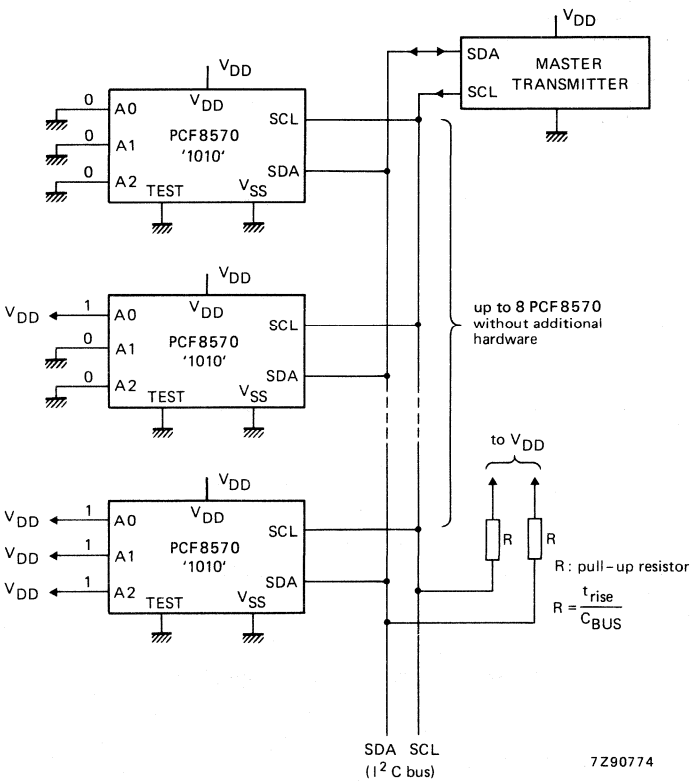


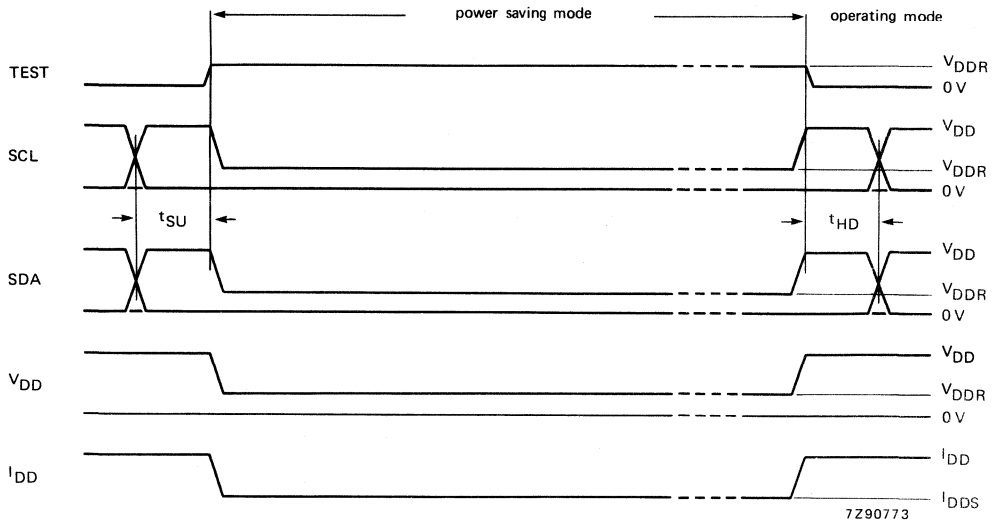
Fig. 13 PCF8570 application diagram.

Note

A0, A1, and A2 inputs must be connected to V_{DD} or V_{SS} but not left open.

POWER SAVING MODE

With the condition TEST = V_{DDR}, the PCF8570 goes into the power saving mode and the I^2C bus logic is reset.



Where:

$t_{SU} \geqslant 4\ \mu s$

$t_{HD} \geqslant 4\ \mu s$

Fig. 14 Timing for power saving mode.

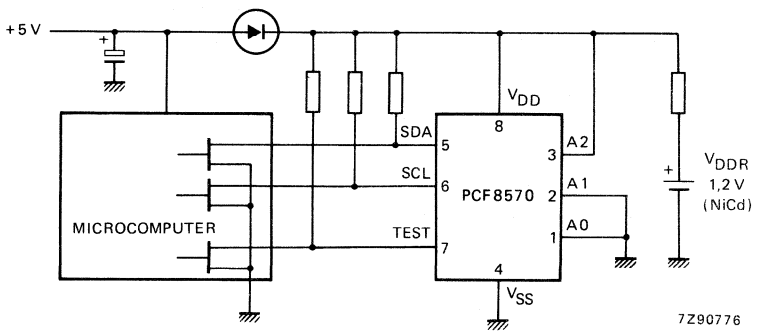
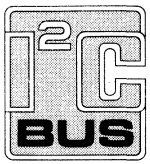


Fig. 15 Application example for power saving mode.

Note to Fig. 15

1. In the operating mode, TEST = 0.
2. In the power saving mode, TEST = V_{DDR}.



Purchase of Philips' I²C components conveys a license under the Philips' I²C patent to use the components in the I²C-system provided the system conforms to the I²C specification defined by Philips.

128 x 8-BIT STATIC RAM

GENERAL DESCRIPTION

The PCF8571 is a low power 1024-bit static CMOS RAM organized as 128 words by 8-bits. Addresses and data are transferred serially via a two-line bidirectional bus (I^2C). The built-in word address register is incremented automatically after each written or read data byte. Three address pins A0, A1, A2 are used for programming the hardware address, allowing the use of up to eight devices connected to the bus without additional hardware.

Features

- Operating supply voltage 2,5 V to 6 V
- Low data retention voltage min. 1,0 V
- Low standby current max. 5 μA
- Power saving mode typ. 50 nA
- Serial input/output bus (I^2C)
- Address by 3 hardware address pins
- Automatic word address incrementing
- 8-lead DIL package

Applications

- Telephony
- Radio and television
- Video cassette recorder
- General purpose

RAM expansion for stored numbers in repertory dialling (e.g. PCD3340 applications)
channel presets

RAM expansion for the microcomputer families MAB8400 and PCF84C00

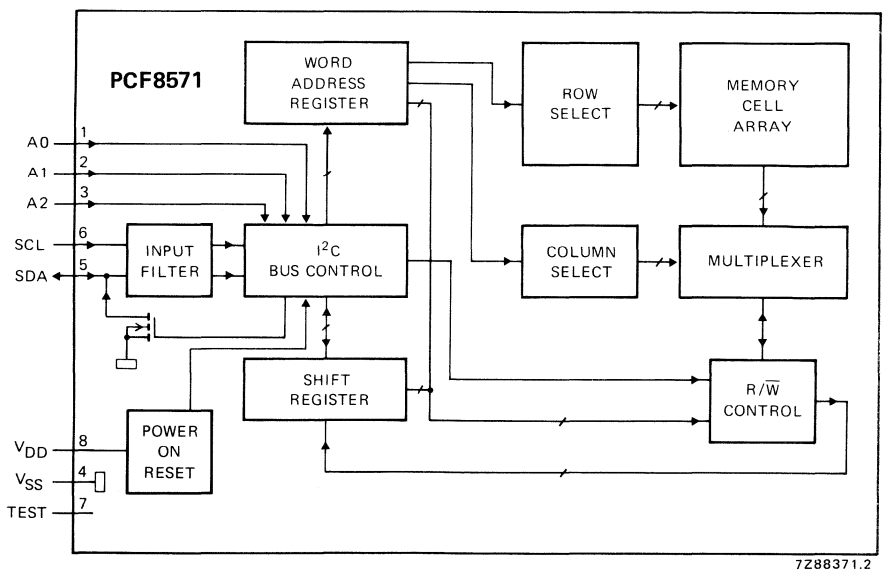


Fig. 1 Block diagram.

PACKAGE OUTLINES

PCF8571P : 8-lead DIL; plastic (SOT-97AE).
PCF8571D : 8-lead DIL; ceramic (cerdip) (SOT-151A).
PCF8571T : 8-lead mini-pack (SO-8L; SOT-176).

PINNING

1 to 3	A0 to A2	address inputs	
4	V_{SS}	negative supply	
5	SDA	serial data line	I^2C bus
6	SCL	serial clock line	I^2C bus
7	TEST	test input for test speed-up; must be connected to V_{SS} when not in use (power saving mode, see Fig. 14 and 15)	
8	V_{DD}	positive supply	

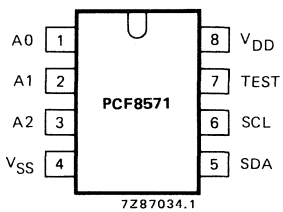


Fig. 2 Pinning diagram.

RATINGS

Limiting values in accordance with the Absolute Maximum System (IEC 134)

Supply voltage range (pin 8)	V_{DD}		−0,8 to + 8,0	V
Voltage range on any input	V_I		−0,8 to V_{DD} + 0,8	V
D.C. input current (any input)	$\pm I_I$	max.	10	mA
D.C. output current (any output)	$\pm I_O$	max.	10	mA
Supply current (pin 4 or pin 8)	$\pm I_{DD}$; I_{SS}	max.	50	mA
Power dissipation per package	P_{tot}	max.	300	mW
Power dissipation per output	P	max.	50	mW
Storage temperature range	T_{stg}		−65 to + 150	°C
Operating temperature range	T_{amb}		−40 to + 85	°C

CHARACTERISTICS

V_{DD} = 2,5 to 6 V; V_{SS} = 0 V; T_{amb} = −40 to + 85 °C unless otherwise specified

parameter	symbol	min.	typ.	max.	unit
SUPPLY					
Supply voltage	V_{DD}	2,5	–	6	V
Supply current $V_I = V_{SS}$ or V_{DD}					
operating at f_{SCL} = 100 kHz;	I_{DD}	–	–	200	μA
standby at f_{SCL} = 0 Hz	I_{DDO}	–	–	15	μA
standby at T_{amb} = −25 to 70 °C	I_{DDO}	–	–	5	μA
Power-on reset voltage level at $V_{SCL} = V_{SDA} = V_{DD}$	V_{POR}	1,5	1,9	2,3	V
Inputs; input/output SDA					
Input voltage LOW**	V_{IL}	−0,8	–	0,3 x V_{DD}	V
Input voltage HIGH**	V_{IH}	0,7 x V_{DD}	–	V_{DD} + 0,8	V
Output current LOW at V_{OL} = 0,4 V	I_{OL}	3	–	–	mA
Output leakage current HIGH at $V_{OH} = V_{DD}$	I_{OH}	–	–	250	nA
Input leakage current at $V_I = V_{DD}$ or V_{SS}	$\pm I_I$	–	–	250	nA
Clock frequency (Fig. 7)	f_{SCL}	0	–	100	kHz
Input capacitance (SCL, SDA) at $V_I = V_{SS}$	C_I	–	–	7	pF
Tolerable spike width on bus	t_{SW}	–	–	100	ns
LOW V_{DD} data retention					
Supply voltage for data retention	V_{DDR}	1	–	6	V
Supply current at V_{DDR} = 1 V	I_{DDR}	–	–	5	μA
Supply current at V_{DDR} = 1 V; T_{amb} = −25 to 70 °C	I_{DDR}	–	–	2	μA
Power saving mode (Fig. 14)					
Supply current at T_{amb} = 25 °C; TEST = V_{DDR}	I_{DDS}	–	50	200	nA

* The power-on reset circuit resets the I²C bus logic when $V_{DD} < V_{POR}$.

** If the input voltages are a diode voltage above or below the supply voltage V_{DD} or V_{SS} an input current will flow: this current must not exceed ± 0,5 mA.

CHARACTERISTICS OF THE I²C BUS

The I²C bus is for 2-way, 2-line communication between different ICs or modules. The two lines are a serial data line (SDA) and a serial clock line (SCL). Both lines must be connected to a positive supply via a pull-up resistor when connected to the output stages of a device. Data transfer may be initiated only when the bus is not busy.

Bit transfer

One data bit is transferred during each clock pulse. The data on the SDA line must remain stable during the HIGH period of the clock pulse as changes in the data line at this time will be interpreted as control signals.

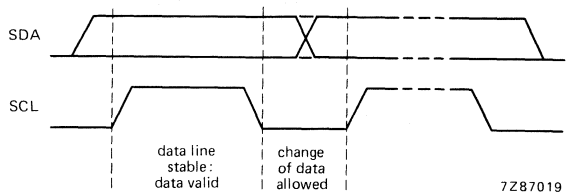


Fig. 3 Bit transfer.

Start and stop conditions

Both data and clock lines remain HIGH when the bus is not busy. A HIGH-to-LOW transition of the data line, while the clock is HIGH is defined as the start condition (S). A LOW-to-HIGH transition of the data line while the clock is HIGH is defined as the stop condition (P).

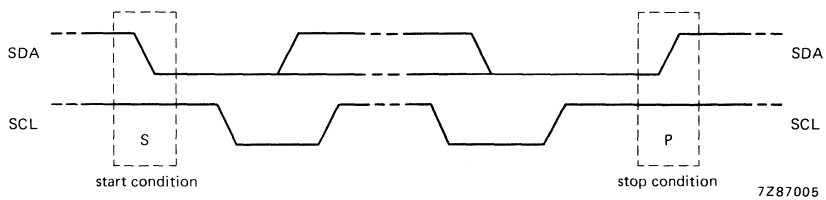


Fig. 4 Definition of start and stop conditions.

System configuration

A device generating a message is a "transmitter", a device receiving a message is the "receiver". The device that controls the message is the "master" and the devices which are controlled by the master are the "slaves".

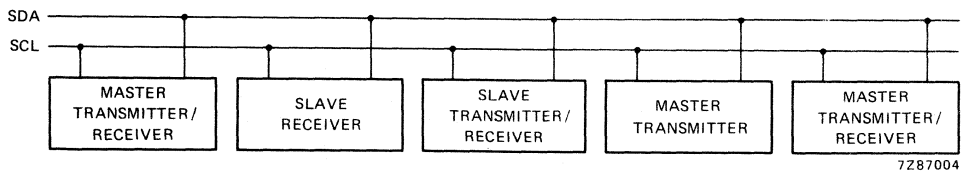


Fig. 5 System configuration.

Acknowledge

The number of data bytes transferred between the start and stop conditions from transmitter to receiver is not limited. Each byte of eight bits is followed by one acknowledge bit. The acknowledge bit is a HIGH level put on the bus by the transmitter whereas the master generates an extra acknowledge related clock pulse. A slave receiver which is addressed must generate an acknowledge after the reception of each byte. Also a master must generate an acknowledge after the reception of each byte that has been clocked out of the slave transmitter. The device that acknowledges has to pull down the SDA line during the acknowledge clock pulse, so that the SDA line is stable LOW during the HIGH period of the acknowledge related clock pulse, set-up and hold times must be taken into account. A master receiver must signal an end of data to the transmitter by *not* generating an acknowledge on the last byte that has been clocked out of the slave. In this event the transmitter must leave the data line HIGH to enable the master to generate a stop condition.

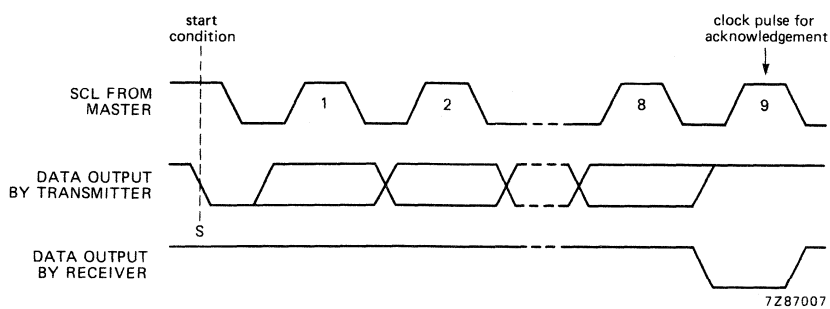


Fig. 6 Acknowledgement on the I^2C bus.

Timing specifications

Within the I^2C bus specifications a high-speed mode and a low-speed mode are defined. The PCF8571 operates in both modes and the timing requirements are as follows:

High-speed mode

Masters generate a bus clock with a maximum frequency of 100 kHz. Detailed timing is shown in Fig. 7.

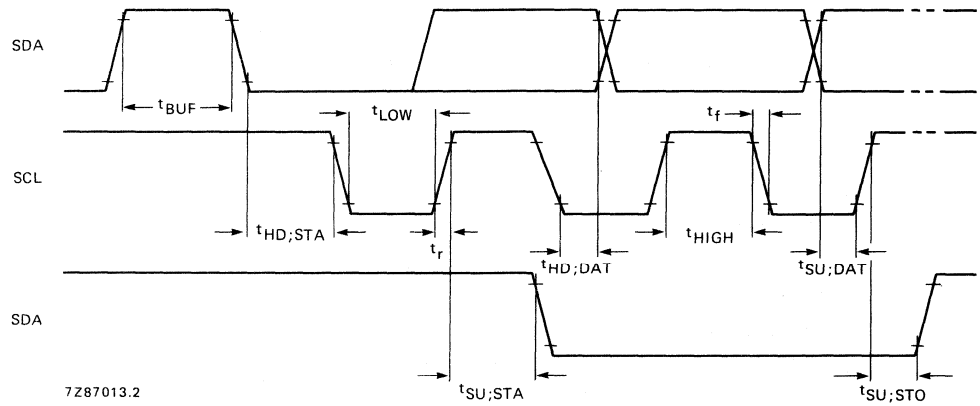


Fig. 7 Timing of the high-speed mode.

Where:

t_{BUF}	$t \geqslant t_{LOWmin}$	The minimum time the bus must be free before a new transmission can start
$t_{HD;STA}$	$t \geqslant t_{HIGHmin}$	Start condition hold time
t_{LOWmin}	4,7 μs	Clock LOW period
$t_{HIGHmin}$	4 μs	Clock HIGH period
$t_{SU;STA}$	$t \geqslant t_{LOWmin}$	Start condition set-up time, only valid for repeated start code
$t_{HD;DAT}$	$t \geqslant 0$ μs	Data hold time
$t_{SU;DAT}$	$t \geqslant 250$ ns	Data set-up time
t_r	$t \leqslant 1$ μs	Rise time of both the SDA and SCL line
t_f	$t \leqslant 300$ ns	Fall time of both the SDA and SCL line
$t_{SU;STO}$	$t \geqslant t_{LOWmin}$	Stop condition set-up time

Note

All the timing values refer to V_{IH} and V_{IL} levels with a voltage swing of V_{SS} to V_{DD}.

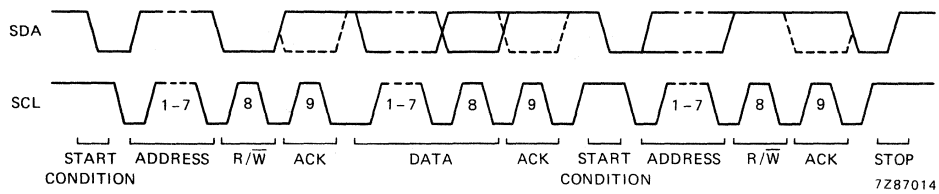


Fig. 8 Complete data transfer in the high-speed mode.

Where:

Clock t_{LOWmin}	4,7 μs
$t_{HIGHmin}$	4 μs

The dashed line is the acknowledgement of the receiver

Mark-to-space ratio	1 : 1 (LOW-to-HIGH)
Max. number of bytes	unrestricted
Premature termination of transfer	allowed by generation of STOP condition
Acknowledge clock bit	must be provided by the master

Low-speed mode

Masters generate a bus clock with a maximum frequency of 2 kHz; a minimum LOW period of 105 μs and a minimum HIGH period of 365 μs. The mark-to-space ratio is 1 : 3 LOW-to-HIGH. Detailed timing is shown in Fig. 9.

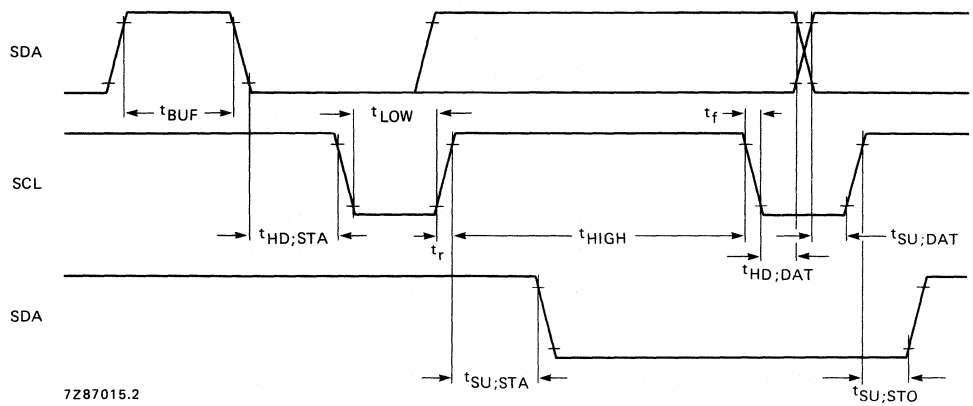


Fig. 9 Timing of the low-speed mode.

Timing specifications (continued)

Where:

t_{BUF}	$t \geqslant 105\ \mu s$ (t_{LOWmin})
$t_{HD;STA}$	$t \geqslant 365\ \mu s$ ($t_{HIGHmin}$)
t_{LOW}	$130\ \mu s \pm 25\ \mu s$
t_{HIGH}	$390\ \mu s \pm 25\ \mu s$
$t_{SU;STA}$	$130\ \mu s \pm 25\ \mu s$ *
$t_{HD;DAT}$	$t \geqslant 0\ \mu s$
$t_{SU;DAT}$	$t \geqslant 250$ ns
t_r	$t \leqslant 1\ \mu s$
t_f	$t \leqslant 300$ ns
$t_{SU;STO}$	$130\ \mu s \pm 25\ \mu s$

Note

All the timing values refer to V_{IH} and V_{IL} levels with a voltage swing of V_{SS} to V_{DD}. For definitions see high-speed mode.

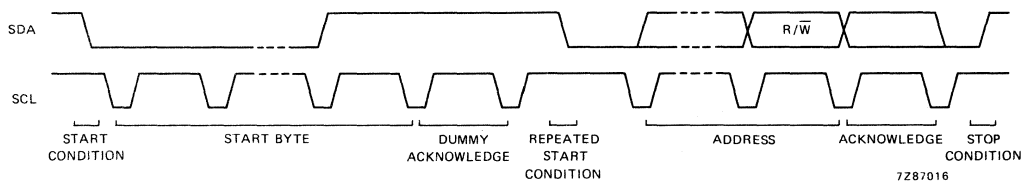


Fig. 10 Complete data transfer in the low-speed mode.

Where:

Clock t_{LOWmin}	$130\ \mu s \pm 25\ \mu s$
$t_{HIGHmin}$	$390\ \mu s \pm 25\ \mu s$
Mark-to-space ratio	1 : 3 (LOW-to-HIGH)
Start byte	0000 0001
Max. number of bytes	6
Premature termination of transfer	not allowed
Acknowledge clock bit	must be provided by master

Note

The general characteristics and detailed specification of the I^2C bus are described in a separate data sheet (serial data buses) in handbook "ICs for digital systems in radio, audio and video equipment".

* Only valid for repeated start code.

Bus protocol

Before any data is transmitted on the I²C bus, the device which should respond is addressed first. The addressing is always done with the first byte transmitted after the start procedure. The I²C bus configuration for different PCF8571 READ and WRITE cycles is shown in Fig. 11.

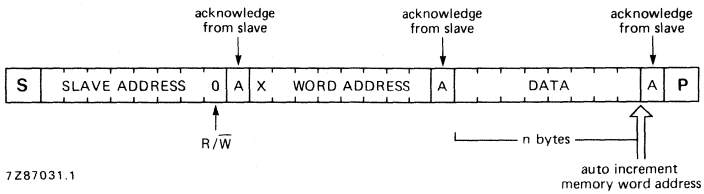


Fig. 11(a) Master transmits to slave receiver (WRITE mode).

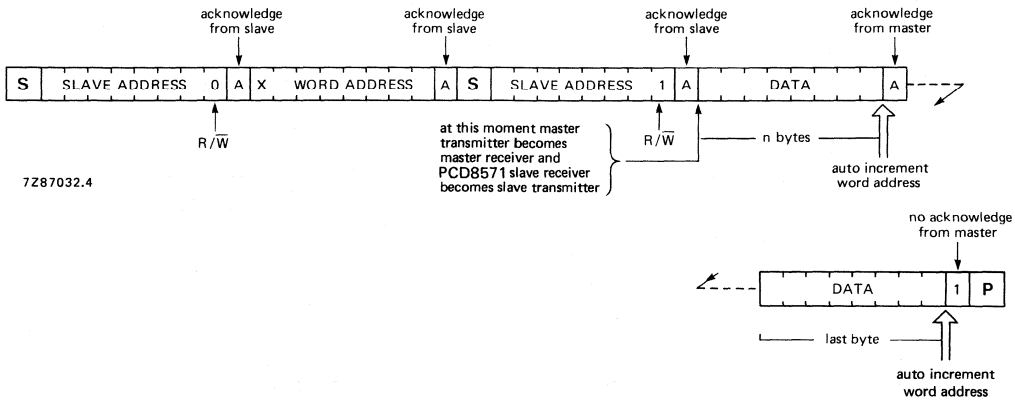


Fig. 11(b) Master reads after setting word address (WRITE word address; READ data).

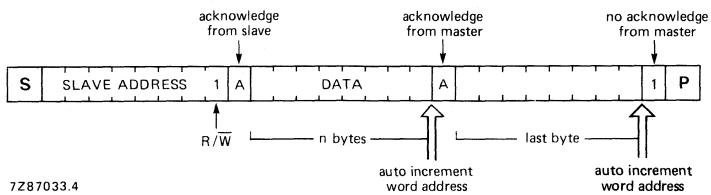


Fig. 11(c) Master reads slave immediately after first byte (READ mode).

Note

X = don't care bit.

APPLICATION INFORMATION

The PCF8571 slave address has a fixed combination 1010 as group 1, while group 2 is fully programmable (see Fig. 12).

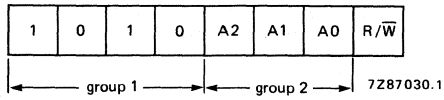


Fig. 12 PCF8571 address.

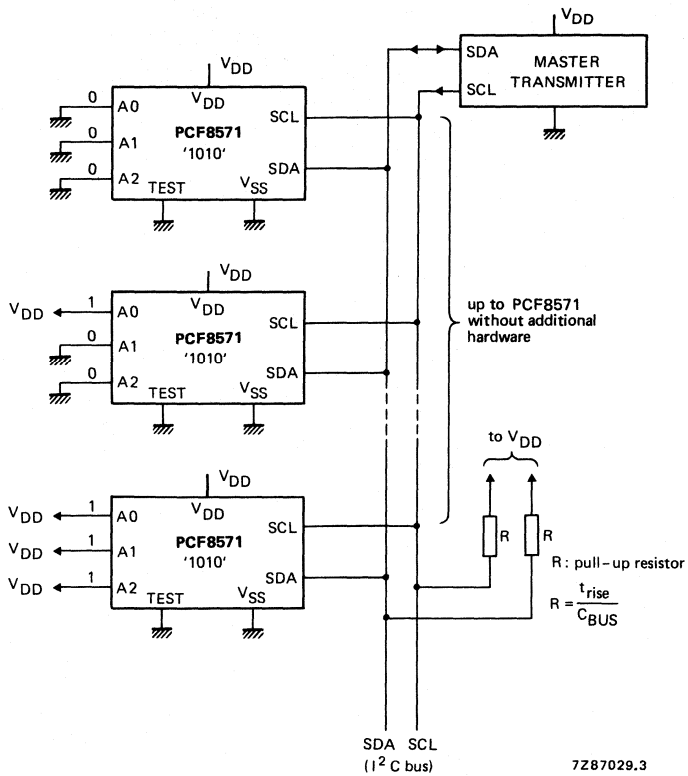


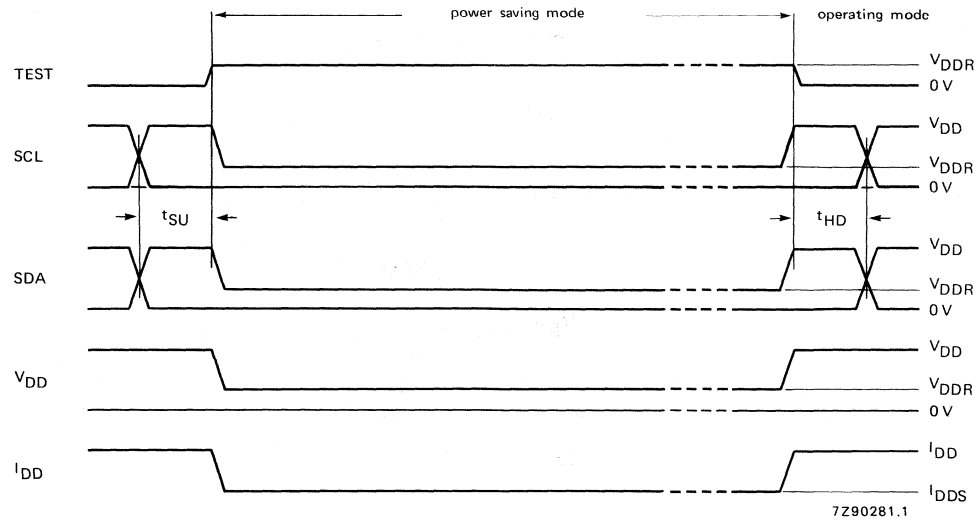
Fig. 13 PCF8571 application diagram.

Note

A0, A1, and A2 inputs must be connected to V_{DD} or V_{SS} but not left open.

POWER SAVING MODE

With the condition TEST = V_{DDR}, the PCF8571 goes into the power saving mode and I^2C bus logic is reset.



Where:
$t_{SU} \geqslant 4\ \mu s$
$t_{HD} \geqslant 4\ \mu s$

Fig. 14 Timing for power saving mode.

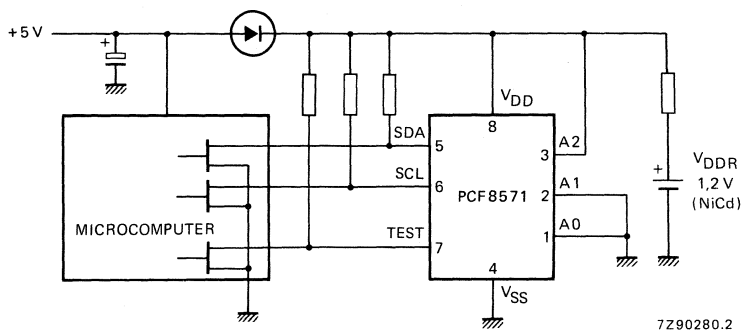
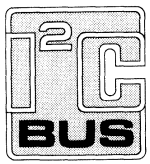


Fig. 15 Application example for power saving mode.

Note

1. In the operating mode, TEST = 0.
2. In the power saving mode, TEST = V_{DDR}.



Purchase of Philips' I²C components conveys a license under the Philips' I²C patent to use the components in the I²C-system provided the system conforms to the I²C specifications defined by Philips.

CLOCK/CALENDAR WITH SERIAL I/O

GENERAL DESCRIPTION

The PCF8573 is a low threshold, monolithic CMOS circuit that functions as a real time clock/calendar in the Inter IC (I^2C) bus-oriented microcomputer systems. The device includes an addressable time counter and an addressable alarm register, both for minutes, hours, days and months. Three special control/status flags, COMP, POWF and NODA, are also available. Information is transferred serially via a two line bidirectional bus (I^2C). Back-up for the clock during supply interruptions is provided by a 1,2 V nickel cadmium battery. The time base is generated from a 32,768 kHz crystal-controlled oscillator.

Features

- Serial input/output bus (I^2C) interface for minutes, hours, days and months
- Additional pulse outputs for seconds and minutes
- Alarm register for presetting a time for alarm or remote switching functions
- Battery back-up for clock function during supply interruption
- Crystal oscillator control (32,768 kHz)

QUICK REFERENCE DATA

Supply voltage range (clock)	$V_{DD}-V_{SS1}$		1,1 to 6,0 V
Supply voltage range (I^2C interface)	$V_{DD}-V_{SS2}$		2,5 to 6,0 V
Crystal oscillator frequency	f_{osc}	typ.	32,768 kHz

PACKAGE OUTLINES

PCF8573P: 16-lead DIL; plastic (SOT-38).
PCF8573T: 16-lead mini-pack; plastic (SO-16L; SOT-162A).

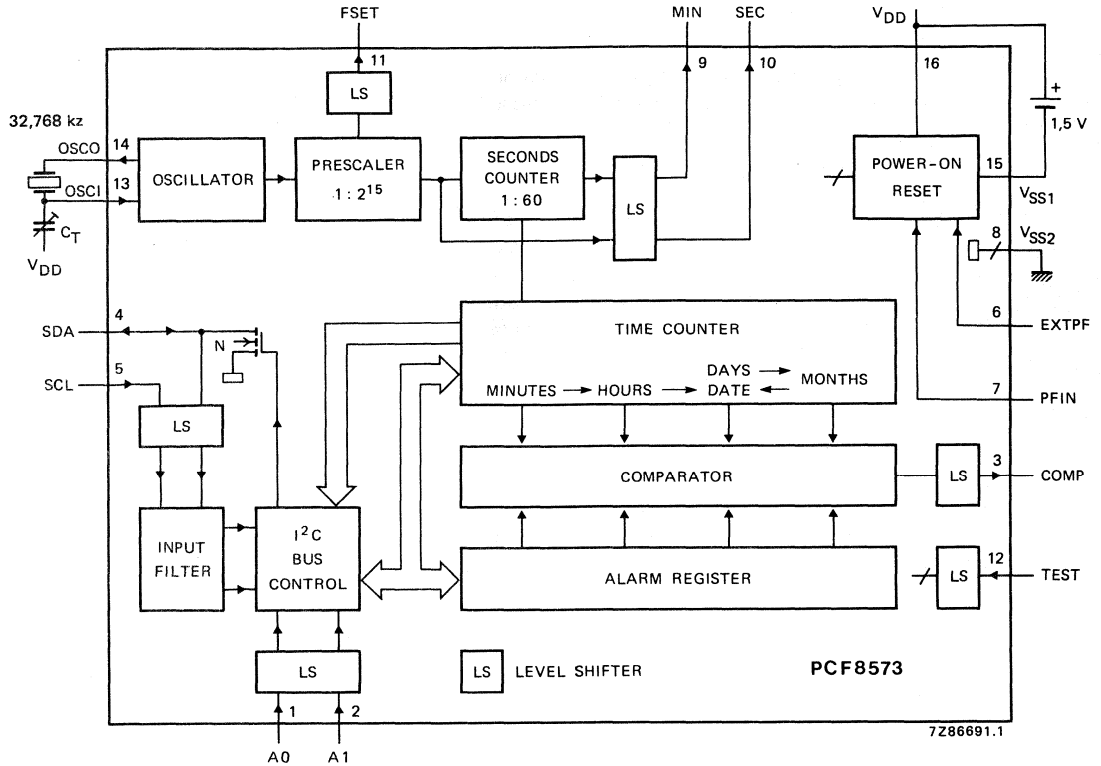


Fig. 1 Block diagram.

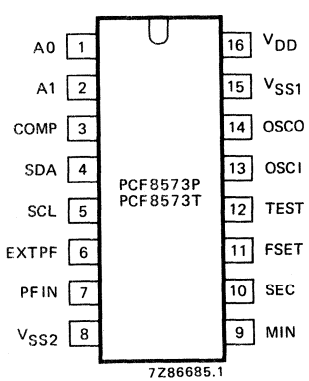


Fig. 2 Pinning diagram.

PINNING

1	A0	address input	
2	A1	address input	
3	COMP	comparator output	
4	SDA	serial data line	I^2C bus
5	SCL	serial clock line	I^2C bus
6	EXTPF	enable power fail flag input	
7	PFIN	power fail flag input	
8	V_{SS2}	negative supply 2 (I^2C interface)	
9	MIN	one pulse per minute output	
10	SEC	one pulse per second output	
11	FSET	oscillator tuning output	
12	TEST	test input; must be connected to V_{SS2} when not in use	
13	OSCI	oscillator input	
14	OSCO	oscillator input/output	
15	V_{SS1}	negative supply 1 (clock)	
16	V_{DD}	common positive supply	

FUNCTIONAL DESCRIPTION

Oscillator

The PCF8573 has an integrated crystal-controlled oscillator which provides the timebase for the prescaler. The frequency is determined by a single 32,768 kHz crystal connected between OSCI and OSCO. A trimmer is connected between OSCI and V_{DD}.

Prescaler and time counter

The prescaler provides a 128 Hz signal at the FSET output for fine adjustment of the crystal oscillator without loading it. The prescaler also generates a pulse once a second to advance the seconds counter. The carry of the prescaler and the seconds counter are available at the outputs SEC, MIN respectively, and are also readable via the I^2C bus. The mark-to-space ratio of both signals is 1 : 1. The time counter is advanced one count by the falling edge of output signal MIN. A transition from HIGH to LOW of output signal SEC triggers MIN to change state. The time counter counts minutes, hours, days and months, and provides a full calendar function which needs to be corrected once every four years. Cycle lengths are shown in Table 1.

Table 1 Cycle length of the time counter

unit	number of bits	counting cycle	carry for following unit	content of month counter
minutes	7	00 to 59	59 → 00	
hours	6	00 to 23	23 → 00	
days	6	01 to 28	28 → 01	2 (see note)
			or 29 → 01	
		01 to 30	30 → 01	4, 6, 9, 11
		01 to 31	31 → 01	1, 3, 5, 7, 8, 10, 12
months	5	01 to 12	12 → 01	

Note: Day counter may be set to 29 by a write transmission with EXECUTE ADDRESS.

Alarm register

The alarm register is a 24-bit memory. It stores the time-point for the next setting of the status flag COMP. Details of writing and reading of the alarm register are included in the description of the characteristics of the I^2C bus.

Comparator

The comparator compares the contents of the alarm register and the time counter, each with a length of 24 bits. When these contents are equal the flag COMP will be set 4 ms after the falling edge of MIN. This set condition occurs once at the beginning of each minute. This information is latched, but can be cleared by an instruction via the I^2C bus. A clear instruction may be transmitted immediately after the flag is set and will be executed. Flag COMP information is also available at the output COMP. The comparison may be based upon hours and minutes only if the internal flag NODA (no date) is set. Flag NODA can be set and cleared by separate instructions via the I^2C bus, but it is undefined until the first set or clear instruction has been received. Both COMP and NODA flags are readable via the I^2C bus.

FUNCTIONAL DESCRIPTION (continued)

Power on/power fail detection

If the voltage $V_{DD}-V_{SS1}$ falls below a certain value the operation of the clock becomes undefined. Thus a warning signal is required to indicate that faultless operation of the clock is not guaranteed. This information is latched in a flag called POWF (Power Fail) and remains latched after restoration of the correct supply voltage until a write procedure with EXECUTE ADDRESS has been received. The flag POWF can be set by an internally generated power fail level-discriminator signal for application with ($V_{DD}-V_{SS1}$) greater than V_{TH1}, or by an externally generated power fail signal for application with ($V_{DD}-V_{SS1}$) less than V_{TH1}. The external signal must be applied to the input PFIN. The input stage operates with signals of any slow rise and fall times. Internally or externally controlled POWF can be selected by input EXTPF as shown in Table 2.

Table 2 Power fail selection

EXTPF	PFIN	function
0	0	power fail is sensed internally
0	1	test mode
1	0	power fail is sensed externally
1	1	no power fail sensed

0 : connected to V_{SS1} (LOW)
1 : connected to V_{DD} (HIGH)

The external power fail control operates by absence of the $V_{DD}-V_{SS2}$ supply. Therefore the input levels applied to PFIN and EXTPF must be within the range of $V_{DD}-V_{SS1}$. A LOW level at PFIN indicates a power fail. POWF is readable via the I^2C bus. A power on reset for the I^2C bus control is generated on-chip when the supply voltage $V_{DD}-V_{SS2}$ is less than V_{TH2}.

Interface level shifters

The level shifters adjust the 5 V operating voltage ($V_{DD}-V_{SS2}$) of the microcontroller to the internal supply voltage ($V_{DD}-V_{SS1}$) of the clock/calendar. The oscillator and counter are not influenced by the $V_{DD}-V_{SS2}$ supply voltage. If the voltage $V_{DD}-V_{SS2}$ is absent ($V_{SS2} = V_{DD}$) the output signal of the level shifter is HIGH because V_{DD} is the common node of the $V_{DD}-V_{SS2}$ and the $V_{DD}-V_{SS1}$ supplies. Because the level shifters invert the input signal, the internal circuit behaves as if a LOW signal is present on the inputs. FSET, SEC, MIN and COMP are CMOS push-pull output stages. The driving capability of these outputs is lost when the supply voltage $V_{DD}-V_{SS2} = 0$.

CHARACTERISTICS OF THE I²C BUS

The I²C bus is for 2-way, 2-line communication between different ICs or modules. The two lines are a serial data line (SDA) and a serial clock line (SCL). Both lines must be connected to a positive supply via a pull-up resistor when connected to the output stages of a device. Data transfer may be initiated only when the bus is not busy.

Bit transfer (see Fig. 3)

One data bit is transferred during each clock pulse. The data on the SDA line must remain stable during the HIGH period of the clock pulse as changes in the data line at this time will be interpreted as control signals.

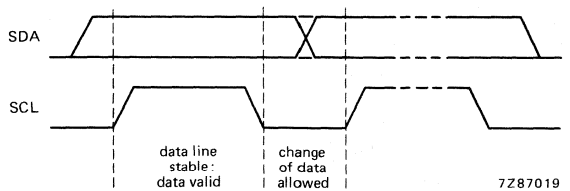


Fig. 3 Bit transfer.

Start and stop conditions (see Fig. 4)

Both data and clock lines remain HIGH when the bus is not busy. A HIGH-to-LOW transition of the data line, while the clock is HIGH is defined as the start condition (S). A LOW-to-HIGH transition of the data line while the clock is HIGH is defined as the stop condition (P).

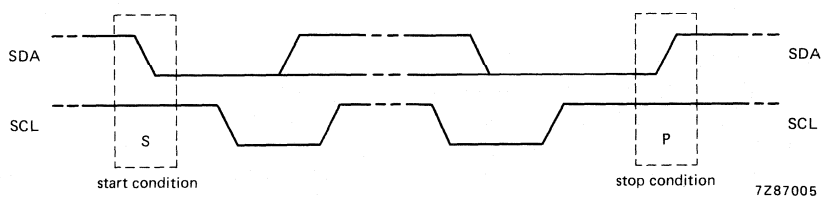


Fig. 4 Definition of start and stop conditions.

System configuration (see Fig. 5)

A device generating a message is a "transmitter", a device receiving a message is the "receiver". The device that controls the message is the "master" and the devices which are controlled by the master are the "slaves".

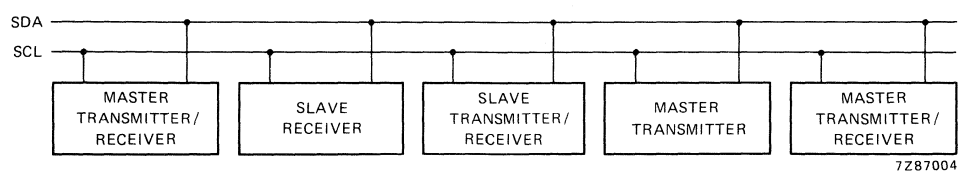


Fig. 5 System configuration.

CHARACTERISTICS OF THE I²C bus (continued)

Acknowledge (see Fig. 6)

The number of data bytes transferred between the start and stop conditions from transmitter to receiver is not limited. Each byte of eight bits is followed by one acknowledge bit. The acknowledge bit is a HIGH level put on the bus by the transmitter whereas the master generates an extra acknowledge related clock pulse. A slave receiver which is addressed must generate an acknowledge after the reception of each byte. Also a master must generate an acknowledge after the reception of each byte that has been clocked out of the slave transmitter. The device that acknowledges has to pull down the SDA line during the acknowledge clock pulse, so that the SDA line is stable LOW during the HIGH period of the acknowledge related clock pulse, set up and hold times must be taken into account. A master receiver must signal an end of data to the transmitter by *not* generating an acknowledge on the last byte that has been clocked out of the slave. In this event the transmitter must leave the data line HIGH to enable the master to generate a stop condition. (See Fig. 13 and Fig. 14.)

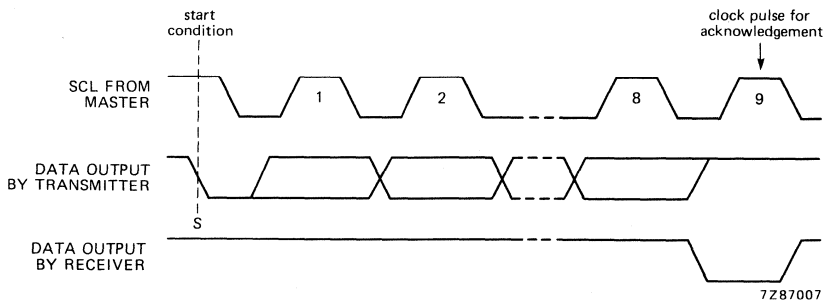


Fig. 6 Acknowledgement on the I²C bus.

Timing specifications

Within the I²C bus specifications a high-speed mode and a low-speed mode are defined. The PCF8573 operates in both modes and the timing requirements are as follows:

High-speed mode

Masters generate a bus clock with a maximum frequency of 100 kHz. Detailed timing is shown in Fig. 7.

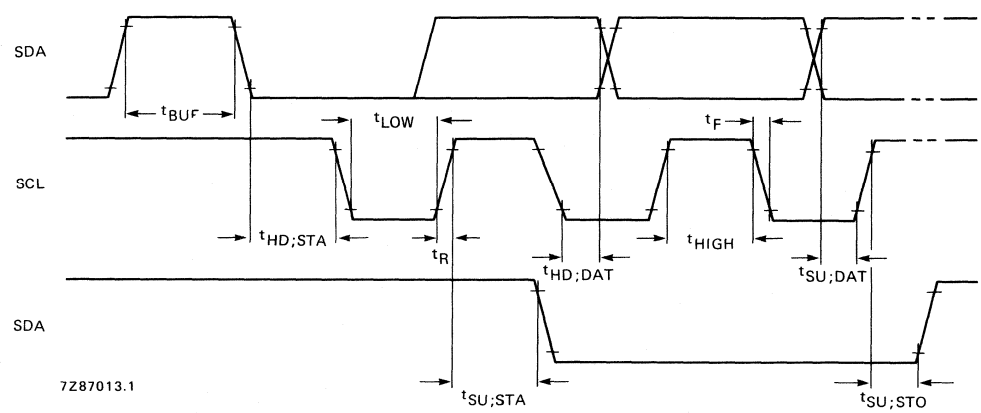


Fig. 7 Timing of the high-speed mode.

Where:

t_{BUF}	$t \geqslant t_{LOWmin}$	The minimum time the bus must be free before a new transmission can start
$t_{HD;STA}$	$t \geqslant t_{HIGHmin}$	Start condition hold time
t_{LOWmin}	4,7 μs	Clock LOW period
$t_{HIGHmin}$	4 μs	Clock HIGH period
$t_{SU;STA}$	$t \geqslant t_{LOWmin}$	Start condition set-up time, only valid for repeated start code
$t_{HD;DAT}$	$t \geqslant 0$ μs	Data hold time
$t_{SU;DAT}$	$t \geqslant 250$ ns	Data set-up time
t_R	$t \leqslant 1$ μs	Rise time of both the SDA and SCL line
t_F	$t \leqslant 300$ ns	Fall time of both the SDA and SCL line
$t_{SU;STO}$	$t \geqslant t_{LOWmin}$	Stop condition set-up time

Note

All the values refer to V_{IH} and V_{IL} levels with a voltage swing of V_{DD} to V_{SS2}.

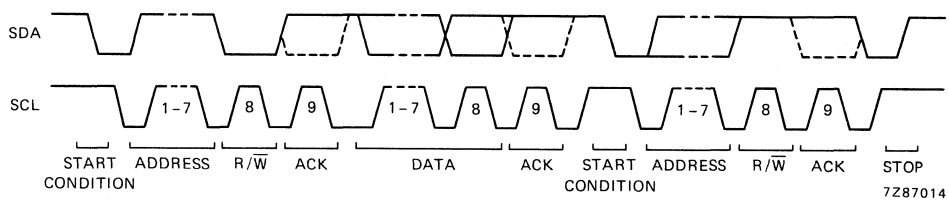


Fig. 8 Complete data transfer in the high-speed mode.

Where:

Clock t_{LOWmin}	4,7 μs
$t_{HIGHmin}$	4 μs

The dashed line is the acknowledgement of the receiver

Mark-to-space ratio	1 : 1 (LOW-to-HIGH)
Max. number of bytes	unrestricted
Premature termination of transfer	allowed by generation of STOP condition
Acknowledge clock bit	must be provided by the master

CHARACTERISTICS OF THE I²C BUS (continued)

Low-speed mode

Masters generate a bus clock with a maximum frequency of 2 kHz; a minimum LOW period of 105 μs and a minimum HIGH period of 365 μs. The mark-to-space ratio is 1 : 3 LOW-to-HIGH. Detailed timing is shown in Fig. 9.

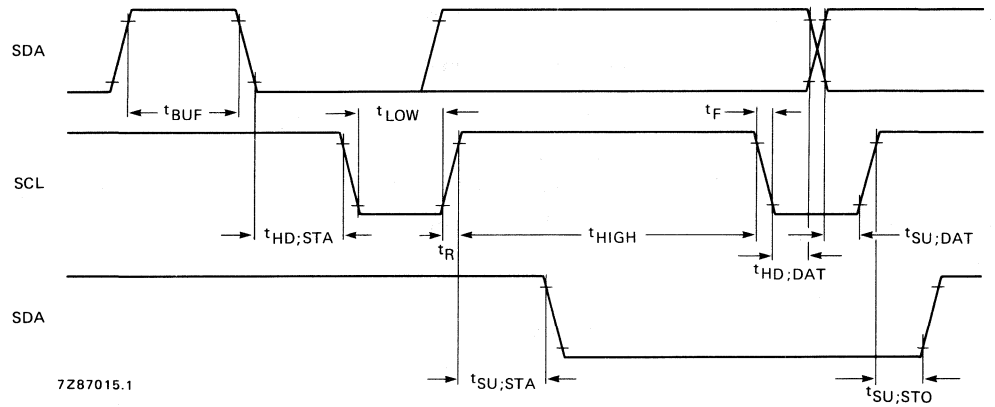


Fig. 9 Timing of the low-speed mode.

Where:

t_{BUF}	$t \geqslant 105\ \mu s$ (t_{LOWmin})
$t_{HD;STA}$	$t \geqslant 365\ \mu s$ ($t_{HIGHmin}$)
t_{LOW}	$130\ \mu s \pm 25\ \mu s$
t_{HIGH}	$390\ \mu s \pm 25\ \mu s$
$t_{SU;STA}$	$130\ \mu s \pm 25\ \mu s$*
$t_{HD;DAT}$	$t \geqslant 0\ \mu s$
$t_{SU;DAT}$	$t \geqslant 250$ ns
t_R	$t \leqslant 1\ \mu s$
t_F	$t \leqslant 300$ ns
$t_{SU;STO}$	$130\ \mu s \pm 25\ \mu s$

Note

All the values refer to V_{IH} and V_{IL} levels with a voltage swing of V_{DD} to V_{SS2}, for definitions see high-speed mode.

* Only valid for repeated start code.

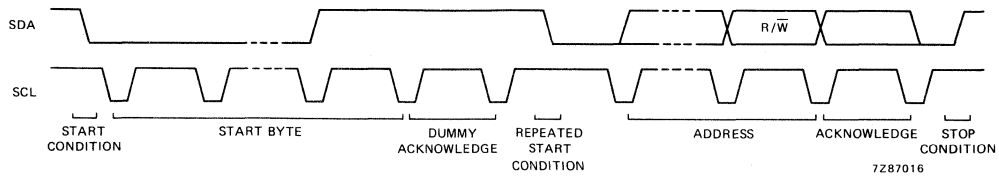


Fig. 10 Complete data transfer in the low-speed mode.

Where:

Clock t_{LOWmin}	130 μs ± 25 μs
$t_{HIGHmin}$	390 μs ± 25 μs
Mark-to-space ratio	1 : 3 (LOW-to-HIGH)
Start byte	0000 0001
Max. number of bytes	6
Premature termination of transfer	not allowed
Acknowledge clock bit	must be provided by master

Note

The general characteristics and detailed specification of the I^2C bus are described in a separate data sheet (serial data buses) in handbook: ICs for digital systems in radio, audio and video equipment.

ADDRESSING

Before any data is transmitted on the I²C bus, the device which should respond is addressed first. The addressing is always done with the first byte transmitted after the start procedure.

Slave address

The clock/calendar acts as a slave receiver or slave transmitter. Therefore the clock signal SCL is only an input signal, but the data signal SDA is a bidirectional line. The clock calendar slave address is shown in Fig. 11.

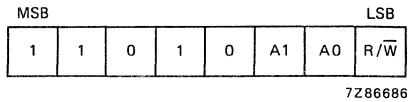


Fig. 11 Slave address.

The subaddress bits A0 and A1 correspond to the two hardware address pins A0 and A1 which allows the device to have 1 of 4 different addresses.

Clock/calendar READ/WRITE cycles

The I²C bus configuration for different clock/calendar READ and WRITE cycles is shown in Fig. 12 and Fig. 13.

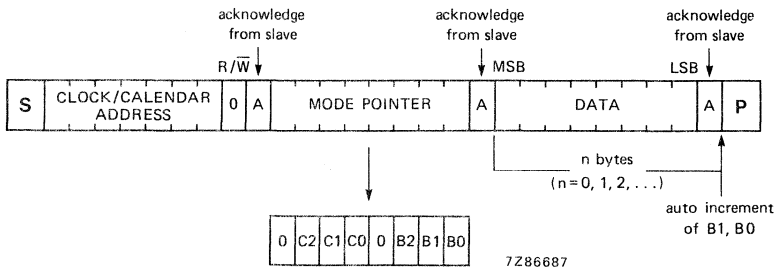


Fig. 12 Master transmitter transmits to clock/calendar slave receiver.

The write cycle is used to set the time counter, the alarm register and the flags. The transmission of the clock/calendar address is followed by the MODE-POINTER-WORD which contains a CONTROL-nibble (Table 3) and an ADDRESS-nibble (Table 4). The ADDRESS-nibble is valid only if the preceding CONTROL-nibble is set to EXECUTE ADDRESS. The third transmitted word contains the data to be written into the time counter or alarm register.

Table 3 CONTROL-nibble

	C2	C1	C0	function
0	0	0	0	execute address
0	0	0	1	read control/status flags
0	0	1	0	reset prescaler, including seconds counter; without carry for minute counter
0	0	1	1	time adjust, with carry for minute counter (see note)
0	1	0	0	reset NODA flag
0	1	0	1	set NODA flag
0	1	1	0	reset COMP flag

Note

If the seconds counter is below 30 there is no carry. This causes a time adjustment of max. −30 s. From the count 30 there is a carry which adjusts the time by max. + 30 s.

Table 4 ADDRESS-nibble

	B2	B1	B0	addressed to:
0	0	0	0	time counter hours
0	0	0	1	time counter minutes
0	0	1	0	time counter days
0	0	1	1	time counter months
0	1	0	0	alarm register hours
0	1	0	1	alarm register minutes
0	1	1	0	alarm register days
0	1	1	1	alarm register months

At the end of each data word the address bits B1, B0 will be incremented automatically provided the preceding CONTROL-nibble is set to EXECUTE ADDRESS. There is no carry to B2.

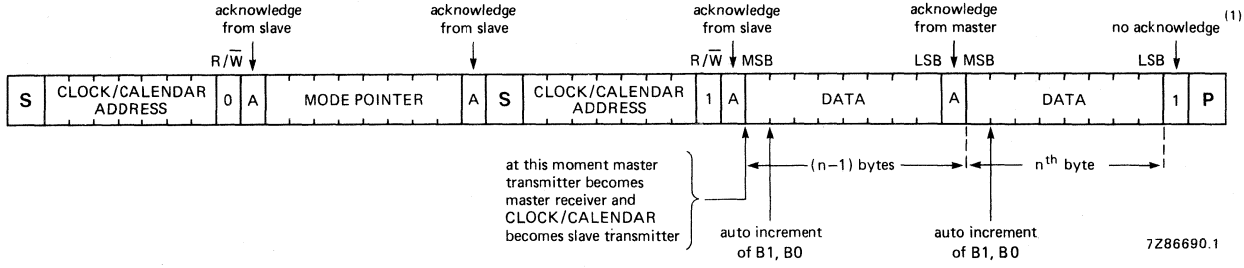
Table 5 shows the placement of the BCD upper and lower digits in the DATA byte for writing into the addressed part of the time counter and alarm register respectively.

Table 5 Placement of BCD digits in the DATA byte

MSB DATA LSB								
upper digit				lower digit				
UD	UC	UB	UA	LD	LC	LB	LA	addressed to:
X	X	D	D	D	D	D	D	hours
X	D	D	D	D	D	D	D	minutes
X	X	D	D	D	D	D	D	days
X	X	X	D	D	D	D	D	months

Where "X" is the don't care bit and "D" is the data bit.

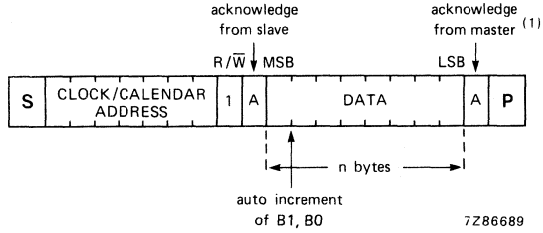
Acknowledgement response of the clock calendar as slave receiver is shown in Table 6.



(1) The master receiver must signal an end of data to the slave transmitter by *not* generating an acknowledge on the *last byte* that has been clocked out of the slave.

Fig. 13 Master transmitter reads clock/calendar after setting mode pointer.

To read the addressed part of the time counter and alarm register, plus information from specified control/status flags, the BCD digits in the DATA byte are organized as shown in Table 7.



(1) The master receiver must signal an end of data to the slave transmitter by *not* generating an acknowledge on the *last byte* that has been clocked out of the slave.

Fig. 14 Master reads clock/calendar immediately after first byte.

The status of the MODE-POINTER-WORD concerning the CONTROL-nibble remains unchanged until a write to MODE POINTER condition occurs.

ADDRESSING (continued)

Table 6 Slave receiver acknowledgement

mode pointer								acknowledge on byte		
								address	mode pointer	data
	C2	C1	C0		B2	B1	B0			
0	0	0	0	0	X	X	X	yes	yes	yes
0	0	0	0	1	X	X	X	yes	no	no
0	0	0	1	X	X	X	X	yes	yes	no
0	0	1	0	X	X	X	X	yes	yes	no
0	0	1	1	X	X	X	X	yes	yes	no
0	1	0	0	X	X	X	X	yes	yes	no
0	1	0	1	X	X	X	X	yes	yes	no
0	1	1	0	X	X	X	X	yes	yes	no
0	1	1	1	X	X	X	X	yes	no	no
1	X	X	X	X	X	X	X	yes	no	no

Where "X" is the don't care bit.

Table 7 Organization of the BCD digits in the DATA byte

MSB DATA LSB								
upper digit				lower digit				
UD	UC	UB	UA	LD	LC	LB	LA	addressed to
0	0	D	D	D	D	D	D	hours
0	D	D	D	D	D	D	D	minutes
0	0	D	D	D	D	D	D	days
0	0	0	D	D	D	D	D	months
0	0	0	*	**	NODA	COMP	POWF	control/status flags

Where: "D" is the data bit.
* = minutes.
** = seconds.

RATINGS

Limiting values in accordance with the Absolute Maximum System (IEC 134)

Supply voltage ranges	$V_{DD}-V_{SS1}$		−0,3 to + 8	V
	$V_{DD}-V_{SS2}$		−0,3 to + 8	V
Voltage on pins 4 and 5			V_{SS2}−0,8 to V_{DD} + 0,8	V*
Voltage on pins 6, 7, 13 and 14			V_{SS1}−0,6 to V_{DD} + 0,6	V
Voltage on any other pin			V_{SS2}−0,6 to V_{DD} + 0,6	V
Input current	I_I	max.	10	mA
Output current	I_O	max.	10	mA
Power dissipation per output	P_O	max.	100	mW
Total power dissipation per package	P_{tot}	max.	200	mW
Operating ambient temperature range	T_{amb}		−40 to + 85	°C
Storage temperature range	T_{stg}		−55 to + 125	°C

HANDLING

Inputs and outputs are protected against electrostatic charge in normal handling. However, to be totally safe, it is desirable to take normal precautions appropriate to handling MOS devices (see 'Handling MOS Devices').

* Impedance min. 500 Ω.

CHARACTERISTICS

V_{SS2} = 0 V; T_{amb} = −40 to + 85 °C unless otherwise specified. Typical values at T_{amb} = + 25 °C

parameter	symbol	min.	typ.	max.	unit	
Supply						
Supply voltage (I^2C interface)	$V_{DD}-V_{SS2}$	2,5	5	6,0	V	
Supply voltage (clock)	$V_{DD}-V_{SS1}$	1,3	1,5	$(V_{DD}-V_{SS2})$	V	←
Supply voltage (clock) at $t_{HD} > 500$ ns	$V_{DD}-V_{SS1}$	1,1	1,5	$(V_{DD}-V_{SS2})$	V	←
Supply current V_{SS1}						
at $V_{DD}-V_{SS1}$ = 1,5 V	$-I_{SS1}$	–	3	10	μA	
at $V_{DD}-V_{SS1}$ = 5 V	$-I_{SS1}$	–	12	50	μA	
Supply current V_{SS2}						
at $V_{DD}-V_{SS2}$ = 5 V (I_O = 0 mA on all outputs)	$-I_{SS2}$	–	–	50	μA	
Inputs SCL, SDA, A0, A1, TEST						
Input voltage HIGH	V_{IH}	0,7 x V_{DD}	–	–	V	
Input voltage LOW	V_{IL}	–	–	0,3 x V_{DD}	V	
Input leakage current at $V_I = V_{SS2}$ to V_{DD}	$\pm I_I$	–	–	1	μA	
Inputs EXTPF, PFIN						
Input voltage HIGH	$V_{IH}-V_{SS1}$	0,7 x $(V_{DD}-V_{SS1})$	–	–	V	
Input voltage LOW	$V_{IL}-V_{SS1}$	0	–	0,3 x $(V_{DD}-V_{SS1})$	V	
Input leakage current						
at $V_I = V_{SS1}$ to V_{DD}	$\pm I_I$	–	–	1	μA	
at T_{amb} = 25 °C; $V_I = V_{SS1}$ to V_{DD}	$\pm I_I$	–	–	0,1	μA	
Outputs SEC, MIN, COMP, FSET (normal buffer outputs)						
Output voltage HIGH						
at $V_{DD}-V_{SS2}$ = 2,5 V; $-I_O$ = 0,1 mA	V_{OH}	$V_{DD}-0,4$	–	–	V	
at $V_{DD}-V_{SS2}$ = 4 to 6 V; $-I_O$ = 0,5 mA	V_{OH}	$V_{DD}-0,4$	–	–	V	
Output voltage LOW						
at $V_{DD}-V_{SS2}$ = 2,5 V; I_O = 0,3 mA	V_{OL}	–	–	0,4	V	
at $V_{DD}-V_{SS2}$ = 4 to 6 V; I_O = 1,6 mA	V_{OL}	–	–	0,4	V	

CHARACTERISTICS (continued)

parameter	symbol	min.	typ.	max.	unit
Output SDA (N-channel open drain)					
Output "ON": I_O = 3 mA at $V_{DD}-V_{SS2}$ = 2,5 to 6 V	V_{OL}	–	–	0,4	V
Output "OFF" (leakage current) at $V_{DD}-V_{SS2}$ = 6 V; V_O = 6 V	I_O	–	–	1	μA
Internal threshold voltage					
Power failure detection	V_{TH1}	1	1,2	1,4	V
Power "ON" reset at $V_{SCL} = V_{SDA} = V_{DD}$	V_{TH2}	1,5	2,0	2,5	V
Rise and fall times of input signals					
Input EXTPF	t_r, t_f	–	–	1	μs
Input PFIN	t_r, t_f	–	–	∞	μs
Input signals except EXTPF and PFIN between V_{IL} and V_{IH} levels					
rise time	t_r	–	–	1	μs
fall time	t_f	–	–	0,3	μs
Frequency at SCL at $V_{DD}-V_{SS2}$ = 4 to 6 V					
Pulse width LOW (see Figs 7 and 9)	t_{LOW}	4,7	–	–	μs
Pulse width HIGH (see Figs 7 and 9)	t_{HIGH}	4	–	–	μs
Noise suppression time constant at SCL and SDA input	T_I	0,25	1	2,5	μs
Input capacitance (SCL, SDA)	C_I	–	–	7	pF
Oscillator					
Integrated oscillator capacitance	C_{out}	–	40	–	pF
Oscillator feedback resistance	R_f	–	3	–	MΩ
Oscillator stability for: $\Delta(V_{DD}-V_{SS1})$ = 100 mV at $V_{DD}-V_{SS1}$ = 1,55 V; T_{amb} = 25 °C	f/f_{osc}	–	2×10^{-6}	–	–
Quartz crystal parameters					
Frequency = 32,768 kHz					
Series resistance	R_S	–	–	40	kΩ
Parallel capacitance	C_L	–	9	–	pF
Trimmer capacitance	C_T	5	–	25	pF

Purchase of Philips' I²C components conveys a license under the Philips' I²C patent to use the components in the I²C-system provided the system conforms to the I²C specifications defined by Philips.

APPLICATION INFORMATION

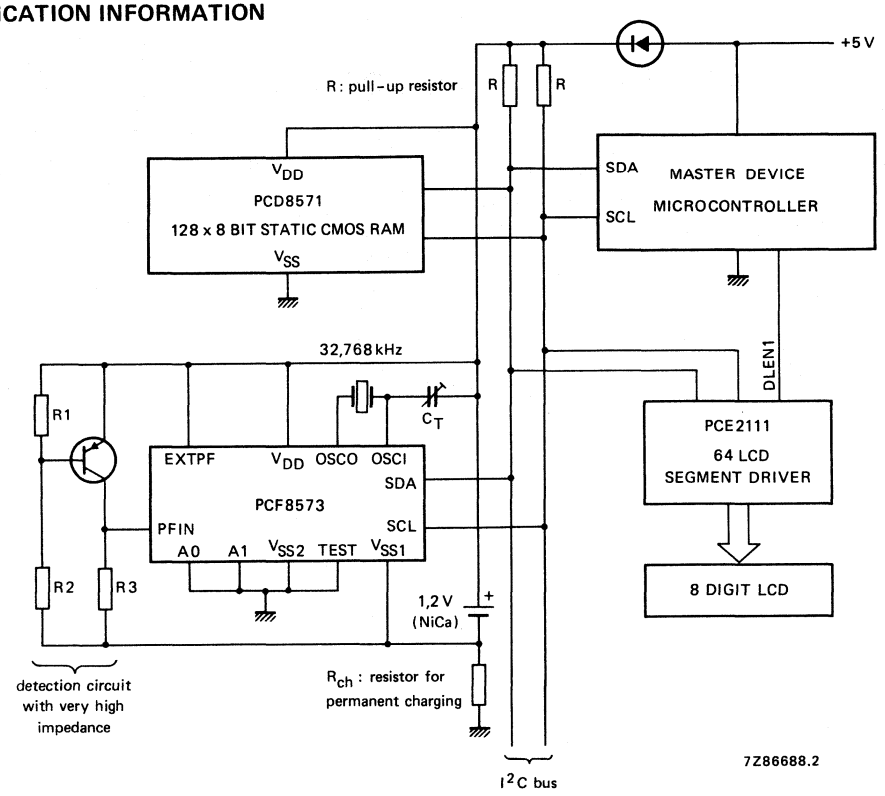


Fig. 15 Application example of the PCF8573 clock/calendar.

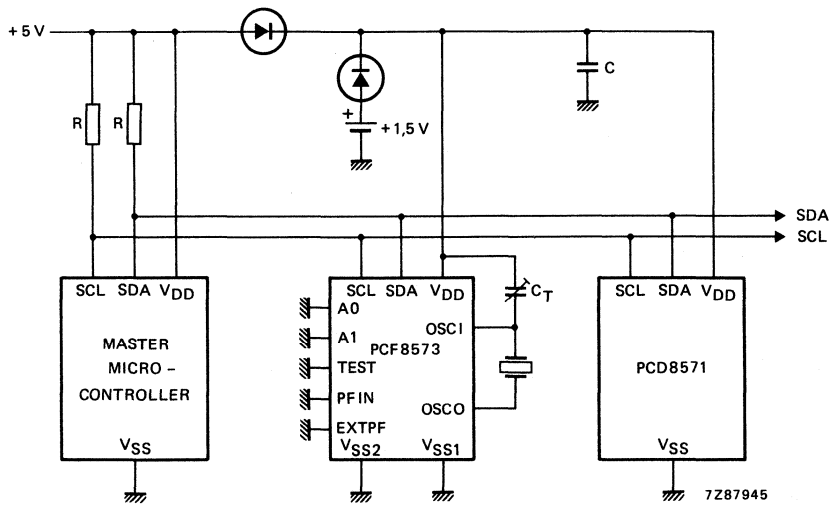


Fig. 16 Application example of the PCF8573 with common V_{SS1} and V_{SS2} supply.

REMOTE 8-BIT I/O FOR I²C BUS

GENERAL DESCRIPTION

The PCF8574 is a single-chip silicon gate CMOS circuit. It provides remote I/O expansion for the MAB8400 and PCF8500 microcomputer families via the two-line serial bidirectional bus (I^2C).
It can also interface microcomputers without a serial interface to the I^2C bus (as a slave function only).
The device consists of an 8-bit quasi-bidirectional port and an I^2C interface.
The PCF8574 has low current consumption and includes latched outputs with high current drive capability for directly driving LEDs. It also possesses an interrupt line (INT) which is connected to the interrupt logic of the microcomputer on the I^2C bus. By sending an interrupt signal on this line, the remote I/O can inform the microcomputer if there is incoming data on its ports without having to communicate via the I^2C bus. This means that the PCF8574 can remain a simple slave device.

Features

- Operating supply voltage 2,5 V to 6 V
- Low stand-by current consumption max. 10 μA
- Bidirectional expander
- Open drain interrupt output
- 8-bit remote I/O port for the I^2C bus
- Peripheral for the MAB8400 and PCF8500 microcomputer families
- Latched outputs with high current drive capability for directly driving LEDs
- Address by 3 hardware address pins for use of up to 8 devices (up to 16 possible with mask option)

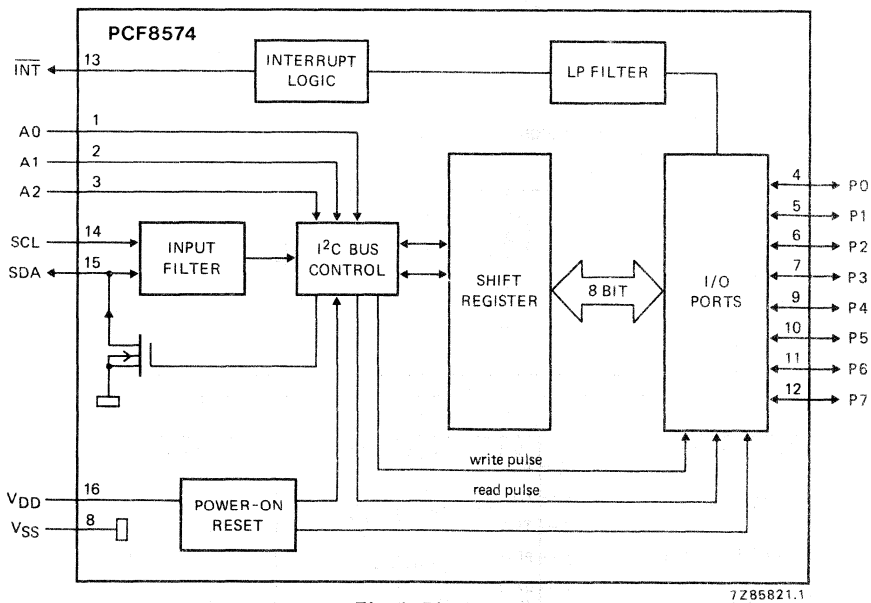


Fig. 1 Block diagram.

PACKAGE OUTLINES

PCF8574P: 16-lead DIL; plastic (SOT-38).
PCF8574T: 16-lead mini-pack; plastic (SO-16L; SOT-162A).

PINNING

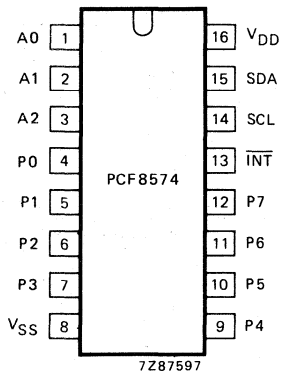


Fig. 2 Pinning diagram.

1 to 3	A0 to A2	address inputs
4 to 7 9 to 12	P0 to P3 P4 to P7	8-bit quasi-bidirectional I/O port
8	V_{SS}	negative supply
13	$\overline{INT}$	interrupt output
14	SCL	serial clock line
15	SDA	serial data line
16	V_{DD}	positive supply

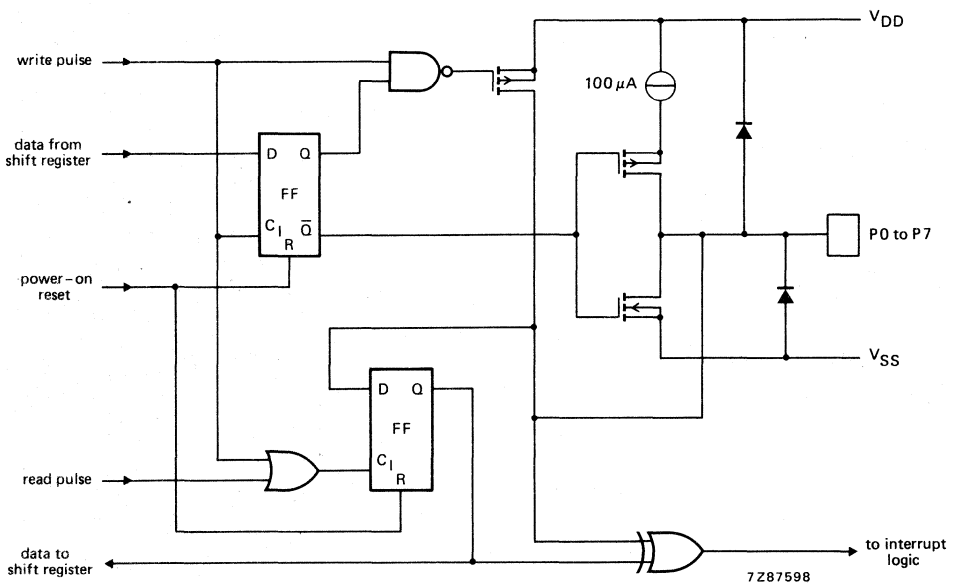


Fig. 3 Simplified schematic diagram of each port.

CHARACTERISTICS OF THE I²C BUS

The I²C bus is for 2-way, 2-line communication between different ICs or modules. The two lines are a serial data line (SDA) and a serial clock line (SCL). Both lines must be connected to a positive supply via a pull-up resistor when connected to the output stages of a device. Data transfer may be initiated only when the bus is not busy.

Bit transfer

One data bit is transferred during each clock pulse. The data on the SDA line must remain stable during the HIGH period of the clock pulse as changes in the data line at this time will be interpreted as control signals.

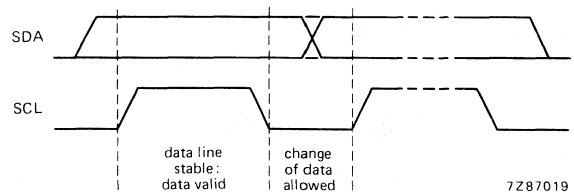


Fig. 4 Bit transfer.

Start and stop conditions

Both data and clock lines remain HIGH when the bus is not busy. A HIGH-to-LOW transition of the data line, while the clock is HIGH is defined as the start condition (S). A LOW-to-HIGH transition of the data line while the clock is HIGH is defined as the stop condition (P).

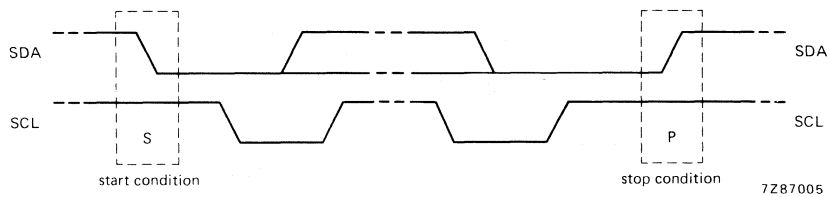


Fig. 5 Definition of start and stop conditions.

System configuration

A device generating a message is a "transmitter", a device receiving a message is the "receiver". The device that controls the message is the "master" and the devices which are controlled by the master are the "slaves".

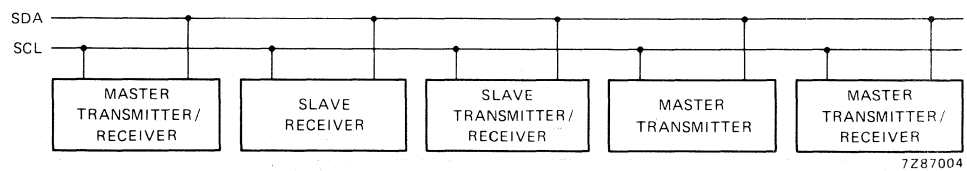


Fig. 6 System configuration.

CHARACTERISTICS OF THE I²C BUS (continued)

Acknowledge

The number of data bytes transferred between the start and stop conditions from transmitter to receiver is not limited. Each byte of eight bits is followed by one acknowledge bit. The acknowledge bit is a HIGH level put on the bus by the transmitter whereas the master generates an extra acknowledge related clock pulse. A slave receiver which is addressed must generate an acknowledge after the reception of each byte. Also a master must generate an acknowledge after the reception of each byte that has been clocked out of the slave transmitter. The device that acknowledges has to pull down the SDA line during the acknowledge clock pulse, so that the SDA line is stable LOW during the HIGH period of the acknowledge related clock pulse, set up and hold times must be taken into account. A master receiver must signal an end of data to the transmitter by *not* generating an acknowledge on the last byte that has been clocked out of the slave. In this event the transmitter must leave the data line HIGH to enable the master to generate a stop condition.

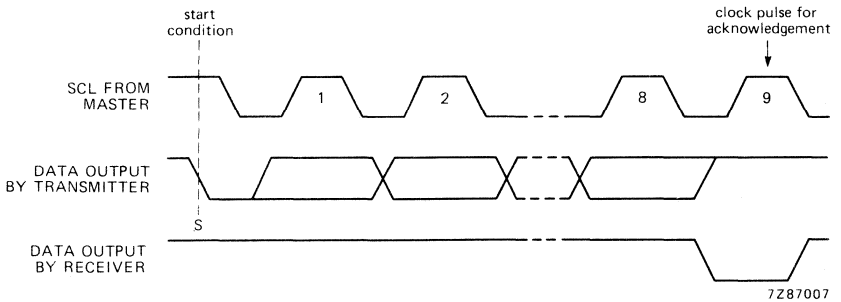


Fig. 7 Acknowledgement on the I²C bus.

Timing specifications

Within the I²C bus specifications a high-speed mode and a low-speed mode are defined. The PCF8574 operates in both modes and the timing requirements are as follows:

High-speed mode

Masters generate a bus clock with a maximum frequency of 100 kHz. Detailed timing is shown in Fig. 8.

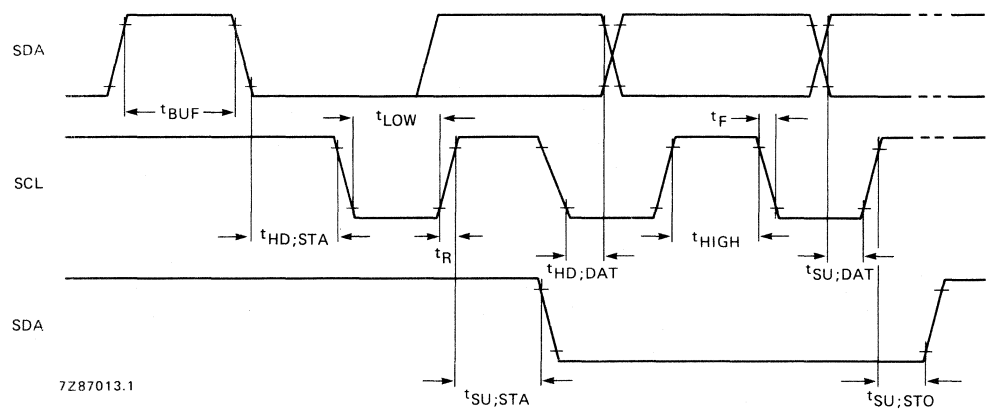


Fig. 8 Timing of the high-speed mode.

Where:

t_{BUF}	$t \geqslant t_{LOWmin}$	The minimum time the bus must be free before a new transmission can start
$t_{HD;STA}$	$t \geqslant t_{HIGHmin}$	Start condition hold time
t_{LOWmin}	4,7 μs	Clock LOW period
$t_{HIGHmin}$	4 μs	Clock HIGH period
$t_{SU;STA}$	$t \geqslant t_{LOWmin}$	Start condition set-up time, only valid for repeated start code
$t_{HD;DAT}$	$t \geqslant 0$ μs	Data hold time
$t_{SU;DAT}$	$t \geqslant 250$ ns	Data set-up time
t_R	$t \leqslant 1$ μs	Rise time of both the SDA and SCL line
t_F	$t \leqslant 300$ ns	Fall time of both the SDA and SCL line
$t_{SU;STO}$	$t \geqslant t_{LOWmin}$	Stop condition set-up time

Note

All the values refer to V_{IH} and V_{IL} levels with a voltage swing of V_{SS} to V_{DD}.

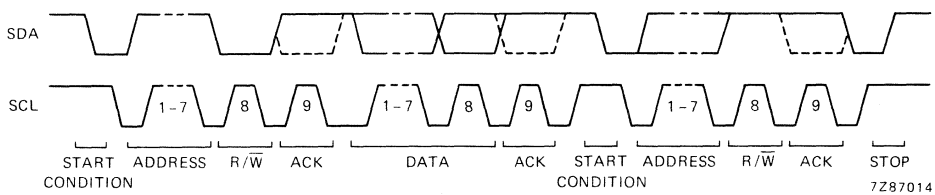


Fig. 9 Complete data transfer in the high-speed mode.

Where:

Clock t_{LOWmin}	4,7 μs
$t_{HIGHmin}$	4 μs
The dashed line is the acknowledgement of the receiver	
Mark-to-space ratio	1 : 1 (LOW-to-HIGH)
Max. number of bytes	unrestricted
Premature termination of transfer	allowed by generation of STOP condition
Acknowledge clock bit	must be provided by the master

CHARACTERISTICS OF THE I²C BUS (continued)

Low-speed mode

Masters generate a bus clock with a maximum frequency of 2 kHz; a minimum LOW period of 105 μs and a minimum HIGH period of 365 μs. The mark-to-space ratio is 1 : 3 LOW-to-HIGH. Detailed timing is shown in Fig. 10.

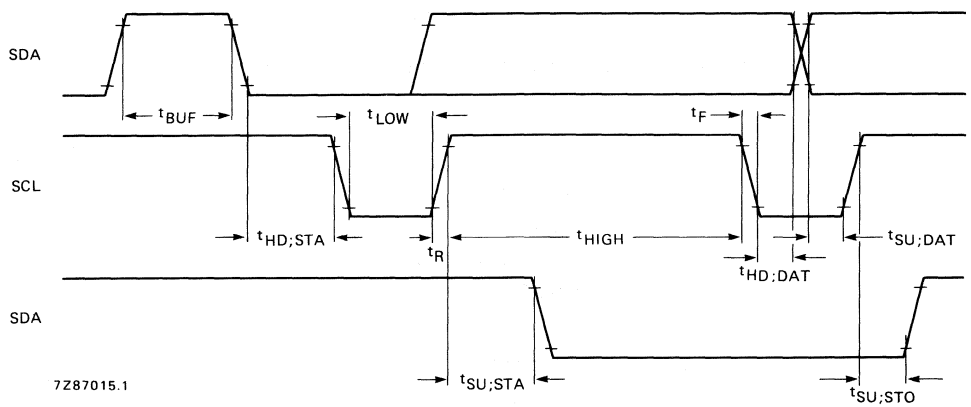


Fig. 10 Timing of the low-speed mode.

Where:

t_{BUF}	$t \geqslant 105\ \mu s$ (t_{LOWmin})
$t_{HD;STA}$	$t \geqslant 365\ \mu s$ ($t_{HIGHmin}$)
t_{LOW}	130 μs ± 25 μs
t_{HIGH}	390 μs ± 25 μs
$t_{SU;STA}$	130 μs ± 25 μs*
$t_{HD;DAT}$	$t \geqslant 0\ \mu s$
$t_{SU;DAT}$	$t \geqslant 250$ ns
t_R	$t \leqslant 1\ \mu s$
t_F	$t \leqslant 300$ ns
$t_{SU;STO}$	130 μs ± 25 μs

Note

All the values refer to V_{IH} and V_{IL} levels with a voltage swing of V_{SS} to V_{DD}, for definitions see high-speed mode.

* Only valid for repeated start code.

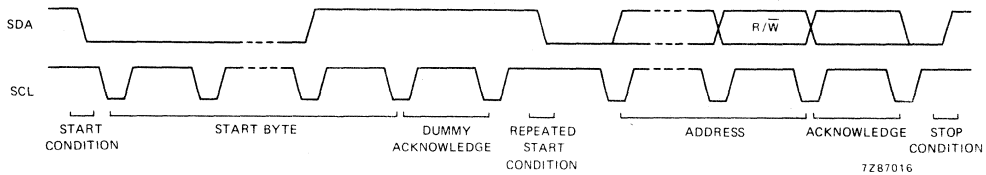


Fig. 11 Complete data transfer in the low-speed mode.

Where:	
Clock t_{LOWmin}	130 μs ± 25 μs
$t_{HIGHmin}$	390 μs ± 25 μs
Mark-to-space ratio	1 : 3 (LOW-to-HIGH)
Start byte	0000 0001
Maximum number of bytes	6
Premature termination of transfer	not allowed
Acknowledge clock bit	must be provided by master

Note

The general characteristics and detailed specification of the I²C bus are described in a separate data sheet (serial data buses) in handbook: ICs for digital systems in radio, audio and video equipment.

FUNCTIONAL DESCRIPTION

Addressing (see Figs 12 and 13)

Each bit of the PCF8574 I/O port can be independently used as an input or an output. Input data is transferred from the port to the microcomputer by the READ mode. Output data is transmitted to the port by the WRITE mode.

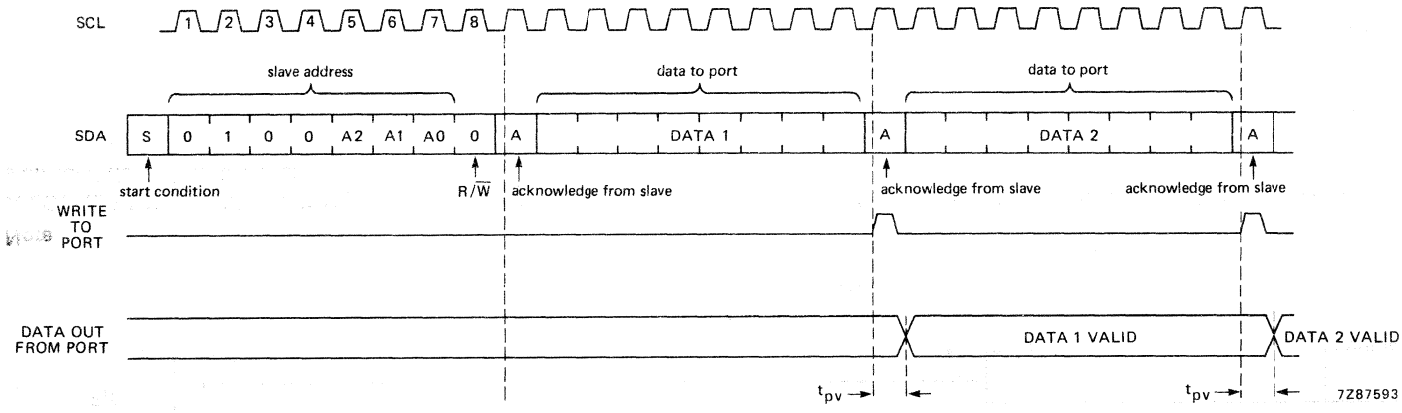


Fig. 12 WRITE mode (output port).

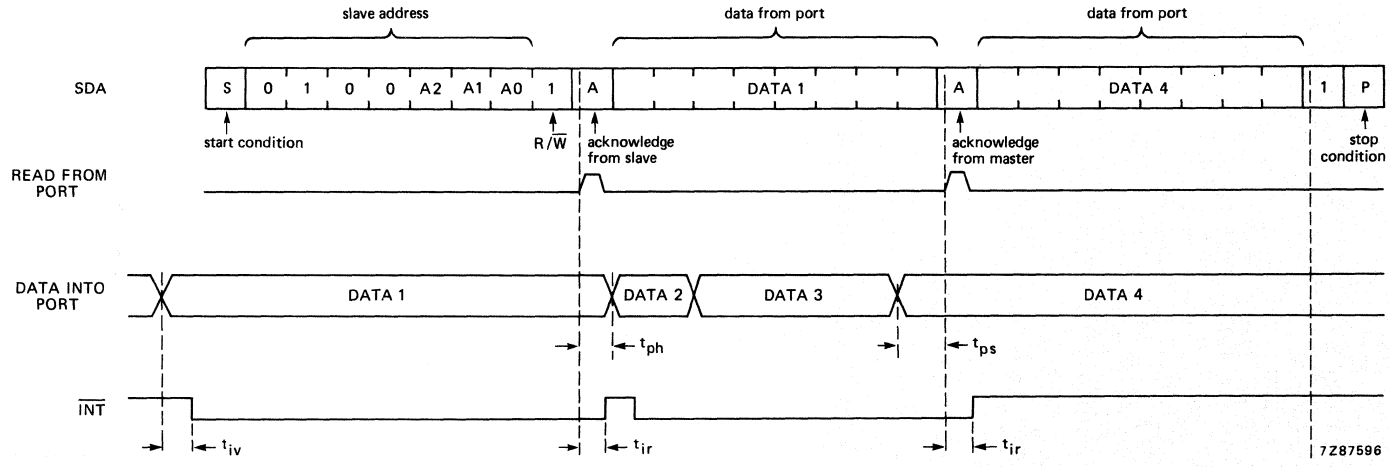


Fig. 13 READ mode (input port).

Note

A LOW-to-HIGH transition of SDA, while SCL is HIGH is defined as the stop condition (P). Transfer of data can be stopped at any moment by a stop condition. When this occurs, data present at the last acknowledge phase is valid (output mode). Input data is lost.

Interrupt (see Figs 14 and 15)

The PCF8574 provides an open drain output ($\overline{INT}$) which can be fed to a corresponding input of the microcomputer. This gives these chips a type of master function which can initiate an action elsewhere in the system.

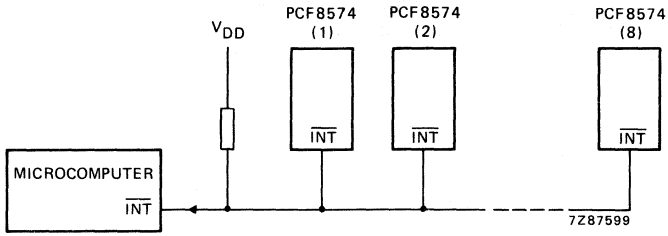


Fig. 14 Application of multiple PCF8574s with interrupt.

An interrupt is generated by any rising or falling edge of the port inputs in the input mode. After time t_{iv} the signal $\overline{INT}$ is valid.
Resetting and reactivating the interrupt circuit is achieved when data on the port is changed to the original setting or data is read from or written to the port which has generated the interrupt.
Resetting occurs as follows:

- In the READ mode at the acknowledge bit after the rising edge of the SCL signal.
- In the WRITE mode at the acknowledge bit after the HIGH to LOW transition of the SCL signal.

Each change of the ports after the resettings will be detected and after the next rising clock edge, will be transmitted as $\overline{INT}$.
Reading from or writing to another device does not affect the interrupt circuit.

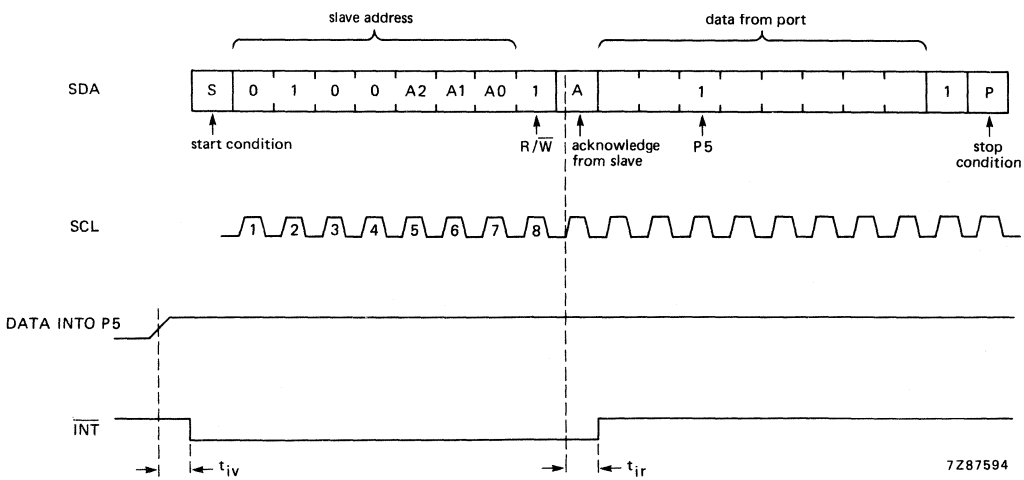


Fig. 15 Interrupt generated by a change of input to port P5.

FUNCTIONAL DESCRIPTION (continued)

Quasi-bidirectional I/O ports (see Fig. 16)

A quasi-bidirectional port can be used as an input or output without the use of a control signal for data direction. The bit designated as an input must first be loaded with a logic 1. In this mode only a current source to V_{DD} is active. An additional strong pull-up to V_{DD} allows fast rising edges into heavily loaded outputs. These devices turn on when an output changes from LOW to HIGH, and are switched off by the negative edge of SCL. SCL should not remain HIGH when a short-circuit to V_{SS} is allowed (input mode).

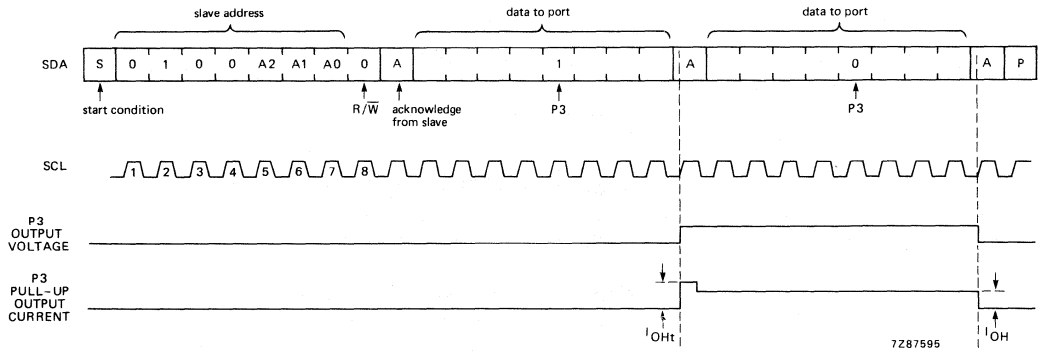


Fig. 16 Transient pull-up current I_{OHt} while P3 changes from LOW-to-HIGH and back to LOW.

RATINGS

Limiting values in accordance with the Absolute Maximum System (IEC 134)

Supply voltage range	V_{DD}		−0,5 to + 7	V
Input voltage range (any pin)	V_I		V_{SS}−0,5 to V_{DD} + 0,5	V
D.C. current into any input	$\pm I_I$	max.	20	mA
D.C. current into any output	$\pm I_O$	max.	25	mA
V_{DD} or V_{SS} current	$\pm I_{DD}$; I_{SS}	max.	100	mA
Total power dissipation	P_{tot}	max.	400	mW
Power dissipation per output	P_o	max.	100	mW
Storage temperature range	T_{stg}		−65 to + 150	°C
Operating ambient temperature range	T_{amb}		−40 to + 85	°C

CHARACTERISTICS

V_{DD} = 2,5 to 6 V; V_{SS} = 0 V; T_{amb} = −40 to + 85 °C unless otherwise specified

parameter	symbol	min.	typ.	max.	unit
Supply (pin 16)					
Supply voltage	V_{DD}	2,5	–	6	V
Supply current					
at V_{DD} = 6 V; no load, inputs at V_{DD}, V_{SS}					
operating; (SCL = 100 kHz)	I_{DD}	–	40	100	μA
standby	I_{DDO}	–	1,5	10	μA
Power-on reset voltage level (note 1)	V_{REF}	–	1,3	2,4	V
Input SCL; input/output SDA (pins 14; 15)					
Input voltage LOW	V_{IL}	−0,5 V	–	$0{,}3V_{DD}$	V
Input voltage HIGH	V_{IH}	$0{,}7V_{DD}$	–	V_{DD} + 0,5 V	V
Output current LOW					
at V_{OL} = 0,4 V	I_{OL}	3	–	–	mA
Input/Output leakage current	$\|I_L\|$	–	–	100	nA
Clock frequency (see Fig. 8)	f_{SCL}	–	–	100	kHz
Tolerable spike width					
at SCL and SDA input	t_s	–	–	100	ns
Input capacitance (SCL, SDA)					
at $V_I = V_{SS}$	C_I	–	–	7	pF
I/O ports (pins 4 to 7; 9 to 12)					
Input voltage LOW	V_{IL}	−0,5 V	–	$0{,}3V_{DD}$	V
Input voltage HIGH	V_{IH}	$0{,}7V_{DD}$	–	V_{DD} + 0,5 V	V
Maximum allowed input current through protection diode					
at $V_I \geqslant V_{DD}$ or $\leqslant V_{SS}$	$\pm I_{IHL}$	–	–	400	μA
Output current LOW					
at V_{OL} = 1 V; V_{DD} = 5 V	I_{OL}	10	30	–	mA
Output current HIGH					
at $V_{OH} = V_{SS}$ (current source only)	$-I_{OH}$	30	100	300	μA
Transient pull-up current HIGH during acknowledge (see Fig. 16)					
at $V_{OH} = V_{SS}$	$-I_{OHt}$	–	0,5	–	mA
Input/Output capacitance	$C_{I/O}$	–	–	10	pF
Port timing; $C_L \leqslant 100$ pF (see Figs 12 and 13)					
Output data valid	t_{pv}	–	–	4	μs
Input data set-up	t_{ps}	0	–	–	μs
Input data hold	t_{ph}	4	–	–	μs

parameter	symbol	min.	typ.	max.	unit
Interrupt $\overline{INT}$ (pin 13)					
Output current LOW at V_{OL} = 0,4 V	I_{OL}	1,6	–	–	mA
Output current HIGH at $V_{OH} = V_{DD}$	$\|I_{OH}\|$	–	–	100	nA
$\overline{INT}$ timing; $C_L \leqslant 100$ pF (see Fig. 13)					
Input data valid	t_{iv}	–	–	4	μs
Reset delay	t_{ir}	–	–	4	μs
Select inputs A0, A1, A2 (pins 1 to 3)					
Input voltage LOW	V_{IH}	−0,5 V	–	$0{,}3V_{DD}$	V
Input voltage HIGH	V_{IH}	$0{,}7V_{DD}$	–	V_{DD} + 0,5 V	V
Input leakage current at $V_I = V_{DD}$ or V_{SS}	$\|I_L\|$	–	–	100	nA

Note 1

The power-on reset circuit resets the I²C bus logic with $V_{DD} < V_{REF}$ and sets all ports to logic 1 (input mode with current source to V_{DD}).

Purchase of Philips' I²C components conveys a license under the Philips' I²C patent to use the components in the I²C-system provided the system conforms to the I²C specifications defined by Philips.

UNIVERSAL LCD DRIVER FOR LOW MULTIPLEX RATES

GENERAL DESCRIPTION

The PCF8576 is a peripheral device which interfaces to almost any liquid crystal display (LCD) having low multiplex rates. It generates the drive signals for any static or multiplexed LCD containing up to four backplanes and up to 40 segments and can easily be cascaded for larger LCD applications. The PCF8576 is compatible with most microprocessors and communicates via a two-line bidirectional bus (I^2C). Communication overheads are minimized by a display RAM with auto-incremented addressing, by hardware subaddressing and by display memory switching (static and duplex drive modes).

Features

- Single-chip LCD controller/driver
- Selectable backplane drive configuration: static or 2/3/4 backplane multiplexing
- Selectable display bias configuration: static, 1/2 or 1/3
- Internal LCD bias generation with voltage-follower buffers
- 40 segment drives: up to twenty 8-segment numeric characters; up to ten 15-segment alphanumeric characters; or any graphics of up to 160 elements
- 40 x 4-bit RAM for display data storage
- Auto-incremented display data loading across device subaddress boundaries
- Display memory bank switching in static and duplex drive modes
- Versatile blinking modes
- LCD and logic supplies may be separated
- Wide power supply range: from 2 V for low-threshold LCDs and up to 9 V for guest-host LCDs and high-threshold (automobile) twisted nematic LCDs
- Low power consumption
- Power-saving mode for extremely low power consumption in battery-operated and telephone applications
- I^2C bus interface
- TTL/CMOS compatible
- Compatible with any 4-bit, 8-bit or 16-bit microprocessors
- May be cascaded for large LCD applications (up to 2560 segments possible)
- Optimized pinning for single plane wiring in both single and multiple PCF8576 applications
- Space-saving 56-lead plastic mini-pack (VSO-56)
- Very low external component count (at most one resistor, even in multiple device applications)
- Compatible with chip-on-glass technology
- Manufactured in silicon gate CMOS process

PACKAGE OUTLINES

PCF8576T: 56-lead mini-pack; plastic (VSO-56; SOT-190).
PCF8576U: uncased chip in tray

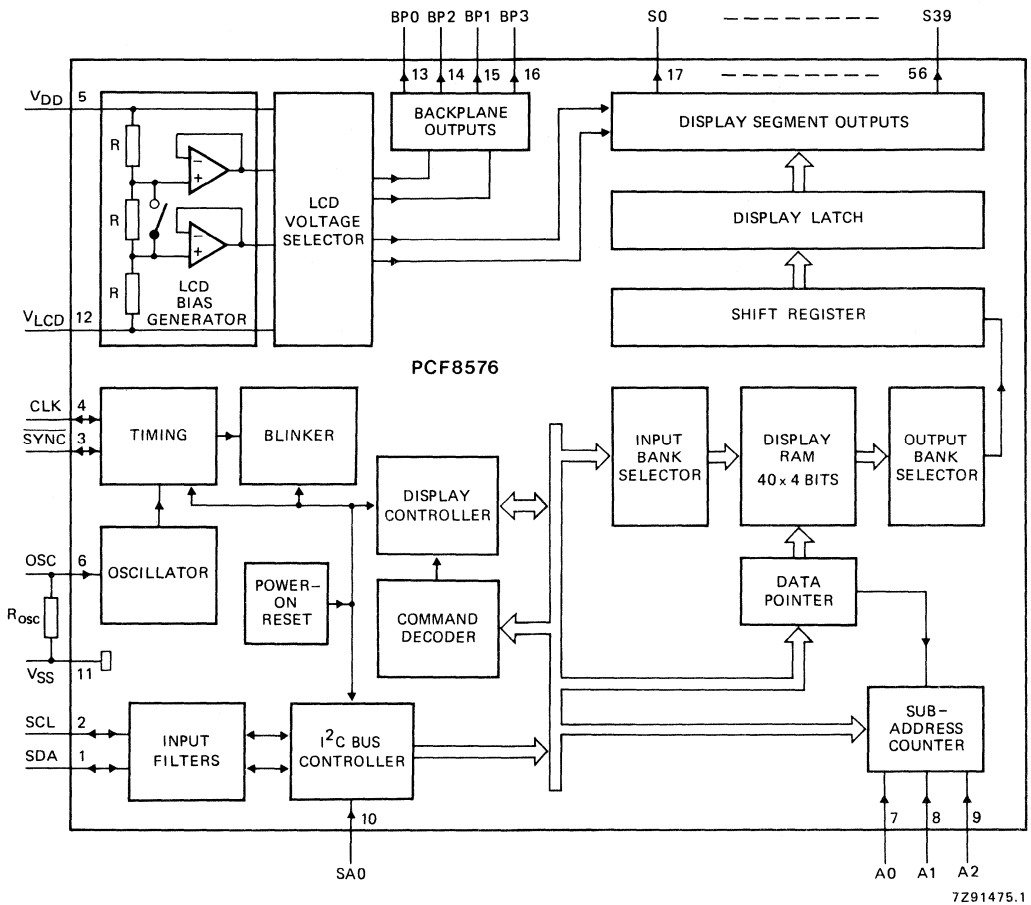


Fig. 1 Block diagram.

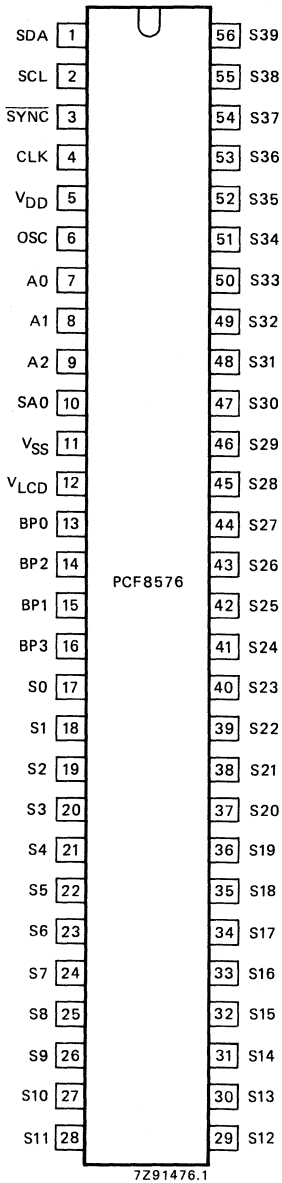


Fig. 2 Pinning diagram.

PINNING

1	SDA	I^2C bus data input/output
2	SCL	I^2C bus clock input/output
3	$\overline{SYNC}$	cascade synchronization input/output
4	CLK	external clock input/output
5	V_{DD}	positive supply voltage
6	OSC	oscillator input
7	A0	I^2C bus subaddress inputs
8	A1	
9	A2	
10	SA0	I^2C bus slave address bit 0 input
11	V_{SS}	logic ground
12	V_{LCD}	LCD supply voltage
13	BP0	LCD backplane outputs
14	BP2	
15	BP1	
16	BP3	
17 to 56	S0 to S39	LCD segment outputs

FUNCTIONAL DESCRIPTION

The PCF8576 is a versatile peripheral device designed to interface any microprocessor to a wide variety of LCDs. It can directly drive any static or multiplexed LCD containing up to four backplanes and up to 40 segments. The display configurations possible with the PCF8576 depend on the number of active backplane outputs required; a selection of display configurations is given in Table 1.

Table 1 Selection of display configurations

active back-plane outputs	no. of segments	7-segment numeric	14-segment alphanumeric	dot matrix
4	160	20 digits + 20 indicator symbols	10 characters + 20 indicator symbols	160 dots (4 x 40)
3	120	15 digits + 15 indicator symbols	8 characters + 8 indicator symbols	120 dots (3 x 40)
2	80	10 digits + 10 indicator symbols	5 characters + 10 indicator symbols	80 dots (2 x 40)
1	40	5 digits + 5 indicator symbols	2 characters + 12 indicator symbols	40 dots

All of the display configurations given in Table 1 can be implemented in the typical system shown in Fig. 3. The host microprocessor maintains the 2-line I²C bus communication channel with the PCF8576. A resistor connected between OSC (pin 6) and V_{SS} (pin 11) controls the device clock frequency. The appropriate biasing voltages for the multiplexed LCD waveforms are generated internally. The only other connections required to complete the system are to the power supplies (V_{DD}, V_{SS} and V_{LCD}) and to the LCD panel chosen for the application.

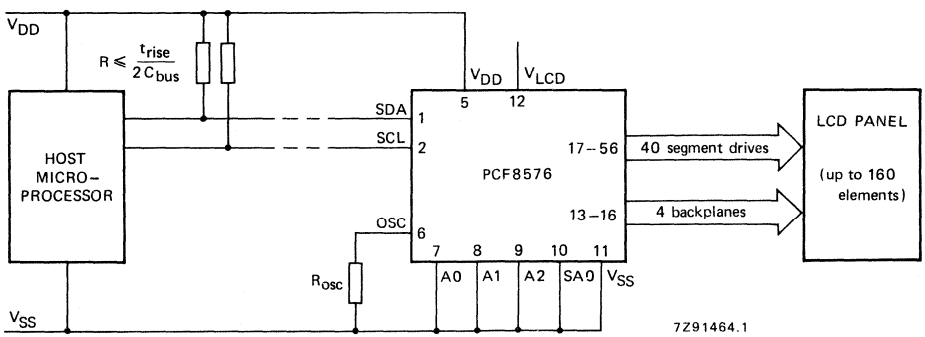


Fig. 3 Typical system configuration.

Power-on reset

At power-on the PCF8576 resets to a defined starting condition as follows:

1. All backplane outputs are set to V_{DD}.
2. All segment outputs are set to V_{DD}.
3. The drive mode '1 : 4 multiplex with 1/3 bias' is selected.
4. Blinking is switched off.
5. Input and output bank selectors are reset (as defined in Table 5).
6. The I^2C bus interface is initialized.
7. The data pointer and the subaddress counter are cleared.

Data transfers on the I^2C bus should be avoided for 1 ms following power-on to allow completion of the reset action.

LCD bias generator

The full-scale LCD voltage (V_{op}) is obtained from $V_{DD} - V_{LCD}$. The LCD voltage may be temperature compensated externally through the V_{LCD} supply to pin 12. Fractional LCD biasing voltages are obtained from an internal voltage divider of three series resistors connected between V_{DD} and V_{LCD}. The centre resistor can be switched out of circuit to provide a ½ bias voltage level for the 1 : 2 multiplex configuration.

LCD voltage selector

The LCD voltage selector coordinates the multiplexing of the LCD according to the selected LCD drive configuration. The operation of the voltage selector is controlled by MODE SET commands from the command decoder. The biasing configurations that apply to the preferred modes of operation, together with the biasing characteristics as functions of $V_{op} = V_{DD} - V_{LCD}$ and the resulting discrimination ratios (D), are given in Table 2.

Table 2 Preferred LCD drive modes: summary of characteristics

LCD drive mode	LCD bias configuration	$\frac{V_{off(rms)}}{V_{op}}$	$\frac{V_{on\ (rms)}}{V_{op}}$	$D = \frac{V_{on(rms)}}{V_{off(rms)}}$
static (1 BP)	static (2 levels)	0	1	∞
1 : 2 MUX (2 BP)	1/2 (3 levels)	$\sqrt{2}/4 = 0{,}354$	$\sqrt{10}/4 = 0{,}791$	$\sqrt{5} = 2{,}236$
1 : 2 MUX (2 BP)	1/3 (4 levels)	$1/3 = 0{,}333$	$\sqrt{5}/3 = 0{,}745$	$\sqrt{5} = 2{,}236$
1 : 3 MUX (3 BP)	1/3 (4 levels)	$1/3 = 0{,}333$	$\sqrt{33}/9 = 0{,}638$	$\sqrt{33}/3 = 1{,}915$
1 : 4 MUX (4 BP)	1/3 (4 levels)	$1/3 = 0{,}333$	$\sqrt{3}/3 = 0{,}577$	$\sqrt{3} = 1{,}732$

LCD voltage selector (continued)

A practical value for V_{op} is determined by equating $V_{off(rms)}$ with a defined LCD threshold voltage (V_{thLCD}), typically when the LCD exhibits approximately 10% contrast. In the static drive mode a suitable choice is $V_{op} \gtrapprox 3\,V_{thLCD}$.
Multiplex drive ratios of 1 : 3 and 1 : 4 with 1/2 bias are possible but the discrimination and hence the contrast ratios are smaller ($\sqrt{3} = 1{,}732$ for 1 : 3 multiplex or $\sqrt{21}/3 = 1{,}528$ for 1 : 4 multiplex).
The advantage of these modes is a reduction of the LCD full scale voltage V_{op} as follows:

1 : 3 multiplex (1/2 bias) : $V_{op} = \sqrt{6}\,V_{off(rms)} = 2{,}449\,V_{off(rms)}$

1 : 4 multiplex (1/2 bias) : $V_{op} = 4\sqrt{3}/3\;V_{off(rms)} = 2{,}309\,V_{off(rms)}$

These compare with $V_{op} = 3\,V_{off(rms)}$ when 1/3 bias is used.

LCD drive mode waveforms

The static LCD drive mode is used when a single backplane is provided in the LCD. Backplane and segment drive waveforms for this mode are shown in Fig. 4.

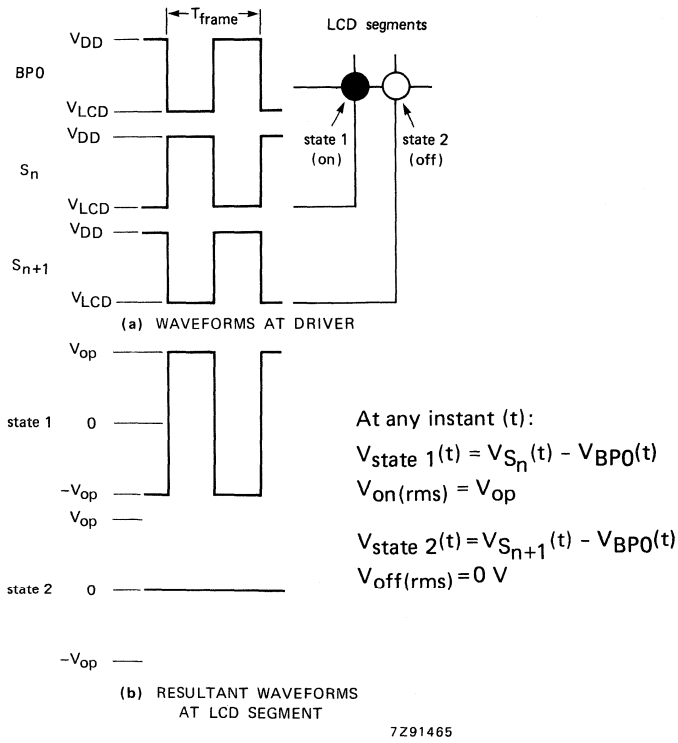


Fig. 4 Static drive mode waveforms: $V_{op} = V_{DD} - V_{LCD}$.

When two backplanes are provided in the LCD the 1 : 2 multiplex drive mode applies. The PCF8576 allows use of 1/2 or 1/3 bias in this mode as shown in Figs 5 and 6.

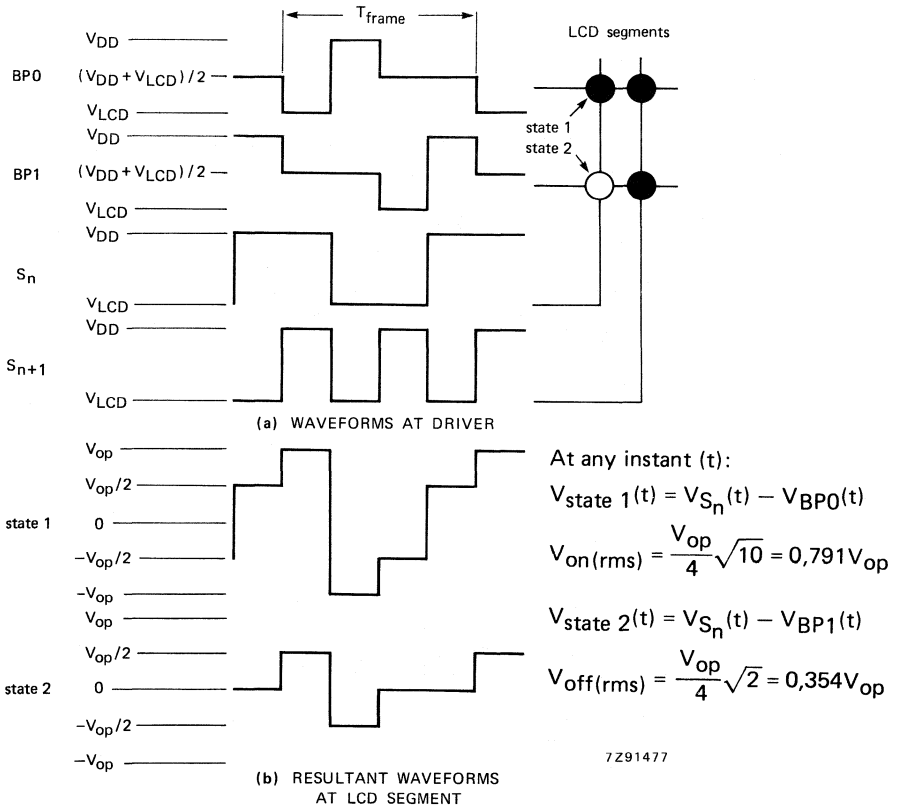


Fig. 5 Waveforms for 1 : 2 multiplex drive mode with 1/2 bias: $V_{op} = V_{DD} - V_{LCD}$.

LCD drive mode waveforms (continued)

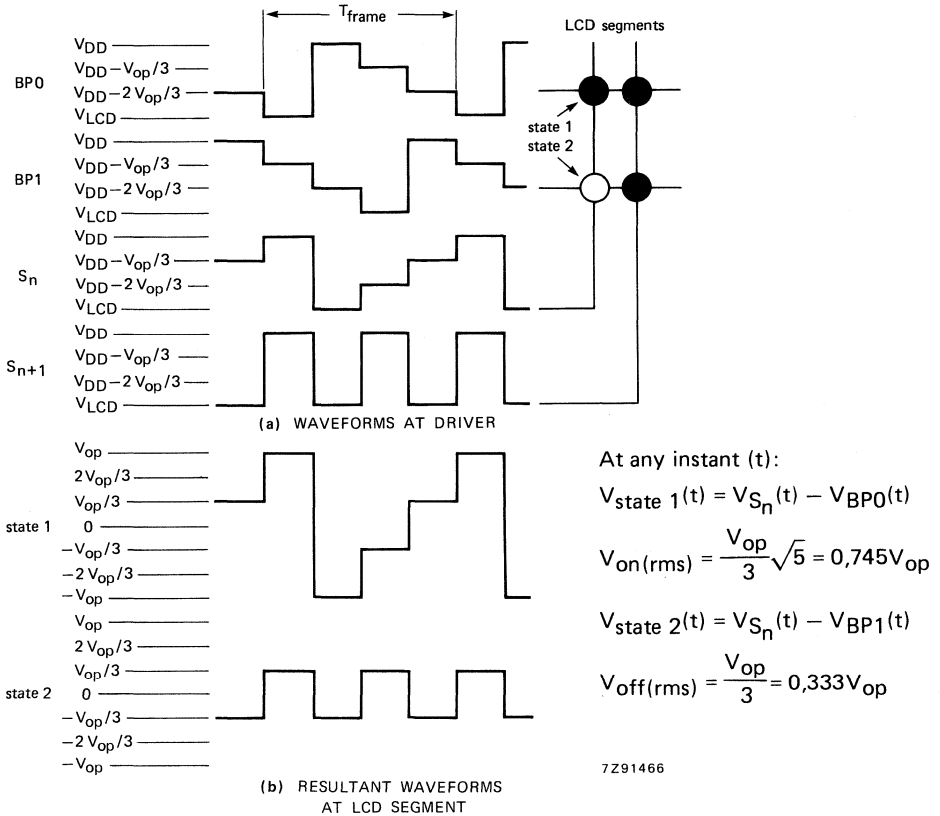


Fig. 6 Waveforms for 1 : 2 multiplex drive mode with 1/3 bias: $V_{op} = V_{DD} - V_{LCD}$.

The backplane and segment drive wavefront for the 1 : 3 multiplex drive mode (three LCD backplanes) and for the 1 : 4 multiplex drive mode (four LCD backplanes) are shown in Figs 7 and 8 respectively.

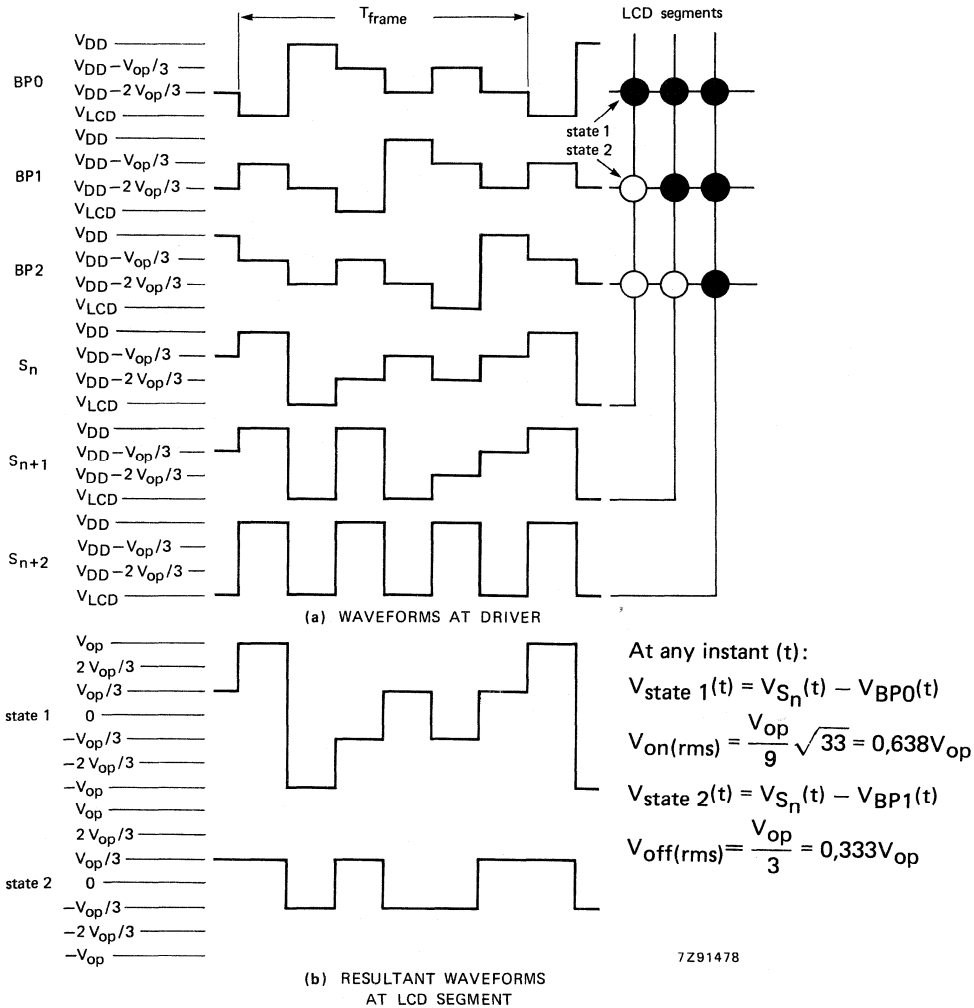


Fig. 7 Waveforms for 1 : 3 multiplex drive mode: $V_{op} = V_{DD} - V_{LCD}$.

LCD drive mode waveforms (continued)

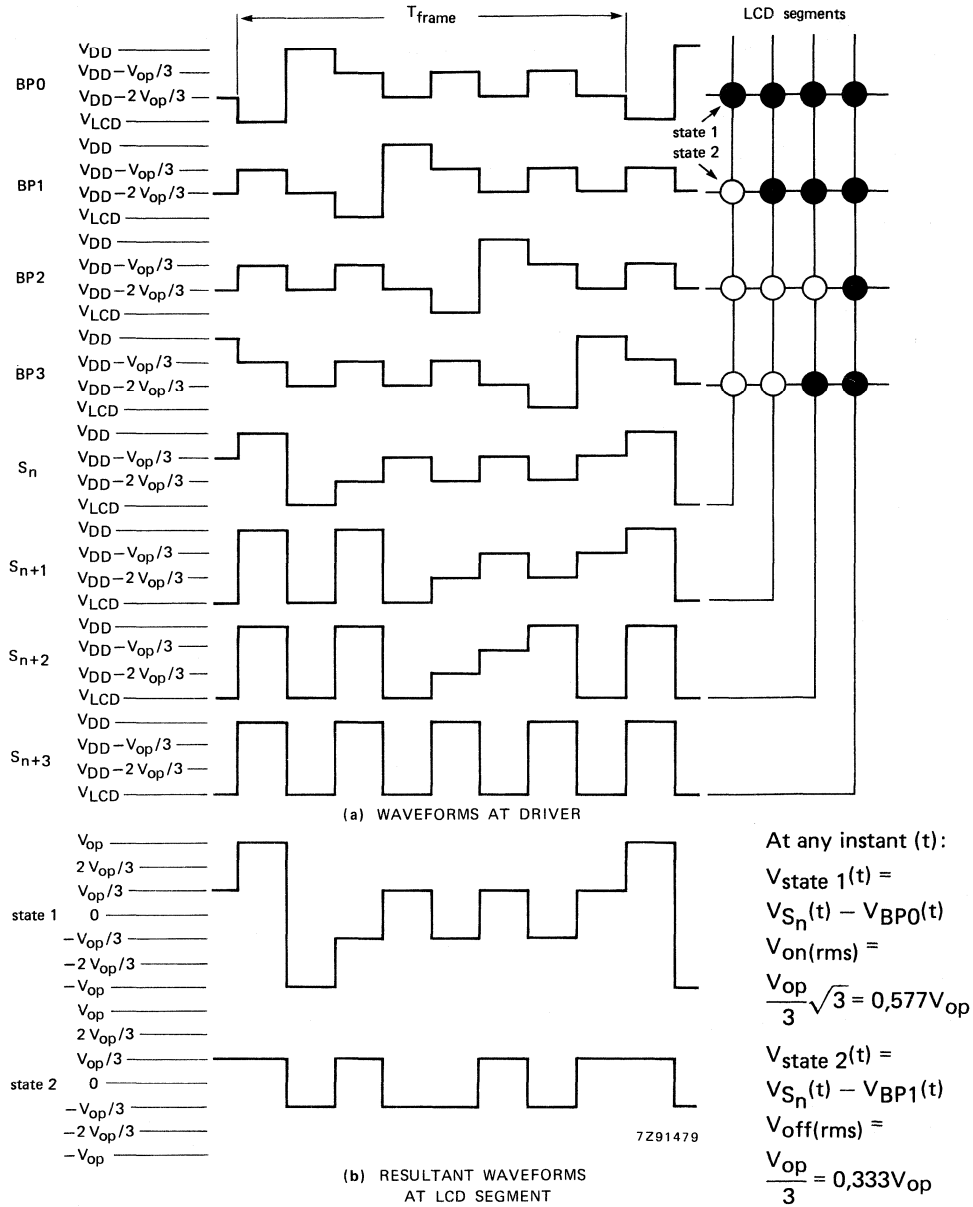


Fig. 8 Waveforms for 1 : 4 multiplex drive mode: $V_{op} = V_{DD} - V_{LCD}$.

Oscillator

Internal clock

The internal logic and the LCD drive signals of the PCF8576 are timed either by the built-in oscillator or from an external clock. When the internal oscillator is used, frequency control is performed by a single resistor connected between OSC (pin 6) and V_{SS} (pin 11) as shown in Fig. 9. In this case, the output from CLK (pin 4) provides the clock signal for cascaded PCF8576s in the system.

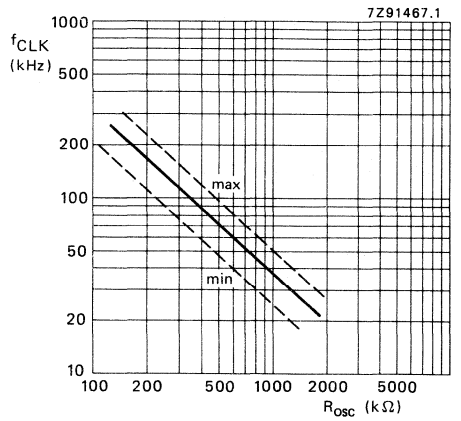


Fig. 9 Oscillator frequency as a function of R_{osc}: $f_{CLK} \approx (3{,}4 \times 10^7 / R_{osc})$ kHz·Ω.

External clock

The condition for external clock is made by tying OSC (pin 6) to V_{DD}; CLK (pin 4) then becomes the external clock input.

The clock frequency (f_{CLK}) determines the LCD frame frequency and the maximum rate for data reception from the I²C bus. To allow I²C bus transmissions at their maximum data rate of 100 kHz, f_{CLK} should be chosen to be above 125 kHz.

A clock signal must always be supplied to the device; removing the clock may freeze the LCD in a d.c. state.

Timing

The timing of the PCF8576 organizes the internal data flow of the device. This includes the transfer of display data from the display RAM to the display segment outputs. In cascaded applications, the synchronization signal $\overline{SYNC}$ maintains the correct timing relationship between the PCF8576s in the system. The timing also generates the LCD frame frequency which it derives as an integer multiple of the clock frequency (Table 3). The frame frequency is set by the choice of value for R_{osc} when internal clock is used, or by the frequency applied to pin 4 when external clock is used.

Table 3 LCD frame frequencies

PCF8576 mode	recommended R_{osc} (kΩ)	f_{frame}	nominal f_{frame} (Hz)
normal mode	180	$f_{CLK}/2880$	64
power-saving mode	1200	$f_{CLK}/480$	64

Timing (continued)

The ratio between the clock frequency and the LCD frame frequency depends on the mode in which the device is operating. In the normal mode, R_{OSC} = 180 kΩ will result in the nominal frame frequency. In the power-saving mode the reduction ratio is six times smaller; this allows the clock frequency to be reduced by a factor of six and for the same frame frequency R_{OSC} will be 1,2 MΩ. The reduced clock frequency and the increased value of R_{OSC} together contribute to a significant reduction in power dissipation. The lower clock frequency has the disadvantage of increasing the response time when large amounts of display data are transmitted on the I^2C bus. When a device is unable to 'digest' a display data byte before the next one arrives, it holds the SCL line LOW until the first display data byte is stored. This slows down the transmission rate of the I^2C bus but no data loss occurs.

Display latch

The display latch holds the display data while the corresponding multiplex signals are generated. There is a one-to-one relationship between the data in the display latch, the LCD segment outputs and one column of the display RAM.

Shift register

The shift register serves to transfer display information from the display RAM to the display latch while previous data are displayed.

Segment outputs

The LCD drive section includes 40 segment outputs S0 to S39 (pins 17 to 56) which should be connected directly to the LCD. The segment output signals are generated in accordance with the multiplexed backplane signals and with the data resident in the display latch. When less than 40 segment outputs are required the unused segment outputs should be left open.

Backplane outputs

The LCD drive section includes four backplane outputs BP0 to BP3 which should be connected directly to the LCD. The backplane output signals are generated in accordance with the selected LCD drive mode. If less than four backplane outputs are required the unused outputs can be left open. In the 1 : 3 multiplex drive mode BP3 carries the same signal as BP1, therefore these two adjacent outputs can be tied together to give enhanced drive capabilities. In the 1 : 2 multiplex drive mode BP0 and BP2, BP1 and BP3 respectively carry the same signals and may also be paired to increase the drive capabilities. In the static drive mode the same signal is carried by all four backplane outputs and they can be connected in parallel for very high drive requirements.

Display RAM

The display RAM is a static 40 x 4-bit RAM which stores LCD data. A logic 1 in the RAM bit-map indicates the 'on' state of the corresponding LCD segment; similarly, a logic 0 indicates the 'off' state. There is a one-to-one correspondence between the RAM addresses and the segment outputs, and between the individual bits of a RAM word and the backplane outputs. The first RAM column corresponds to the 40 segments operated with respect to backplane BP0 (Fig. 10). In multiplexed LCD applications the segment data of the second, third and fourth column of the display RAM are time-multiplexed with BP1, BP2 and BP3 respectively.

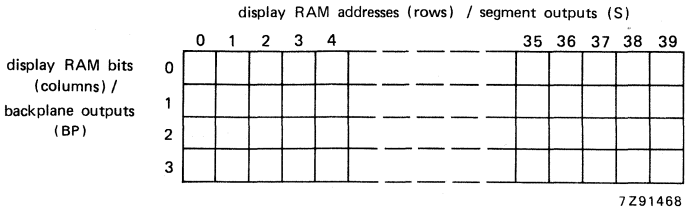


Fig. 10 Display RAM bit-map showing direct relationship between display RAM addresses and segment outputs, and between bits in a RAM word and backplane outputs.

When display data are transmitted to the PCF8576 the display bytes received are stored in the display RAM according to the selected LCD drive mode. To illustrate the filling order, an example of a 7-segment numeric display showing all drive modes is given in Fig. 11; the RAM filling organization depicted applies equally to other LCD types.

With reference to Fig. 11, in the static drive mode the eight transmitted data bits are placed in bit 0 of eight successive display RAM addresses. In the 1 : 2 multiplex drive mode the eight transmitted data bits are placed in bits 0 and 1 of four successive display RAM addresses. In the 1 : 3 multiplex drive mode these bits are placed in bits 0, 1 and 2 of three successive addresses, with bit 2 of the third address left unchanged. This last bit may, if necessary, be controlled by an additional transfer to this address but care should be taken to avoid overriding adjacent data because full bytes are always transmitted. In the 1 : 4 multiplex drive mode the eight transmitted data bits are placed in bits 0, 1, 2 and 3 of two successive display RAM addresses.

Data pointer

The addressing mechanism for the display RAM is realized using the data pointer. This allows the loading of an individual display data byte, or a series of display data bytes, into any location of the display RAM. The sequence commences with the initialization of the data pointer by the LOAD DATA POINTER command. Following this, an arriving data byte is stored starting at the display RAM address indicated by the data pointer thereby observing the filling order shown in Fig. 11. The data pointer is automatically incremented according to the LCD configuration chosen. That is, after each byte is stored, the contents of the data pointer are incremented by eight (static drive mode), by four (1 : 2 multiplex drive mode), by three (1 : 3 multiplex drive mode) or by two (1 : 4 multiplex drive mode).

Subaddress counter

The storage of display data is conditioned by the contents of the subaddress counter. Storage is allowed to take place only when the contents of the subaddress counter agree with the hardware subaddress applied to A0, A1 and A2 (pins 7, 8, and 9). The subaddress counter value is defined by the DEVICE SELECT command. If the contents of the subaddress counter and the hardware subaddress do not agree then data storage is inhibited but the data pointer is incremented as if data storage had taken place. The subaddress counter is also incremented when the data pointer overflows.

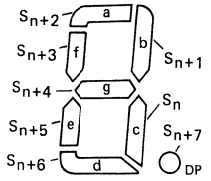
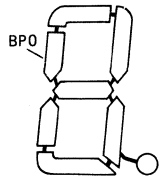
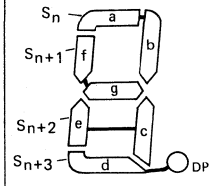
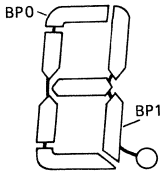
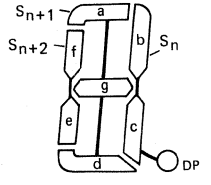
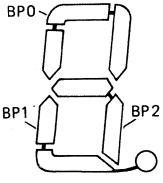
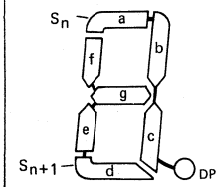
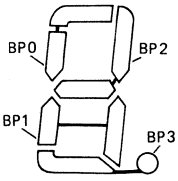
<table>
<tr><th>drive mode</th><th>LCD segments</th><th>LCD backplanes</th><th>display RAM filling order</th><th>transmitted display byte</th></tr>
<tr><td>static</td><td> 
</td><td> 
</td><td>
<table>
<tr><th></th><th></th><th>n</th><th>n + 1</th><th>n + 2</th><th>n + 3</th><th>n + 4</th><th>n + 5</th><th>n + 6</th><th>n + 7</th></tr>
<tr><td>bit/</td><td>0</td><td>c</td><td>b</td><td>a</td><td>f</td><td>g</td><td>e</td><td>d</td><td>DP</td></tr>
<tr><td>BP</td><td>1</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td></tr>
<tr><td></td><td>2</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td></tr>
<tr><td></td><td>3</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td></tr>
</table>
</td><td>msb lsb
c b a f g e d DP</td></tr>
<tr><td>1 : 2
multiplex</td><td> 
</td><td> 
</td><td>
<table>
<tr><th></th><th></th><th>n</th><th>n + 1</th><th>n + 2</th><th>n + 3</th></tr>
<tr><td>bit/</td><td>0</td><td>a</td><td>f</td><td>e</td><td>d</td></tr>
<tr><td>BP</td><td>1</td><td>b</td><td>g</td><td>c</td><td>DP</td></tr>
<tr><td></td><td>2</td><td>x</td><td>x</td><td>x</td><td>x</td></tr>
<tr><td></td><td>3</td><td>x</td><td>x</td><td>x</td><td>x</td></tr>
</table>
</td><td>msb lsb
a b f g e c d DP</td></tr>
<tr><td>1 : 3
multiplex</td><td> 
</td><td> 
</td><td>
<table>
<tr><th></th><th></th><th>n</th><th>n + 1</th><th>n + 2</th></tr>
<tr><td>bit/</td><td>0</td><td>b</td><td>a</td><td>f</td></tr>
<tr><td>BP</td><td>1</td><td>DP</td><td>d</td><td>e</td></tr>
<tr><td></td><td>2</td><td>c</td><td>g</td><td>x</td></tr>
<tr><td></td><td>3</td><td>x</td><td>x</td><td>x</td></tr>
</table>
</td><td>msb lsb
b DP c a d g f e</td></tr>
<tr><td>1 : 4
multiplex</td><td> 
</td><td> 
</td><td>
<table>
<tr><th></th><th></th><th>n</th><th>n + 1</th></tr>
<tr><td>bit/</td><td>0</td><td>a</td><td>f</td></tr>
<tr><td>BP</td><td>1</td><td>c</td><td>e</td></tr>
<tr><td></td><td>2</td><td>b</td><td>g</td></tr>
<tr><td></td><td>3</td><td>DP</td><td>d</td></tr>
</table>
</td><td>msb lsb
a c b DP f e g d</td></tr>
</table>

Fig. 11 Relationships between LCD layout, drive mode, display RAM filling order and display data transmitted over the I²C bus (x = data bit unchanged).

7Z91469

Subaddress counter (continued)

The storage arrangements described lead to extremely efficient data loading in cascaded applications. When a series of display bytes are being sent to the display RAM, automatic wrap-over to the next PCF8576 occurs when the last RAM address is exceeded. Subaddressing across device boundaries is successful even if the change to the next device in the cascade occurs within a transmitted character (such as during the 14th display data byte transmitted in 1 : 3 multiplex mode).

Output bank selector

This selects one of the four bits per display RAM address for transfer to the display latch. The actual bit chosen depends on the particular LCD drive mode in operation and on the instant in the multiplex sequence. In 1 : 4 multiplex, all RAM addresses of bit 0 are the first to be selected, these are followed by the contents of bit 1, bit 2 and then bit 3. Similarly in 1 : 3 multiplex, bits 0, 1 and 2 are selected sequentially. In 1 : 2 multiplex, bits 0 then 1 are selected and, in the static mode, bit 0 is selected.

The PCF8576 includes a RAM bank switching feature in the static and 1 : 2 multiplex drive modes. In the static drive mode, the BANK SELECT command may request the contents of bit 2 to be selected for display instead of bit 0 contents. In the 1 : 2 drive mode, the contents of bits 2 and 3 may be selected instead of bits 0 and 1. This gives the provision for preparing display information in an alternative bank and to be able to switch to it once it is assembled.

Input bank selector

The input bank selector loads display data into the display RAM according to the selected LCD drive configuration. Display data can be loaded in bit 2 in static drive mode or in bits 2 and 3 in 1 : 2 drive mode by using the BANK SELECT command. The input bank selector functions independently of the output bank selector.

Blinker

The display blinking capabilities of the PCF8576 are very versatile. The whole display can be blinked at frequencies selected by the BLINK command. The blinking frequencies are integer multiples of the clock frequency; the ratios between the clock and blinking frequencies depend on the mode in which the device is operating, as shown in Table 4.

An additional feature is for an arbitrary selection of LCD segments to be blinked. This applies to the static and 1 : 2 LCD drive modes and can be implemented without any communication overheads. By means of the output bank selector, the displayed RAM banks are exchanged with alternate RAM banks at the blinking frequency. This mode can also be specified by the BLINK command.

In the 1 : 3 and 1 : 4 multiplex modes, where no alternate RAM bank is available, groups of LCD segments can be blinked by selectively changing the display RAM data at fixed time intervals.

If the entire display is to be blinked at a frequency other than the nominal blinking frequency, this can be effectively performed by resetting and setting the display enable bit E at the required rate using the MODE SET command.

Blinker (continued)

Table 4 Blinking frequencies

blinking mode	normal operating mode ratio	power-saving mode ratio	nominal blinking frequency f_{blink} (Hz)
off	—	—	blinking off
2 Hz	$f_{CLK}/92160$	$f_{CLK}/15360$	2
1 Hz	$f_{CLK}/184320$	$f_{CLK}/30720$	1
0,5 Hz	$f_{CLK}/368640$	$f_{CLK}/61440$	0,5

CHARACTERISTICS OF THE I²C BUS

The I²C bus is for 2-way, 2-line communication between different ICs or modules. The two lines are a serial data line (SDA) and a serial clock line (SCL). Both lines must be connected to a positive supply via a pull-up resistor when connected to the output stages of a device. Data transfer may be initiated only when the bus is not busy.

Bit transfer

One data bit is transferred during each clock pulse. The data on the SDA line must remain stable during the HIGH period of the clock pulse as changes in the data line at this time will be interpreted as control signals.

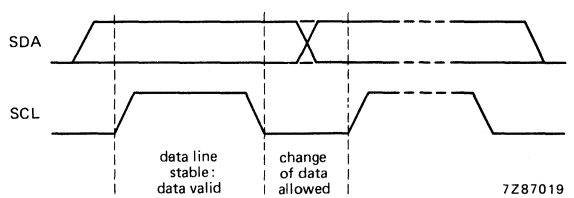


Fig. 12 Bit transfer.

Start and stop conditions

Both data and clock lines remain HIGH when the bus is not busy. A HIGH-to-LOW transition of the data line while the clock is HIGH is defined as the start condition (S). A LOW-to-HIGH transition of the data line while the clock is HIGH is defined as the stop condition (P).

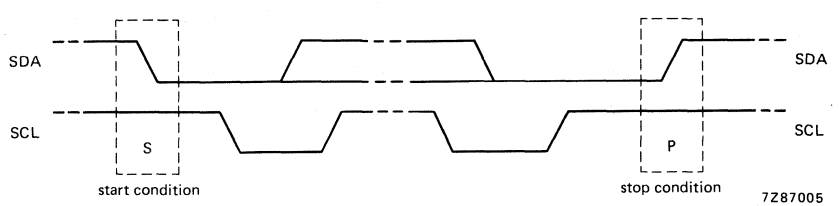


Fig. 13 Definition of start and stop conditions.

System configuration

A device generating a message is a "transmitter", a device receiving a message is a "receiver". The device that controls the message is the "master" and the devices which are controlled by the master are the "slaves".

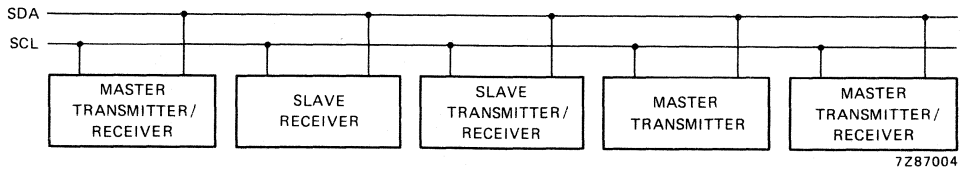


Fig. 14 System configuration.

Acknowledge

The number of data bytes transferred between the start and stop conditions from transmitter to receiver is not limited. Each byte is followed by one acknowledge bit. The acknowledge bit is a HIGH level put on the bus by the transmitter whereas the master generates an extra acknowledge related clock pulse. A slave receiver which is addressed must generate an acknowledge after the reception of each byte. Also a master must generate an acknowledge after the reception of each byte that has been clocked out of the slave transmitter. The device that acknowledges has to pull down the SDA line during the acknowledge clock pulse, so that the SDA line is stable LOW during the HIGH period of the acknowledge related clock pulse, set up and hold times must be taken into account. A master receiver must signal an end of data to the transmitter by *not* generating an acknowledge on the last byte that has been clocked out of the slave. In this event the transmitter must leave the data line HIGH to enable the master to generate a stop condition.

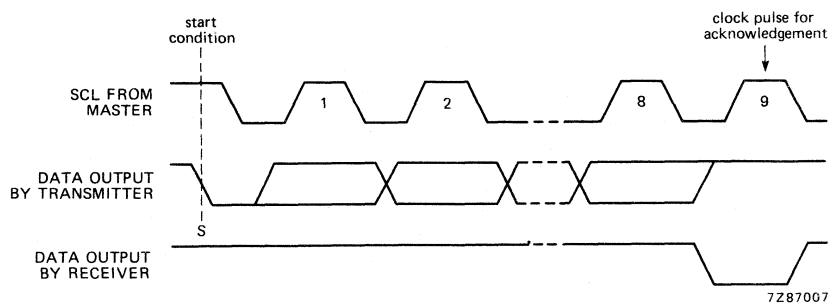


Fig. 15 Acknowledgement on the I²C bus.

Note

The general characteristics and detailed specification of the I²C bus are described in a separate data sheet (serial data buses) in handbook: ICs for digital systems in radio, audio and video equipment.

PCF8576 I²C bus controller

The PCF8576 acts as an I²C slave receiver. It does not initiate I²C bus transfers or transmit data to an I²C master receiver. The only data output from the PCF8576 are the acknowledge signals of the selected devices. Device selection depends on the I²C bus slave address, on the transferred command data and on the hardware subaddress.

In single device applications, the hardware subaddress inputs A0, A1 and A2 are normally tied to V_{SS} which defines the hardware subaddress 0. In multiple device applications A0, A1 and A2 are tied to V_{SS} or V_{DD} according to a binary coding scheme such that no two devices with a common I²C slave address have the same hardware subaddress.

In the power-saving mode it is possible that the PCF8576 is not able to keep up with the highest transmission rates when large amounts of display data are transmitted. If this situation occurs, the PCF8576 forces the SCL line LOW until its internal operations are completed. This is known as the 'clock synchronization feature' of the I²C bus and serves to slow down fast transmitters. Data loss does not occur.

Input filters

To enhance noise immunity in electrically adverse environments, RC low-pass filters are provided on the SDA and SCL lines.

I²C bus protocol

Two I²C bus slave addresses (0111000 and 0111001) are reserved for PCF8576. The least-significant bit of the slave address that a PCF8576 will respond to is defined by the level tied at its input SA0 (pin 10). Therefore, two types of PCF8576 can be distinguished on the same I²C bus which allows:

(a) up to 16 PCF8576s on the same I²C bus for very large LCD applications;

(b) the use of two types of LCD multiplex on the same I²C bus.

The I²C bus protocol is shown in Fig. 16. The sequence is initiated with a start condition (S) from the I²C bus master which is followed by one of the two PCF8576 slave addresses available. All PCF8576s with the corresponding SA0 level acknowledge in parallel the slave address but all PCF8576s with the alternative SA0 level ignore the whole I²C bus transfer. After acknowledgement, one or more command bytes (m) follow which define the status of the addressed PCF8576s. The last command byte is tagged with a cleared most-significant bit, the continuation bit C. The command bytes are also acknowledged by all addressed PCF8576s on the bus.

After the last command byte, a series of display data bytes (n) may follow. These display data bytes are stored in the display RAM at the address specified by the data pointer and the subaddress counter. Both data pointer and subaddress counter are automatically updated and the data are directed to the intended PCF8576 device. The acknowledgement after each byte is made only by the (A0, A1, A2) addressed PCF8576. After the last display byte, the I²C bus master issues a stop condition (P).

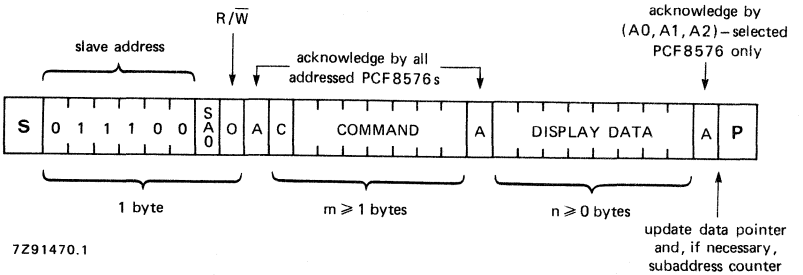


Fig. 16 I²C bus protocol.

Command decoder

The command decoder identifies command bytes that arrive on the I²C bus. All available commands carry a continuation bit C in their most-significant bit position (Fig. 17). When this bit is set, it indicates that the next byte of the transfer to arrive will also represent a command. If the bit is reset, it indicates the last command byte of the transfer. Further bytes will be regarded as display data.

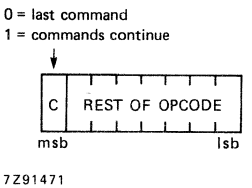


Fig. 17 General format of command byte.

The five commands available to the PCF8576 are defined in Table 5.

Command decoder (continued)

Table 5 Definition of PCF8576 commands

<table>
<tr><th>command/opcode</th><th>options</th><th>description</th></tr>
<tr><td>MODE SET

C | 1 | 0 | LP | E | B | M1 | M0</td><td>LCD drive mode | bits M1 M0
static (1 BP) | 0 1
1 : 2 MUX (2 BP) | 1 0
1 : 3 MUX (3 BP) | 1 1
1 : 4 MUX (4 BP) | 0 0

LCD bias | bit B
1/3 bias | 0
1/2 bias | 1

display status | bit E
disabled (blank) | 0
enabled | 1

mode | bit LP
normal mode | 0
power-saving mode | 1</td><td>Defines LCD drive mode

Defines LCD bias configuration

Defines display status
The possibility to disable the display allows implementation of blinking under external control

Defines power dissipation mode</td></tr>
<tr><td>LOAD DATA POINTER

C | 0 | P5 P4 P3 P2 P1 P0</td><td>bits P5 P4 P3 P2 P1 P0
6-bit binary value of 0 to 39</td><td>Six bits of immediate data, bits P5 to P0, are transferred to the data pointer to define one of forty display RAM addresses</td></tr>
<tr><td>DEVICE SELECT

C | 1 1 0 0 | A2 A1 A0</td><td>bits A0 A1 A2
3-bit binary value of 0 to 7</td><td>Three bits of immediate data, bits A0 to A2, are transferred to the subaddress counter to define one of eight hardware subaddresses</td></tr>
</table>

<table>
<tr><th>command/opcode</th><th>options</th><th>description</th></tr>
<tr><td>BANK SELECT

C | 1 1 1 1 1 0 | I | O</td><td>
<table>
<tr><td>static</td><td>1 : 2 MUX</td><td>bit I</td></tr>
<tr><td>RAM bit 0
RAM bit 2</td><td>RAM bits 0, 1
RAM bits 2, 3</td><td>0
1</td></tr>
</table>
<table>
<tr><td>static</td><td>1 : 2 MUX</td><td>bit O</td></tr>
<tr><td>RAM bit 0
RAM bit 2</td><td>RAM bits 0, 1
RAM bits 2, 3</td><td>0
1</td></tr>
</table>
</td><td>Defines input bank selection (storage of arriving display data)

Defines output bank selection (retrieval of LCD display data)

The BANK SELECT command has no effect in 1 : 3 and 1 : 4 multiplex drive modes</td></tr>
<tr><td>BLINK

C | 1 1 1 0 | A | BF1 BF0</td><td>
<table>
<tr><td>blink frequency</td><td>bits BF1</td><td>BF0</td></tr>
<tr><td>off</td><td>0</td><td>0</td></tr>
<tr><td>2 Hz</td><td>0</td><td>1</td></tr>
<tr><td>1 Hz</td><td>1</td><td>0</td></tr>
<tr><td>0,5 Hz</td><td>1</td><td>1</td></tr>
</table>
<table>
<tr><td>blink mode</td><td>bit A</td></tr>
<tr><td>normal blinking
alternation blinking</td><td>0
1</td></tr>
</table>
</td><td>Defines the blinking frequency

Selects the blinking mode; normal operation with frequency set by bits BF1, BF0, or blinking by alternation of display RAM banks. Alternation blinking does not apply in 1 : 3 and 1 : 4 multiplex drive modes</td></tr>
</table>

Display controller

The display controller executes the commands identified by the command decoder. It contains the status registers of the PCF8576 and coordinates their effects. The controller is also responsible for loading display data into the display RAM as required by the filling order.

Cascaded operation

In large display configurations, up to 16 PCF8576s can be distinguished on the same I²C bus by using the 3-bit hardware subaddress (A0, A1, A2) and the programmable I²C slave address (SA0). It is also possible to cascade up to 16 PCF8576s. When cascaded, several PCF8576s are synchronized so that they can share the backplane signals from one of the devices in the cascade. Such an arrangement is cost-effective in large LCD applications since the backplane outputs of only one device need to be through-plated to the backplane electrodes of the display. The other PCF8576s of the cascade contribute additional segment outputs but their backplane outputs are left open (Fig. 18).

The $\overline{SYNC}$ line is provided to maintain the correct synchronization between all cascaded PCF8576s. This synchronization is guaranteed after the power-on reset. The only time that $\overline{SYNC}$ is likely to be needed is if synchronization is accidently lost (e.g. by noise in adverse electrical environments; or by the definition of a multiplex mode when PCF8576s with differing SA0 levels are cascaded). $\overline{SYNC}$ is organized as an input/output pin; the output section being realized as an open-drain driver with an internal pull-up resistor. A PCF8576 asserts the $\overline{SYNC}$ line at the onset of its last active backplane signal and monitors the $\overline{SYNC}$ line at all other times. Should synchronization in the cascade be lost, it will be restored by the first PCF8576 to assert $\overline{SYNC}$. The timing relationships between the backplane waveforms and the $\overline{SYNC}$ signal for the various drive modes of the PCF8576 are shown in Fig. 19.

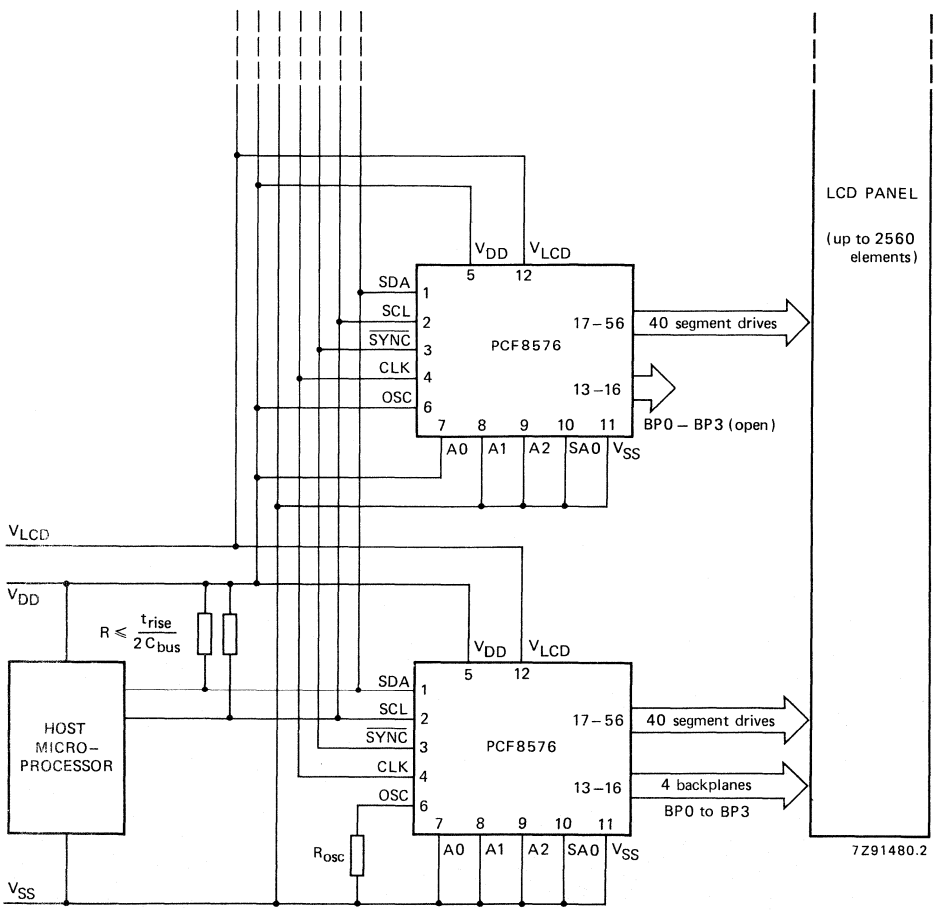


Fig. 18 Cascaded PCF8576 configuration.

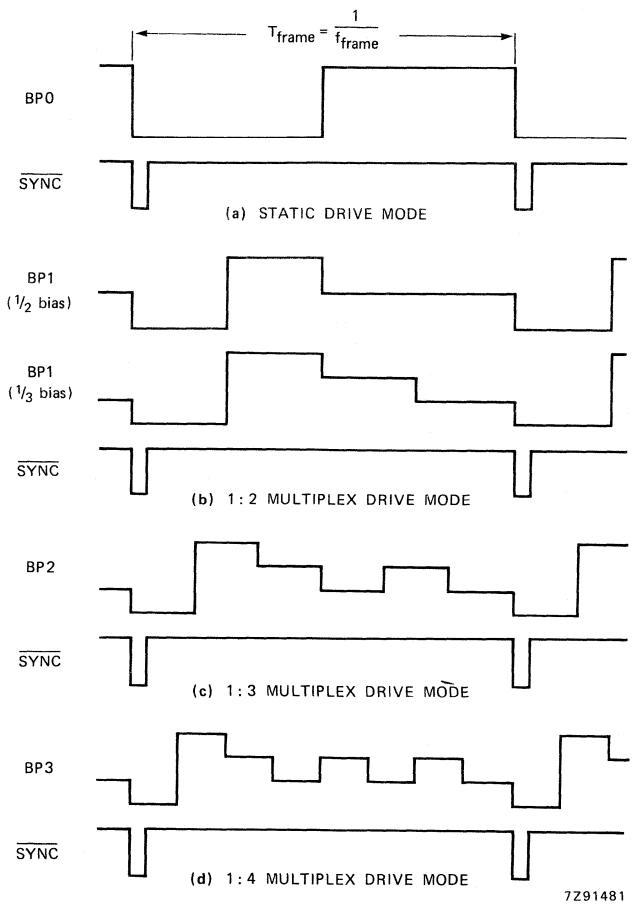


Fig. 19 Synchronization of the cascade for the various PCF8576 drive modes.

For single plane wiring of packaged PCF8576s and chip-on-glass cascading, see application information.

RATINGS

Limiting values in accordance with the Absolute Maximum System (IEC 134)

Supply voltage range	V_{DD}		−0,5 to + 11 V
LCD supply voltage range	V_{LCD}		V_{DD}−11 to V_{DD} V
Input voltage range (SCL; SDA; A0 to A2; OSC; CLK; $\overline{SYNC}$; SA0)	V_I		V_{SS} −0,5 to V_{DD} + 0,5 V
Output voltage range (S0 to S39; BP0 to BP3)	V_O		V_{LCD}−0,5 to V_{DD} + 0,5 V
D.C. input current	$\pm I_I$	max.	20 mA
D.C. output current	$\pm I_O$	max.	25 mA
V_{DD}, V_{SS} or V_{LCD} current	$\pm I_{DD}$, $\pm I_{SS}$, $\pm I_{LCD}$	max.	50 mA
Power dissipation per package	P_{tot}	max.	400 mW
Power dissipation per output	P_O	max.	100 mW
Storage temperature range	T_{stg}		−65 to + 150 °C

D.C. CHARACTERISTICS

V_{SS} = 0 V; V_{DD} = 2 to 9 V; V_{LCD} = V_{DD}−2 to V_{DD}−9 V;
T_{amb} = −40 to + 85 °C; unless otherwise specified

parameter	symbol	min.	typ.	max.	unit
Operating supply voltage	V_{DD}	2	–	9	V
LCD supply voltage (note 1)	V_{LCD}	V_{DD}−9	–	V_{DD}−2	V
Operating supply current (normal mode) at f_{CLK} = 200 kHz (note 2)	I_{DD}	–	–	180	μA
Power-saving mode supply current at V_{DD} = 3,5 V; V_{LCD} = 0 V; f_{CLK} = 35 kHz (note 2)	I_{LP}	–	–	60	μA
LCD supply current (normal mode) at f_{CLK} = 200 kHz (note 2)	I_{LCD}	–	–	120	μA
Logic					
Input voltage LOW	V_{IL}	V_{SS}	–	0,3 V_{DD}	V
Input voltage HIGH	V_{IH}	0,7 V_{DD}	–	V_{DD}	V
Output voltage LOW at I_O = 0 mA	V_{OL}	–	–	0,05	V
Output voltage HIGH at I_O = 0 mA	V_{OH}	V_{DD}−0,05	–	–	V
Output current LOW (CLK, $\overline{SYNC}$) at V_{OL} = 1,0 V; V_{DD} = 5 V	I_{OL1}	1	–	–	mA
Output current HIGH (CLK) at V_{OH} = 4,0 V; V_{DD} = 5 V	I_{OH}	–	–	−1	mA
Output current LOW (SDA; SCL) at V_{OL} = 0,4 V; V_{DD} = 5 V	I_{OL2}	3	–	–	mA
Leakage current (SA0; A0 to A2; CLK; SCL; SDA) at V_I = V_{SS} or V_{DD}	$\pm I_{L1}$	–	–	1	μA

parameter	symbol	min.	typ.	max.	unit
Leakage current (OSC) at $V_I = V_{DD}$	$\pm I_{L2}$	–	–	1	μA
Pull-up resistor ($\overline{SYNC}$)	R_{SYNC}	20	50	150	kΩ
Power-on reset level (note 3)	V_{REF}	–	1,0	1,6	V
Tolerable spike width on bus	t_{sw}	–	–	100	ns
Input capacitance (note 4)	C_I	–	–	7	pF
LCD outputs					
D.C. voltage component (BP0 to BP3) at C_{BP} = 35 nF	$\pm V_{BP}$	–	20	–	mV
D.C. voltage component (S0 to S39) at C_S = 5 nF	$\pm V_S$	–	20	–	mV
Output impedance (BP0 to BP3) at $V_{LCD} = V_{DD}-5$ V (note 5)	R_{BP}	–	–	5	kΩ
Output impedance (S0 to S39) at $V_{LCD} = V_{DD}-5$ V (note 5)	R_S	–	–	7,0	kΩ

A.C. CHARACTERISTICS (note 6)

V_{SS} = 0 V; V_{DD} = 2 to 9 V; $V_{LCD} = V_{DD}-2$ to $V_{DD}-9$ V;

$T_{amb} = -40$ to + 85 °C; unless otherwise specified

parameter	symbol	min.	typ.	max.	unit
Oscillator frequency (normal mode) at V_{DD} = 5 V; R_{osc} = 180 kΩ (note 7)	f_{CLK}	125	185	288	kHz
Oscillator frequency (power-saving mode) at V_{DD} = 3,5 V; R_{osc} = 1,2 MΩ	f_{CLKLP}	21	31	48	kHz
CLK HIGH time	t_{CLKH}	1	–	–	μs
CLK LOW time	t_{CLKL}	1	–	–	μs
$\overline{SYNC}$ propagation delay	t_{PSYNC}	–	–	400	ns
$\overline{SYNC}$ LOW time	t_{SYNCL}	1	–	–	μs
Driver delays with test loads at $V_{LCD} = V_{DD}-5$ V	t_{PLCD}	–	–	30	μs

A.C. CHARACTERISTICS (continued)

parameter	symbol	min.	typ.	max.	unit
I²C bus high-speed mode					
Bus free time	t_{BUF}	4,7	–	–	μs
Start condition hold time	$t_{HD;STA}$	4	–	–	μs
SCL LOW time	t_{LOW}	4,7	–	–	μs
SCL HIGH time	t_{HIGH}	4	–	–	μs
Start condition set-up time (repeated start code only)	$t_{SU;STA}$	4,7	–	–	μs
Data hold time	$t_{HD;DAT}$	0	–	–	μs
Data set-up time	$t_{SU;DAT}$	250	–	–	ns
Rise time	t_R	–	–	1	μs
Fall time	t_F	–	–	300	ns
Stop condition set-up time	$t_{SU;STO}$	4,7	–	–	μs
I²C bus low-speed mode					
Bus free time	t_{BUF}	105	–	–	μs
Start condition hold time	$t_{HD;STA}$	365	–	–	μs
SCL LOW time	t_{LOW}	105	–	155	μs
SCL HIGH time	t_{HIGH}	365	–	415	μs
Start condition set-up time (repeated start code only)	$t_{SU;STA}$	105	–	155	μs
Data hold time	$t_{HD;DAT}$	0	–	–	μs
Data set-up time	$t_{SU;DAT}$	250	–	–	ns
Rise time	t_R	–	–	1	μs
Fall time	t_F	–	–	300	ns
Stop condition set-up time	$t_{SU;STO}$	105	–	155	μs

Notes to characteristics

1. $V_{LCD} < V_{DD}-3$ V for 1/3 bias.
2. Outputs open; inputs at V_{SS} or V_{DD}; external clock with 50% duty cycle; I²C bus inactive.
3. Resets all logic when $V_{DD} < V_{REF}$.
4. Periodically sampled, not 100% tested.
5. Outputs measured one at a time.
6. All timing values referred to V_{IH} and V_{IL} levels with an input voltage swing of V_{SS} to V_{DD}.
7. At $f_{CLK} < 125$ kHz, I²C bus maximum transmission speed is derated.

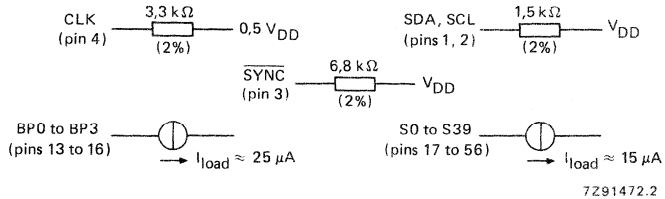


Fig. 20 Test loads.

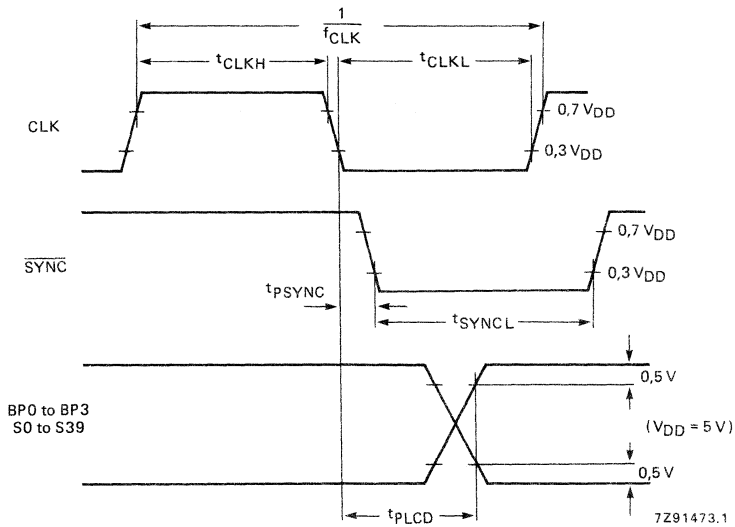


Fig. 21 Driver timing waveforms.

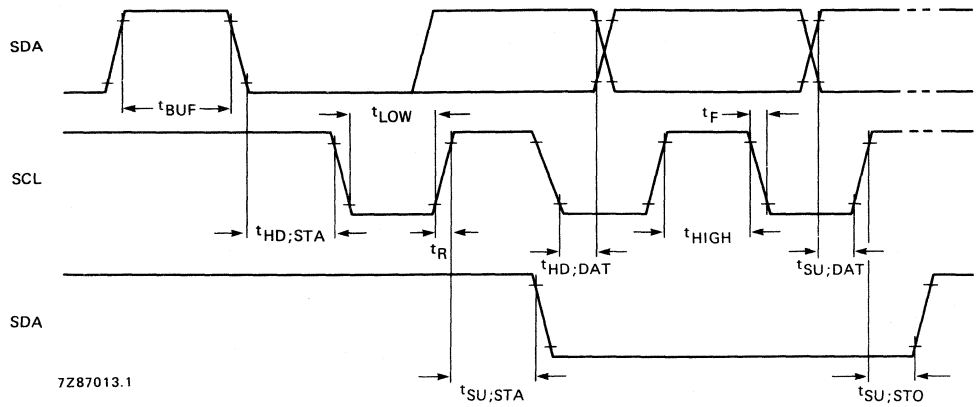


Fig. 22 I²C bus high-speed mode timing waveforms.

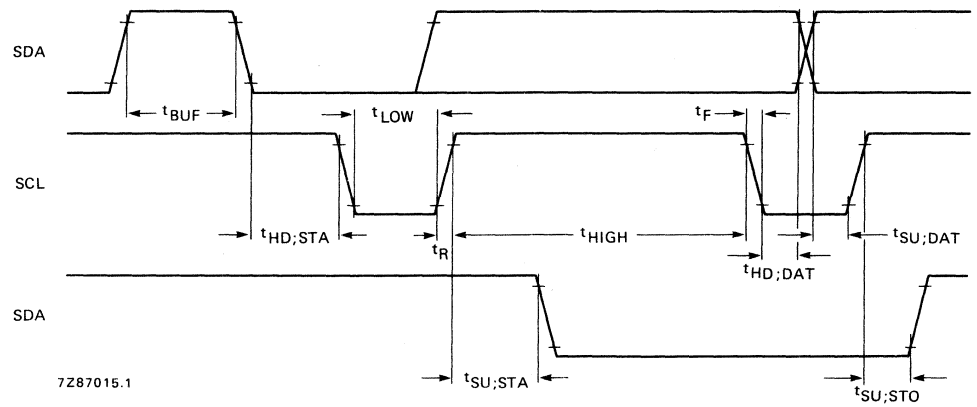
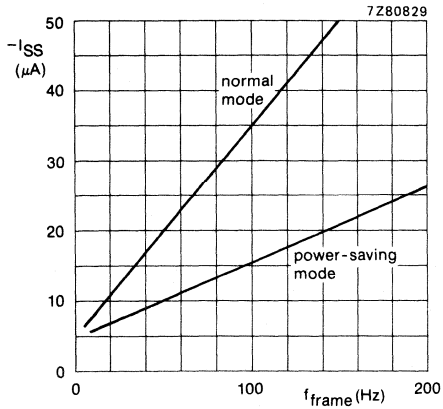
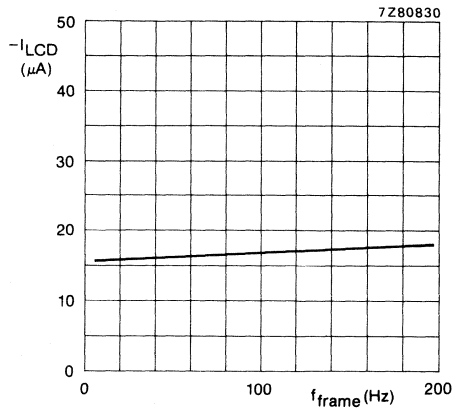


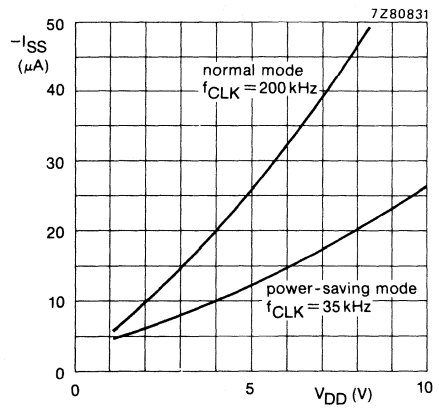
Fig. 23 I²C bus low-speed mode timing waveforms.



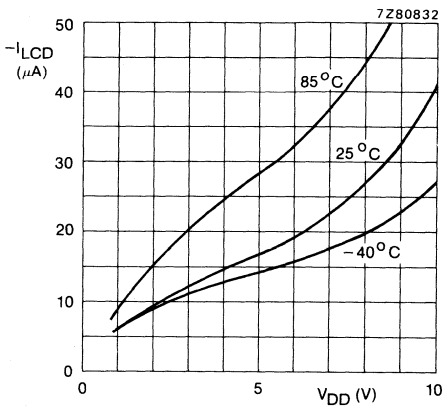
(a) V_{DD} = 5 V; V_{LCD} = 0 V; T_{amb} = 25 °C.



(b) V_{DD} = 5 V; V_{LCD} = 0 V; T_{amb} = 25 °C.

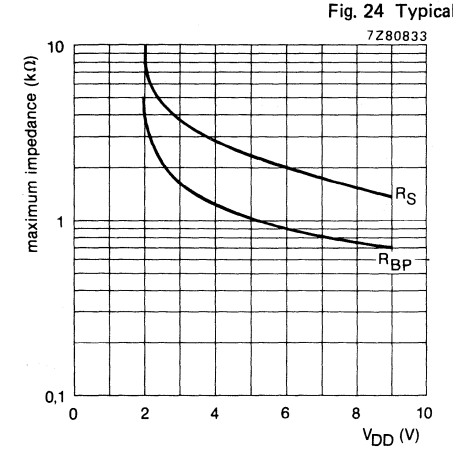


(c) V_{LCD} = 0 V; external clock; T_{amb} = −40 to + 85 °C.

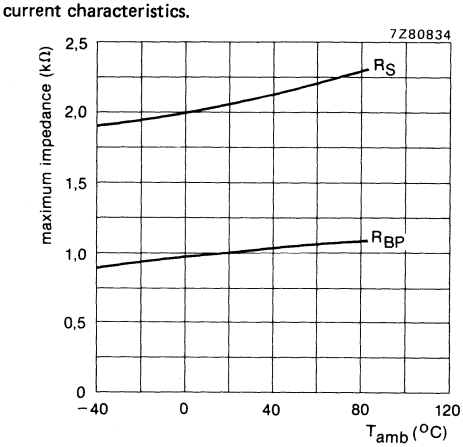


(d) V_{LCD} = 0 V; external clock; f_{CLK} = nominal frequency.

Fig. 24 Typical supply current characteristics.



(a) V_{LCD} = 0 V; T_{amb} = 25 °C.



(b) V_{DD} = 5 V; V_{LCD} = 0 V.

Fig. 25 Typical characteristics of LCD outputs.

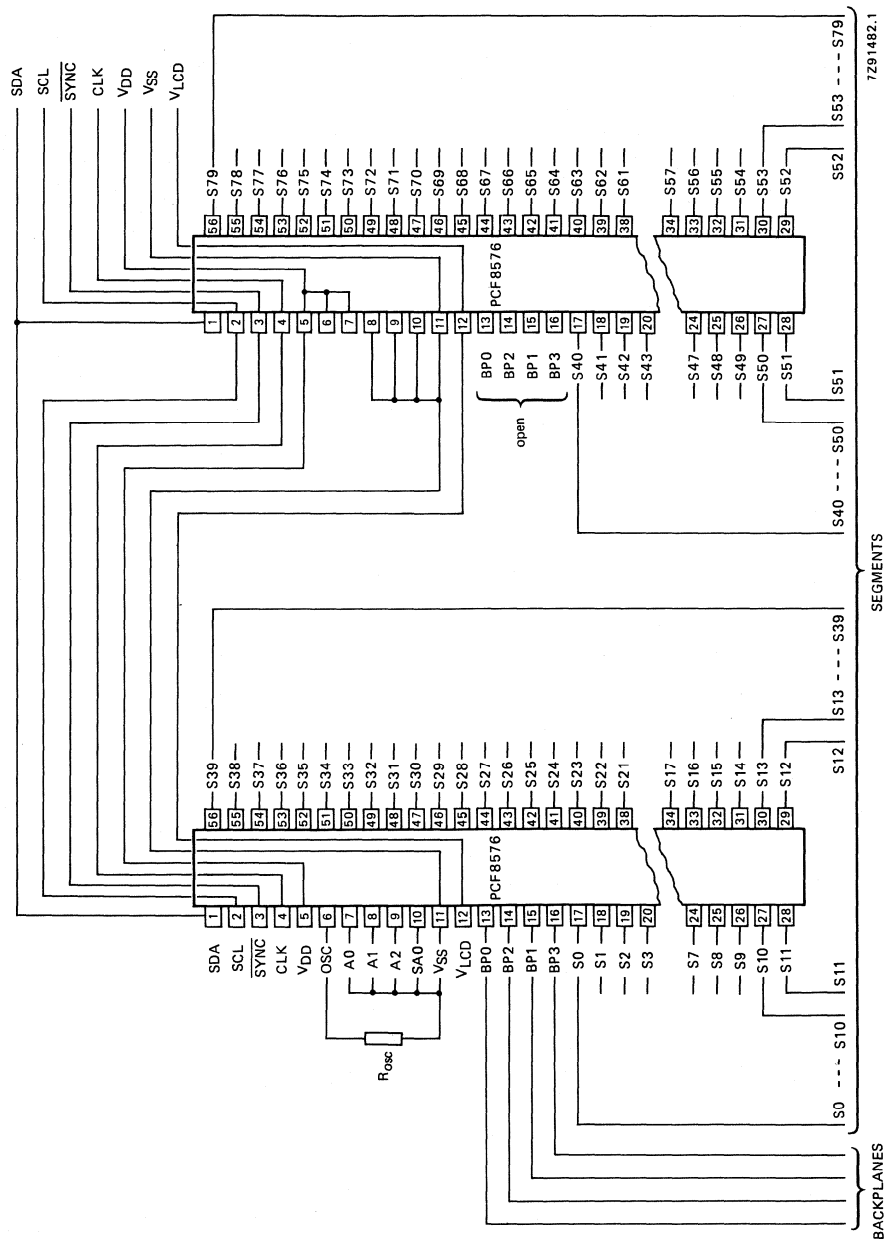


Fig. 26 Single plane wiring of packaged PCF8576s.

APPLICATION INFORMATION

Chip-on-glass cascadability in single plane

In chip-on-glass technology, where driver devices are bonded directly onto the glass of the LCD, it is important that the devices may be cascaded without the crossing of conductors, but the paths of conductors can be continued on the glass under the chip. All of this is facilitated by the PCF8576 bonding pad layout (Fig. 27). Pads needing bus interconnection between all PCF8576s of the cascade are V_{DD}, V_{SS}, CLK, SCL, SDA and $\overline{SYNC}$. These lines may be led to the corresponding pads of the next PCF8576 through the wide opening between the V_{LCD} pad and the backplane output pads. The only bussed line that does not require a second opening to lead through to the next PCF8576 is V_{LCD}, being the cascade centre. The placing of V_{LCD} adjacent to V_{SS} allows the two supplies to be tied together.

Fig. 28 shows the connection diagram for a cascaded PCF8576 application with single plane wiring. Note the use of the open space between the V_{LCD} pad and the backplane output pads to route V_{DD}, V_{SS}, CLK, SCL, SDA and $\overline{SYNC}$. The external connections may be made to either end of the cascade, wherever most convenient for the connector.

When an external clocking source is to be used, OSC of all devices should be tied to V_{DD}. The pads OSC, A0, A1, A2 and SA0 have been placed between V_{SS} and V_{DD} to facilitate wiring of oscillator, hardware subaddress and slave address.

APPLICATION INFORMATION (continued)

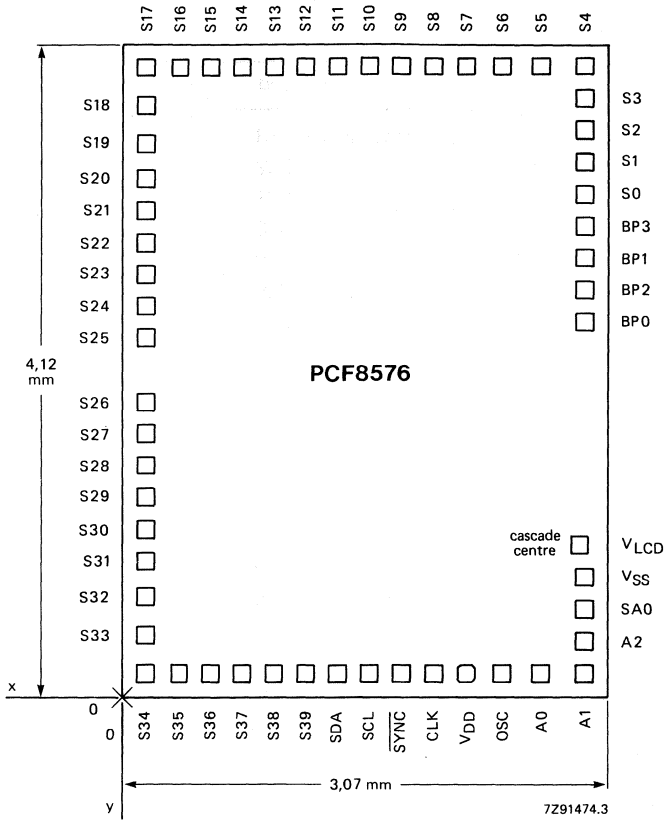


Fig. 27 PCF8576 bonding pad locations.

Bonding pad locations

All x/y coordinates are referenced to left-hand bottom corner (0/0, Fig. 27).

Dimensions in μm

pad	x	y	
S34	160	160	bottom
S35	380	↑	↑
S36	580	↑	↑
S37	780	↑	↑
S38	980	↑	↑
S39	1180	↑	↑
SDA	1380	↑	↑
SCL	1580	↑	↑
SYNC	1780	↑	↑
CLK	1980	↑	↑
V_{DD}	2180	↑	↑
OSC	2400	↑	↑
A0	2640	↑	↑
A1	2910	160	bottom
S17	160	3960	top
S16	380	↓	↓
S15	580	↓	↓
S14	780	↓	↓
S13	980	↓	↓
S12	1180	↓	↓
S11	1380	↓	↓
S10	1580	↓	↓
S9	1780	↓	↓
S8	1980	↓	↓
S7	2180	↓	↓
S6	2400	↓	↓
S5	2640	↓	↓
S4	2910	3960	top

pad	x	y	
S33	160	400	left
S32	↑	640	↑
S31	↑	860	↑
S30	↑	1060	↑
S29	↑	1260	↑
S28	↑	1460	↑
S27	↑	1660	↑
S26	↑	1860	↑
S25	↑	2260	↑
S24	↑	2460	↑
S23	↑	2660	↑
S22	↑	2860	↑
S21	↑	3060	↑
S20	↑	3260	↑
S19	↓	3480	↑
S18	160	3720	left
A2	2910	360	right
SA0	↓	560	↓
V_{SS}	↓	760	↓
V_{LCD}	↓	960	↓
BP0	↓	2360	↓
BP2	↓	2560	↓
BP1	↓	2760	↓
BP3	↓	2960	↓
S0	↓	3160	↓
S1	↓	3360	↓
S2	↓	3560	↓
S3	2910	3760	right

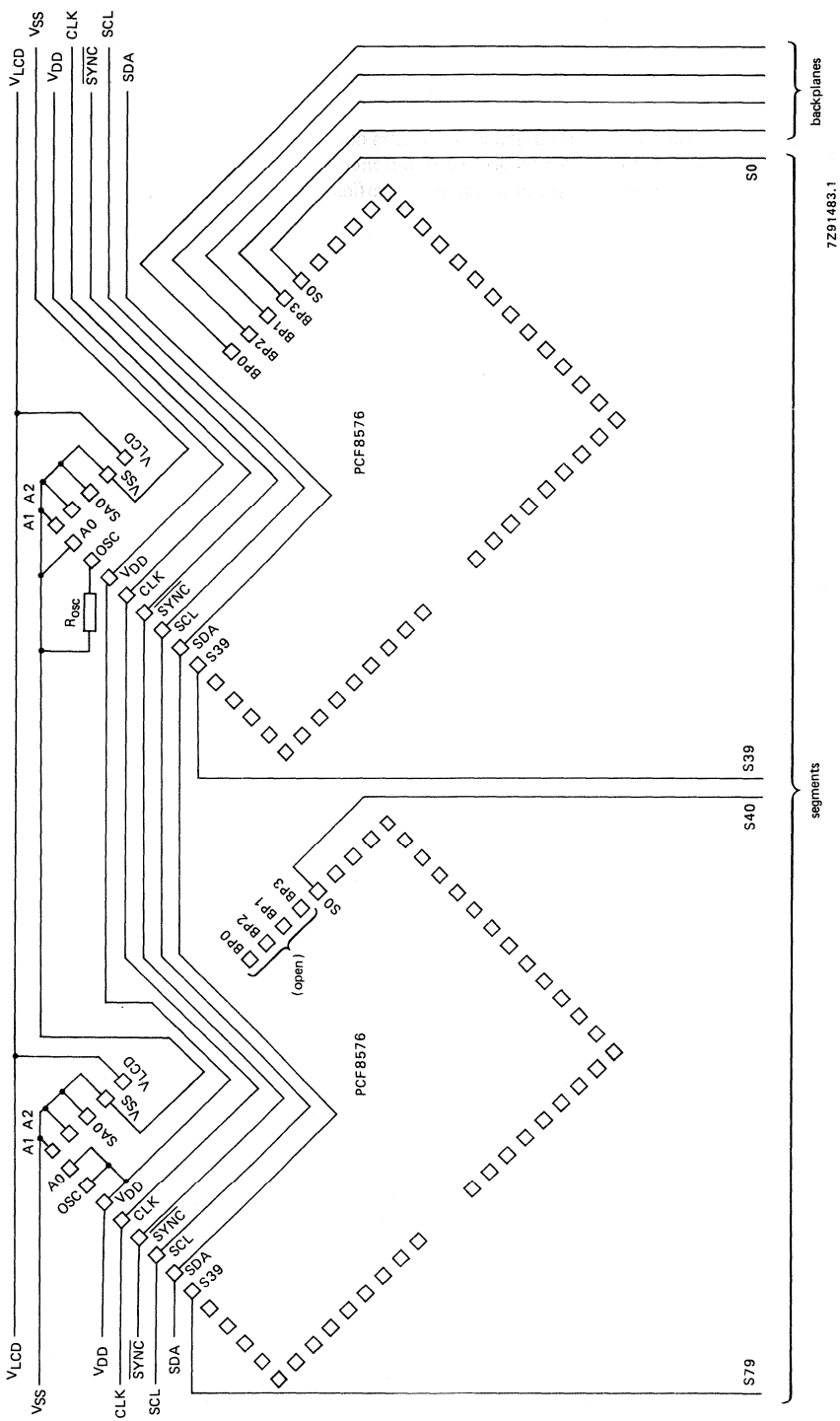
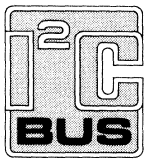


Fig. 28 Chip-on-glass application; cascaded PCF8576s with single-plane wiring (viewed from back of chip).



Purchase of Philips' I²C components conveys a license under the Philips' I²C patent to use the components in the I²C-system provided the system conforms to the I²C specifications defined by Philips.

LCD DIRECT/DUPLEX DRIVER WITH I²C BUS INTERFACE

GENERAL DESCRIPTION

The PCF8577 is a single chip, silicon gate CMOS circuit. It is designed to drive liquid crystal displays with up to 32 segments directly, or 64 segments in a duplex manner.

The two-line I²C bus interface substantially reduces wiring overheads in remote display applications. Bus traffic is minimized in multiple IC applications by automatic address incrementing, hardware sub-addressing and display memory switching (direct drive mode).

The PCF8577 and PCF8577A differ only in their slave address.

Features

- Direct/duplex drive modes with up to 32/64 LCD-segment drive capability per device
- Operating supply voltage: 2,5 to 9 V
- Low power consumption
- I²C bus interface
- Optimized pinning for single plane wiring
- Single-pin built-in oscillator
- Auto-incremented loading across device sub-address boundaries
- Display memory switching in direct drive mode
- May be used for I²C bus output expander
- System expansion up to 256 segments (512 segments with PCF8577A)
- Power-on-reset sets all segments off (to blank)

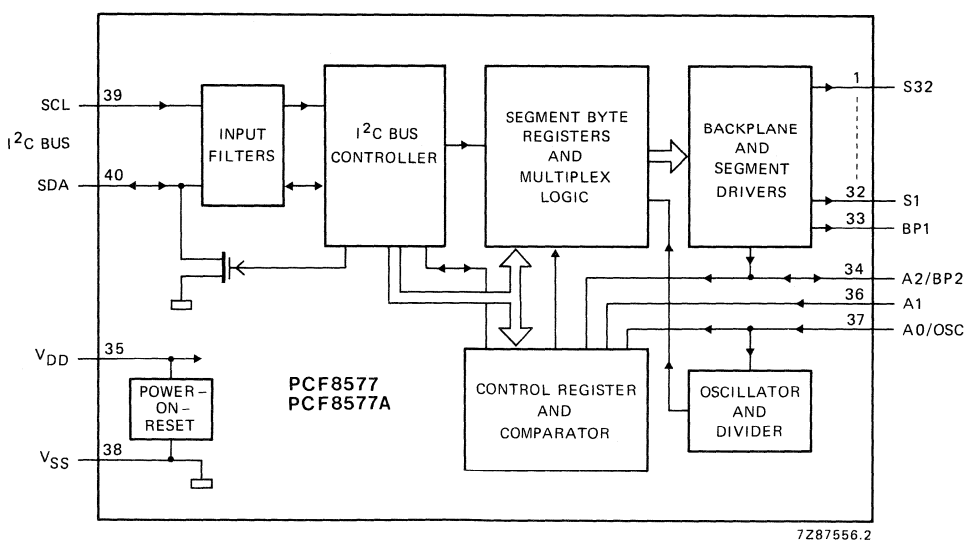


Fig. 1 Block diagram.

PACKAGE OUTLINES

PCF8577P, PCF8577AP: 40-lead DIL; plastic (SOT-129).
PCF8577T, PCF8577AT: 40-lead mini-pack; plastic (VSO-40; SOT-158A).

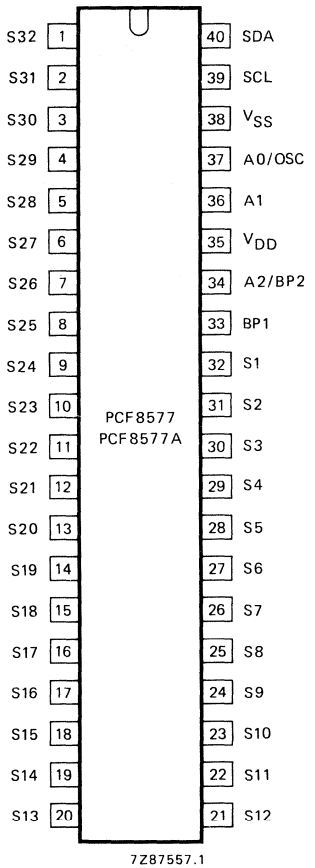


Fig. 2 Pinning diagram.

PINNING

Supply

35	V_{DD}	positive supply
38	V_{SS}	negative supply

I^2C bus

40	SDA	I^2C bus data line
39	SCL	I^2C bus clock line

Inputs

36	A1	hardware address line
37	A0/OSC	hardware address line/oscillator pin

Outputs

1 – 32	S1 – S32	segment outputs

Input – Output

34	A2/BP2	harware address line/cascade sync input/backplane output
33	BP1	cascade sync input/backplane output

FUNCTIONAL DESCRIPTION

Hardware sub-address A0, A1, A2

The hardware sub-address lines A0, A1, A2 are used to program the device sub-address for each PCF8577 on the bus. Lines A0 and A2 are shared with OSC and BP2 respectively to reduce pin-out requirements.

A0/OSC Line A0 is defined as LOW (logic 0) when this pin is used for the local oscillator or when connected to V_{SS}. Line A0 is defined as HIGH (logic 1) when connected to V_{DD}.

A1 Line A1 must be defined as LOW (logic 0) or as HIGH (logic 1) by connection to V_{SS} or V_{DD} respectively.

A2/BP2 In the direct drive mode the second backplane signal BP2 is not used and the A2/BP2 pin is exclusively the A2 input. Line A2 is defined as LOW (logic 0) when connected to V_{SS} or, if this is not possible, by leaving it unconnected (internal pull-down). Line A2 is defined as HIGH (logic 1) when connected to V_{DD}.

In the duplex drive mode the second backplane signal BP2 is required and the A2 signal is undefined. In this mode device selection is made exclusively from lines A0 and A1.

Oscillator A0/OSC

The PCF8577 has a single-pin built-in oscillator which provides the modulation for the LCD segment driver outputs. One external resistor and one external capacitor are connected to the A0/OSC pin to form the oscillator. In an expanded system containing more than one PCF8577 the backplane signals are usually common to all devices and only one oscillator is needed. The devices which are not used for the oscillator are put into the expansion mode by connecting the A0/OSC pin to either V_{DD} or V_{SS} depending on the required state for A0. In the expansion mode each PCF8577 is synchronized from the backplane signal(s).

User-accessible registers

There are nine user-accesible 1-byte registers. The first is a control register which is used to control the loading of data into the segment byte registers and to select display options. The other eight are segment byte registers, split into two banks of storage, which store the segment data. The set of even numbered segment byte registers is called BANK A. Odd numbered segment byte registers are called BANK B.

There are two slave addresses, one for PCF8577, and one for PCF8577A (see Fig. 14). All addressed devices load the second byte into the control register and each device maintains an identical copy of the control byte in the control register at all times (see I²C bus protocol Fig. 15).

The control register is shown in more detail in Fig. 3. The least-significant bits select which device and which segment byte register are loaded next. This part of the register is therefore called the Segment Byte Vector (SBV).

The upper three bits of the SBV (V5 to V3) are compared with the hardware sub-address input signals A2, A1 and A0. If they are the same then the device is enabled for loading, if not the device ignores incoming data but remains active.

The three least-significant bits of the SBV (V2 to V0) address one of the segment byte registers within the enabled chip for loading segment data.

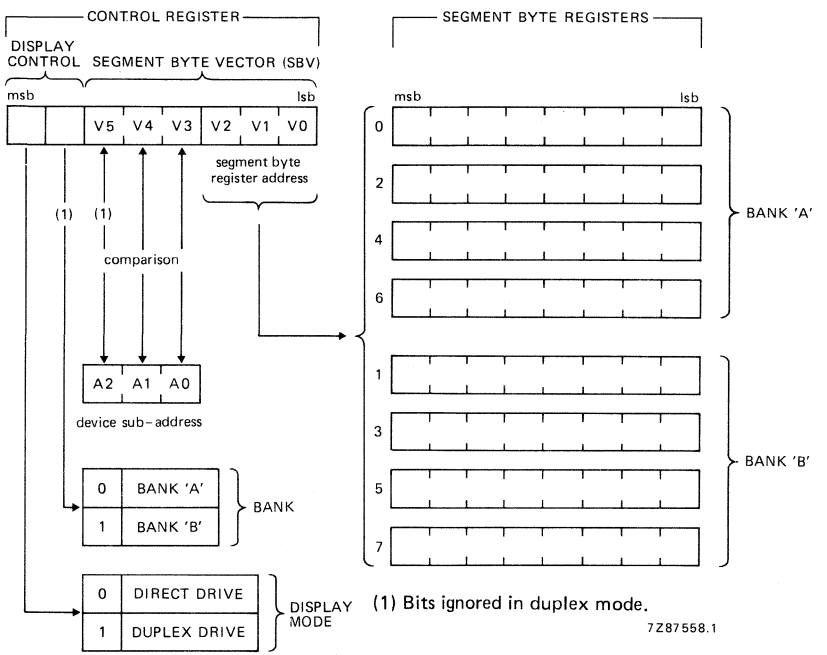


Fig. 3 PCF8577 register organization.

FUNCTIONAL DESCRIPTION (continued)

The control register also has two display control bits. These bits are named MODE and BANK. The MODE bit selects whether the display outputs are configured for direct or duplex drive displays. The BANK bit allows the user to display BANK A or BANK B.

Auto-incremented loading

After each segment byte is loaded the SBV is incremented automatically, thus auto-incremented loading occurs if more than one segment byte is received in a data transfer.

Since the SBV addresses both device and segment registers, auto-incremented loading may proceed across device boundaries provided that the hardware sub-addresses are arranged contiguously.

Direct drive mode

The PCF8577 is set to the direct drive mode by loading the MODE control bit with logic 0. In this mode only four bytes are needed to store the data for the 32 segment drivers. Setting the BANK bit to logic 0 selects even bytes (BANK A); setting the BANK bit to logic 1 selects odd bytes (BANK B).

In the direct drive mode the SBV is auto-incremented by two after the loading of each segment byte register. This means that auto-incremented loading of BANK A or BANK B is possible. Either bank may be completely or partially loaded irrespective of which bank is being displayed. Direct drive output waveforms are shown in Fig. 4.

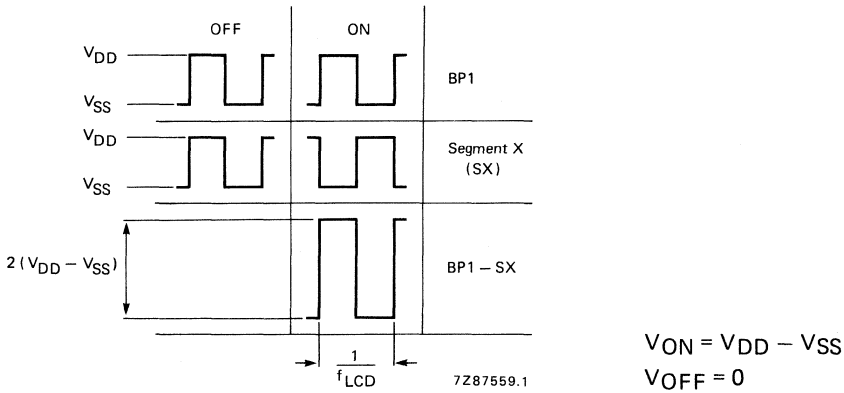


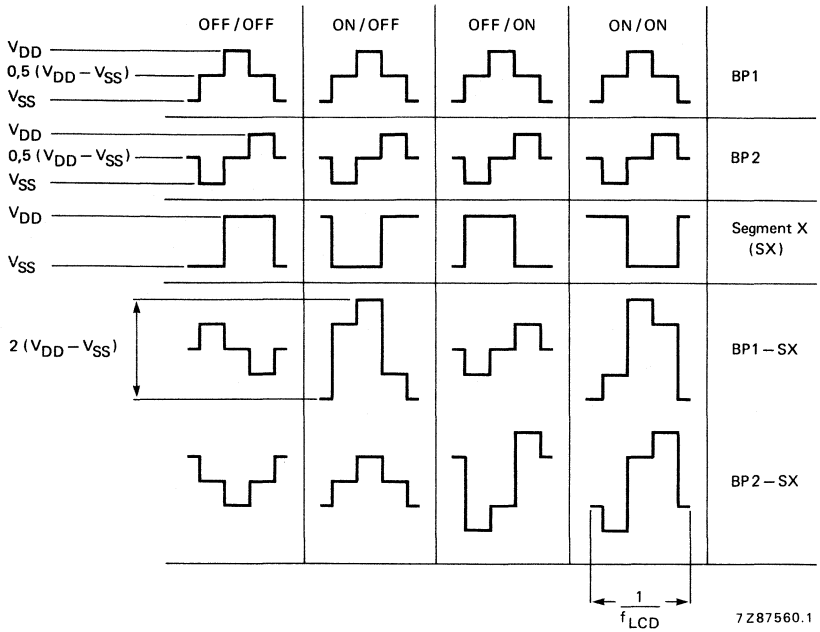
Fig. 4 Direct drive mode display output waveforms.

Duplex mode

The PCF8577 is set to the duplex mode by loading the MODE bit with logic 1. In this mode a second backplane signal (BP2) is needed and pin A2/BP2 is used for this; therefore A2 and its equivalent SBV bit V5 are undefined. The SBV auto-increments by one between loaded bytes.

All of the segment bytes are needed to store data for the 32 segment drivers and the BANK bit is ignored.

Duplex mode output waveforms are shown in Fig. 5.



$V_{ON} = 0{,}79\ (V_{DD} - V_{SS})$

$V_{OFF} = 0{,}35\ (V_{DD} - V_{SS})$

$\frac{V_{ON}}{V_{OFF}} = 2{,}26$

Fig. 5 Duplex mode display output waveforms.

DEVELOPMENT DATA

CHARACTERISTICS OF THE I²C BUS

The I²C bus is for 2-way, 2-line communication between different ICs or modules. The two lines are a serial data line (SDA) and a serial clock line (SCL). Both lines must be connected to a positive supply via a pull-up resistor when connected to the output stages of a device. Data transfer may be initiated only when the bus is not busy.

Bit transfer

One data bit is transferred during each clock pulse. The data on the SDA line must remain stable during the HIGH period of the clock pulse as changes in the data line at this time will be interpreted as control signals.

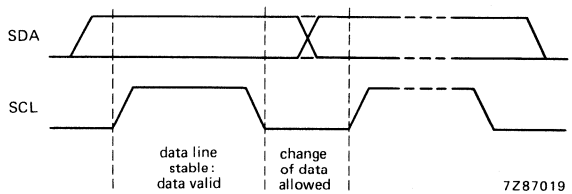


Fig. 6 Bit transfer.

Start and stop conditions

Both data and clock lines remain HIGH when the bus is not busy. A HIGH-to-LOW transition of the data line, while the clock is HIGH is defined as the start condition (S). A LOW-to-HIGH transition of the data line while the clock is HIGH is defined as the stop condition (P).

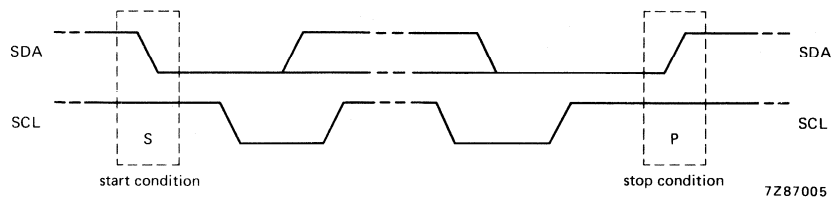


Fig. 7 Definition of start and stop conditions.

System configuration

A device generating a message is a "transmitter", a device receiving a message is the "receiver". The device that controls the message is the "master" and the devices which are controlled by the master are the "slaves".

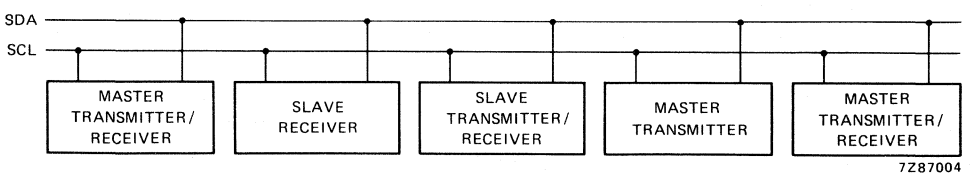


Fig. 8 System configuration.

Acknowledge

The number of data bytes transferred between the start and stop conditions from transmitter to receiver is not limited. Each byte is followed by one acknowledge bit. The acknowledge bit is a HIGH level put on the bus by the transmitter whereas the master generates an extra acknowledge related clock pulse. A slave receiver which is addressed must generate an acknowledge after the reception of each byte. Also a master must generate an acknowledge after the reception of each byte that has been clocked out of the slave transmitter. The device that acknowledges has to pull down the SDA line during the acknowledge clock pulse, so that the SDA line is stable LOW during the HIGH period of the acknowledge related clock pulse, set up and hold times must be taken into account. A master receiver must signal an end of data to the transmitter by *not* generating an acknowledge on the last byte that has been clocked out of the slave. In this event the transmitter must leave the data line HIGH to enable the master to generate a stop condition.

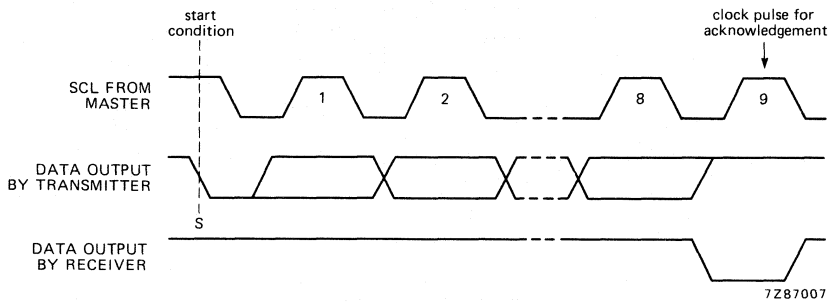


Fig. 9 Acknowledgement on the I²C bus.

Timing specifications

Within the I²C bus specifications a high-speed mode and a low-speed mode are defined. The PCF8577 operates in both modes and the timing requirements are as follows:

High-speed mode

Masters generate a bus clock with a maximum frequency of 100 kHz. Detailed timing is shown in Fig. 10.

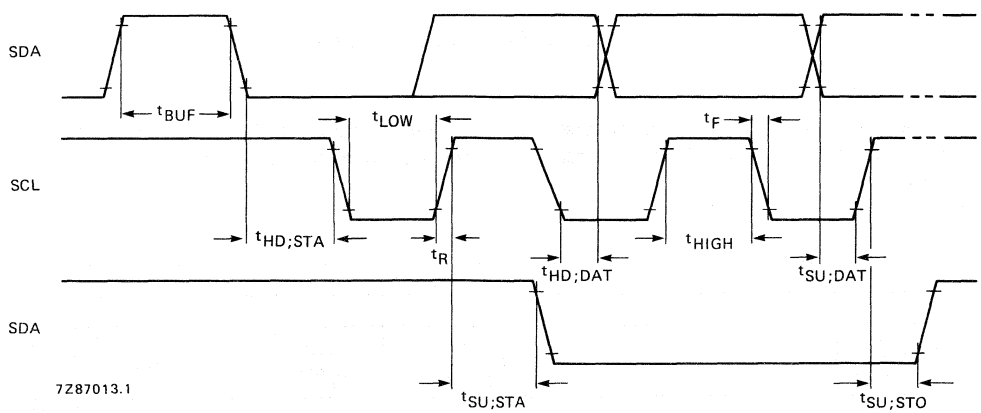


Fig. 10 Timing of the high-speed mode.

CHARACTERISTICS OF THE I^2C BUS (continued)

Where:

t_{BUF}	$t \geqslant t_{LOWmin}$	The minimum time the bus must be free before a new transmission can start
$t_{HD;STA}$	$t \geqslant t_{HIGHmin}$	Start condition hold time
t_{LOWmin}	4,7 μs	Clock LOW period
$t_{HIGHmin}$	4 μs	Clock HIGH period
$t_{SU;STA}$	$t \geqslant t_{LOWmin}$	Start condition set-up time, only valid for repeated start code
$t_{HD;DAT}$	$t \geqslant 0$ μs	Data hold time
$t_{SU;DAT}$	$t \geqslant 250$ ns	Data set-up time
t_R	$t \leqslant 1$ μs	Rise time of both the SDA and SCL line
t_F	$t \leqslant 300$ ns	Fall time of both the SDA and SCL line
$t_{SU;STO}$	$t \geqslant t_{LOWmin}$	Stop condition set-up time

Note

All the timing values referred to V_{IH} and V_{IL} levels with a voltage swing of V_{SS} to V_{DD}.

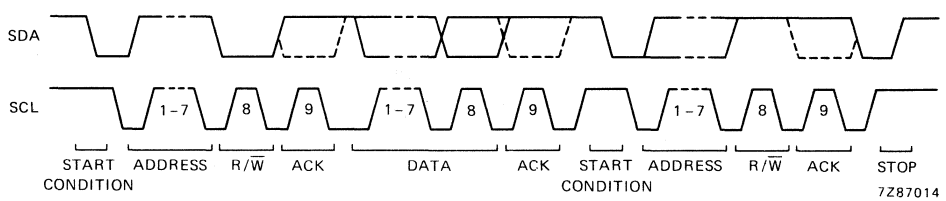


Fig. 11 Complete data transfer in the high-speed mode.

Where:

Clock t_{LOWmin}	4,7 μs
$t_{HIGHmin}$	4 μs
The dashed line is the acknowledgement of the receiver	
Mark-to-space ratio	1 : 1 (LOW-to-HIGH)
Max. number of bytes	unrestricted
Premature termination of transfer	allowed by generation of STOP condition
Acknowledge clock bit	must be provided by the master

Low-speed mode

Masters generate a bus clock with a maximum frequency of 2 kHz; a minimum LOW period of 105 μs and a minimum HIGH period of 365 μs. The mark-to-space ratio is 1 : 3 LOW-to-HIGH. Detailed timing is shown in Fig. 12.

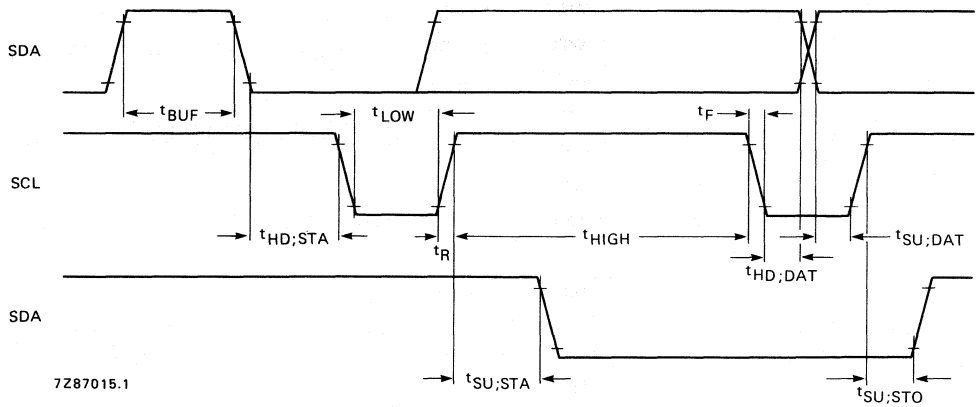


Fig. 12 Timing of the low-speed mode.

Where:

t_{BUF}	$t \geqslant 105\ \mu s$ (t_{LOWmin})
$t_{HD;STA}$	$t \geqslant 365\ \mu s$ ($t_{HIGHmin}$)
t_{LOW}	130 μs ± 25 μs
t_{HIGH}	390 μs ± 25 μs
$t_{SU;STA}$	130 μs ± 25 μs*
$t_{HD;DAT}$	$t \geqslant 0\ \mu s$
$t_{SU;DAT}$	$t \geqslant 250$ ns
t_R	$t \leqslant 1\ \mu s$
t_F	$t \leqslant 300$ ns
$t_{SU;STO}$	130 μs ± 25 μs

Note

All the timing values referred to V_{IH} and V_{IL} levels with a voltage swing of V_{SS} to V_{DD}, for definitions see high-speed mode.

* Only valid for repeated start code.

CHARACTERISTICS OF THE I²C BUS (continued)

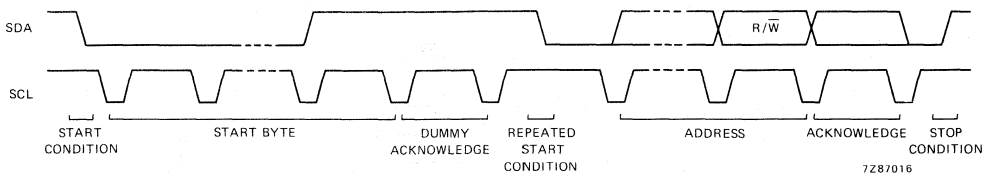


Fig. 13 Complete data transfer in the low-speed mode.

Where:

Clock t_{LOWmin}	130 μs ± 25 μs
$t_{HIGHmin}$	390 μs ± 25 μs
Mark-to-space ratio	1 : 3 (LOW-to-HIGH)
Start byte	0000 0001
Maximum number of bytes	6
Premature termination of transfer	not allowed
Acknowledge clock bit	must be provided by master

Note

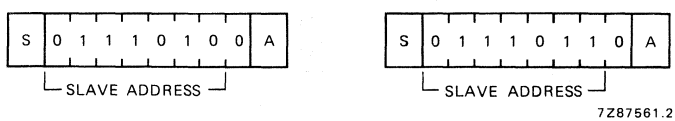
The general characteristics and detailed specification of the I²C bus are described in a separate data sheet (serial data buses) in handbook: ICs for digital systems in radio, audio and video equipment.

ADDRESSING

Before any data is transmitted on the I²C bus, the device which should respond is addressed first. The addressing is always done with the first byte transmitted after the start procedure.

Slave address

The slave address for PCF8577 and PCF8577A are shown in Fig. 14.



(a) PCF8577. (b) PCF8577A.

Fig. 14 PCF8577 and PCF8577A slave addresses.

I²C bus protocol

The PCF8577 I²C bus protocol is shown in Fig. 15.

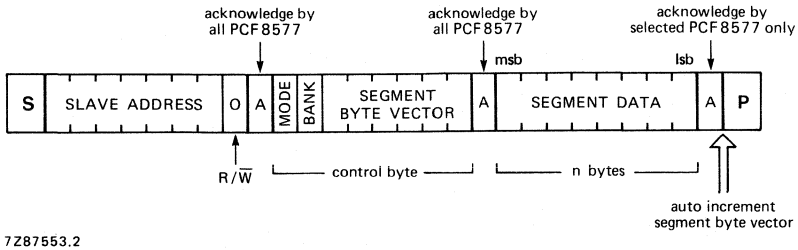


Fig. 15 I²C bus protocol.

The PCF8577 is a slave receiver and has a fixed slave address (Fig. 14). All PCF8577 on the same bus acknowledge the slave address in parallel. The second byte is always the control byte and is loaded into the control register of each PCF8577 on the bus. Subsequent data bytes are loaded into the segment registers of the selected device. Any number of data bytes may be loaded in one transfer and in an expanded system rollover of the SBV from 111 111 to 000 000 is allowed. If a stop (P) condition is given after the control byte acknowledge the segment data remains unchanged. This allows the BANK bit to be toggled without changing the segment register contents. During loading of segment data only the selected PCF8577 gives an acknowledge. Loading is terminated by generating a stop (P) condition.

DEVELOPMENT DATA

DISPLAY MEMORY MAPPING

The mapping between the eight segment registers and the segment outputs S1 to S32 is shown in Tables 1 and 2.

Since only one register bit per segment is needed in the direct drive mode, the BANK bit allows swapping of display information. If BANK is set to logic 0 even bytes (BANK A) are displayed; if BANK is set to logic 1 odd bytes (BANK B) are displayed. BP1 is always used for the backplane output in the direct drive mode.

Table 1 Segment byte — segment driver mapping in the direct drive mode.

MODE	BANK	V2	V1	V0	SEGMENT REGISTER / BIT	MSB 7	6	5	4	3	2	1	LSB 0	BACKPLANE
0	0	0	0	0	0	S8	S7	S6	S5	S4	S3	S2	S1	BP1
0	1	0	0	1	1	S8	S7	S6	S5	S4	S3	S2	S1	BP1
0	0	0	1	0	2	S16	S15	S14	S13	S12	S11	S10	S9	BP1
0	1	0	1	1	3	S16	S15	S14	S13	S12	S11	S10	S9	BP1
0	0	1	0	0	4	S24	S23	S22	S21	S20	S19	S18	S17	BP1
0	1	1	0	1	5	S24	S23	S22	S21	S20	S19	S18	S17	BP1
0	0	1	1	0	6	S32	S31	S30	S29	S28	S27	S26	S25	BP1
0	1	1	1	1	7	S32	S31	S30	S29	S28	S27	S26	S25	BP1

Mapping example: bit 0 of register 7 controls the LCD segment S25 if BANK bit is a logic 1.

DISPLAY MEMORY MAPPING (continued)

Even bytes (BANK A) correspond to backplane 1 (BP1) and odd bytes (BANK B) correspond to backplane 2 (BP2).

Table 2 Segment byte – segment driver mapping in the duplex mode.

MODE	BANK	V2	V1	V0	SEGMENT REGISTER / BIT	MSB 7	6	5	4	3	2	1	LSB 0	BACKPLANE
1	x	0	0	0	0	S8	S7	S6	S5	S4	S3	S2	S1	**BP1**
1	x	0	0	1	1	S8	S7	S6	S5	S4	S3	S2	S1	**BP2**
1	x	0	1	0	2	S16	S15	S14	S13	S12	S11	S10	S9	**BP1**
1	x	0	1	1	3	S16	S15	S14	S13	S12	S11	S10	S9	**BP2**
1	x	1	0	0	4	S24	S23	S22	S21	S20	S19	S18	S17	**BP1**
1	x	1	0	1	5	S24	S23	S22	S21	S20	S19	S18	S17	**BP2**
1	x	1	1	0	6	S32	S31	S30	S29	S28	S27	S26	S25	**BP1**
1	x	1	1	1	7	S32	S31	S30	S29	S28	S27	S26	S25	**BP2**

X = don't care.

Mapping example: bit 7 of register 5 controls the LCD segment S24/BP2.

RATINGS

Limiting values in accordance with the Absolute Maximum System (IEC 134)

Supply voltage range	V_{DD}	−0,5 to 11	V
Voltage on any pin	V_I	V_{SS} − 0,8 to V_{DD} + 0,8	V
D.C. input current	$\pm I_I$	max. 20	mA
D.C. output current	$\pm I_O$	max. 25	mA
V_{DD} or V_{SS} current	$\pm I_{DD}$, I_{SS}	max. 50	mA
Power dissipation per package	P_{tot}	max. 500*	mW
Power dissipation per output	P	max. 100	mW
Operating ambient temperature range	T_{amb}	−40 to +85	°C
Storage temperature range	T_{stg}	−65 to +150	°C

* Derate 7,7 mW/K when $T_{amb} > 60$ °C.

DEVELOPMENT DATA

CHARACTERISTICS

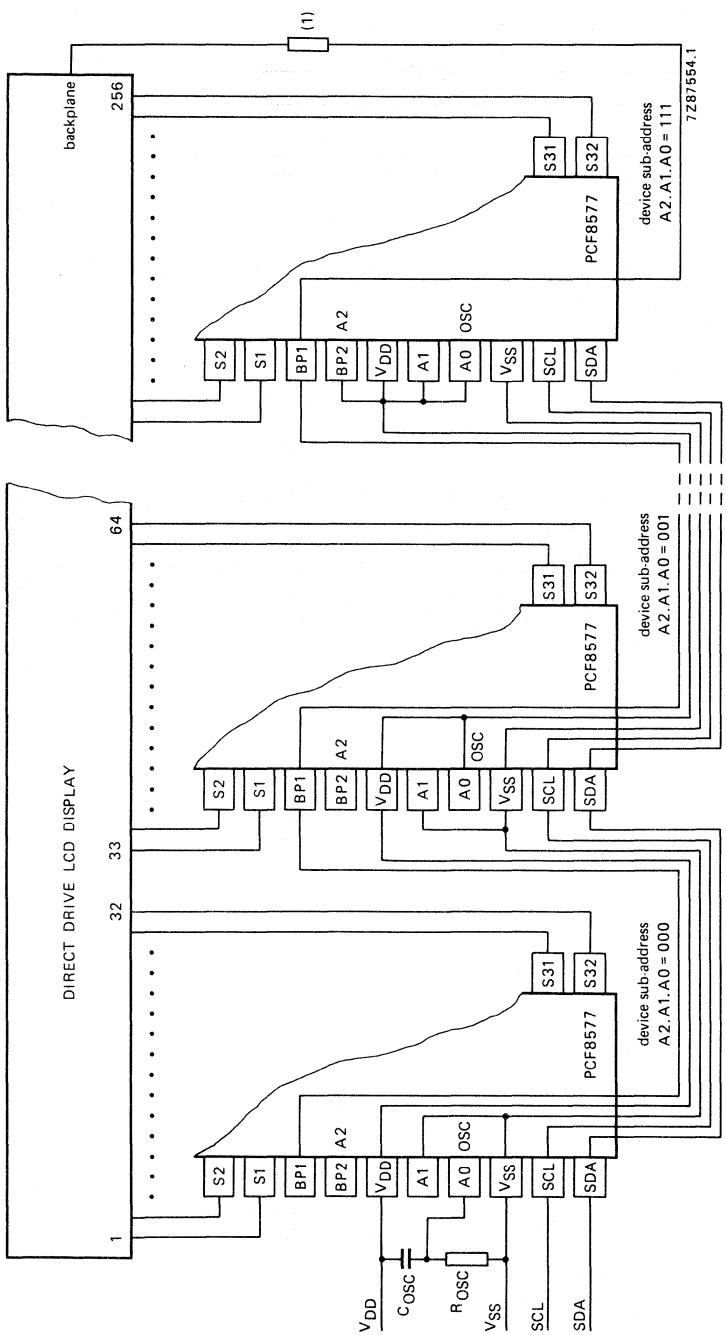
V_{DD} = 2,5 to 9 V; V_{SS} = 0 V; T_{amb} = −40 to + 85 °C unless otherwise specified

parameter	symbol	min.	typ.*	max.	unit
Supply voltage	V_{DD}	2,5	–	9,0	V
Supply current					
f_{SCL} = 100 kHz; no load; R_{OSC} = 1 MΩ	I_{DD}	–	80	250	μA
f_{SCL} = 0; no load; R_{OSC} = 1 MΩ; V_{DD} = 5 V; T_{amb} = 25 °C	I_{DD}	–	35	70	μA
Power-on-reset level**	V_{REF}	–	1,1	2,0	V
Input SCL; input/output SDA					
input voltage LOW	V_{IL}	0	–	0,8	V
input voltage HIGH	V_{IH}	2,0	–	9,0	V
output current LOW at V_{OL} = 0,4 V	I_{OL}	3,0	–	–	mA
output leakage current HIGH at V_{OH} = V_{DD}	I_{OH}	–	–	250	nA
tolerable spike width on bus	t_{sw}	–	–	100	ns
input capacitance at V_I = V_{SS}	C_I	–	–	7	pF
A1 input leakage current at V_I = V_{SS} or V_{DD}	I_I	–	–	250	nA
A2/BP2 input current at V_I = V_{DD}	I_I	–	2,0	–	μA
A0/OSC input current at V_I = V_{SS} or V_{DD}	$\pm I_I$	–	5,0	–	μA
DC component of LCD driver	$\pm V_{BP}$	–	20	–	mV
Segment loads	C_{SX}	–	–	5	nF
	R_{SX}	1	–	–	MΩ
Segment output current at V_{OL} = 0,4 V; V_{DD} = 5 V	I_{OL}	0,3	–	–	mA
Segment output current at V_{OH} = V_{DD} − 0,4 V; V_{DD} = 5 V	$-I_{OH}$	0,3	–	–	mA
Backplane load (direct drive)	C_{BP}	–	–	50	nF
	R_{BP}	100	–	–	kΩ
Backplane loads (duplex drive)	C_{BP}	–	–	35	nF
	R_{BP}	100	–	–	kΩ
Rise and fall times (V_{BP} − V_{SX}) at maximum load	t_r, t_f	–	–	200	μs
Display frequency at C_{OSC} = 680 pF; R_{OSC} = 1 MΩ	f_{LCD}	65	90	120	Hz

* V_{DD} = 5 V; T_{amb} = 25 °C.

** The power-on-reset circuit resets the I²C bus logic with $V_{DD} < V_{REF}$.

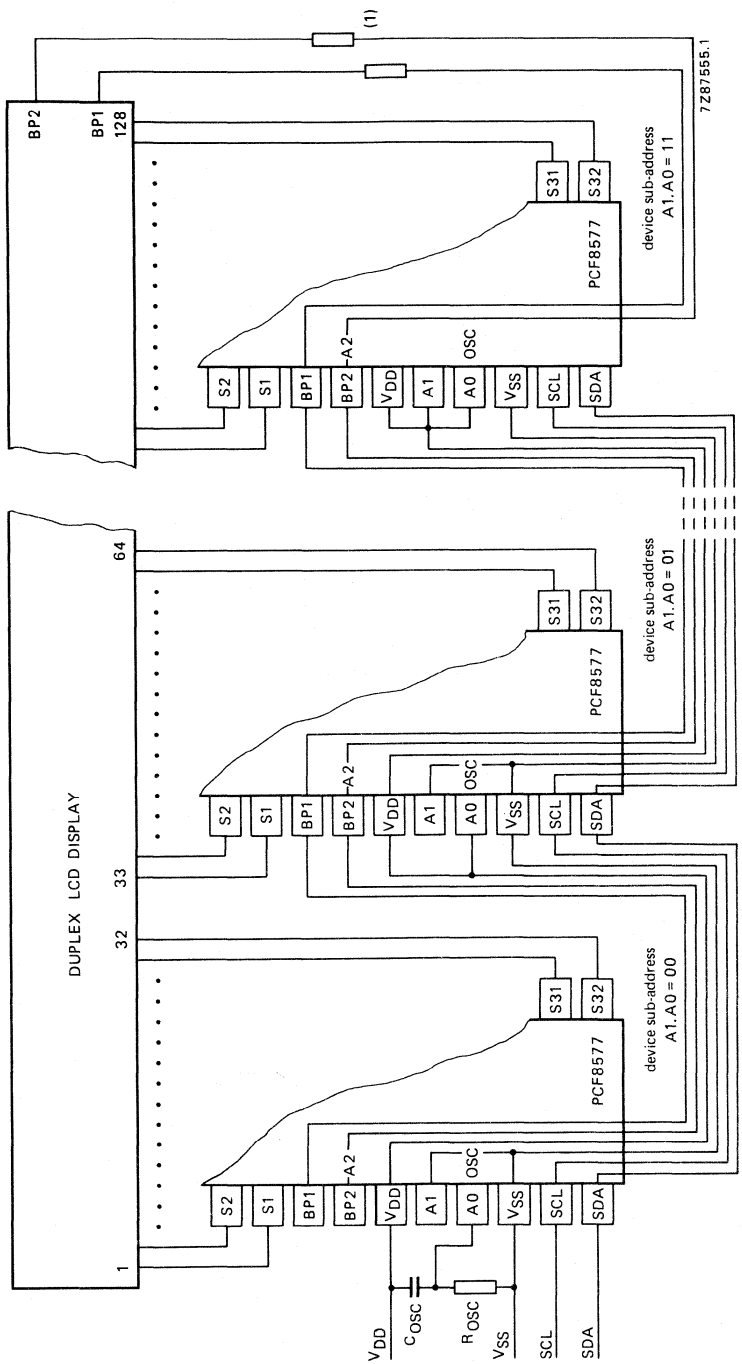
APPLICATION INFORMATION



(1) The series resistance of the display backplane must be greater than 1 Ω.

Fig. 16 Direct drive display; expansion to 256 segments using eight PCF8577.

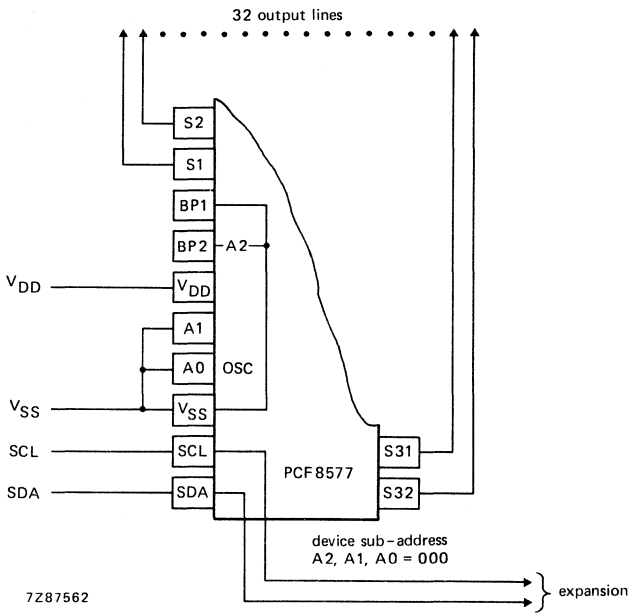
DEVELOPMENT DATA



(1) The series resistances of the display backplanes must be greater than 1 kΩ.

Fig. 17 Duplex display; expansion to 2 x 128 segments using four PCF8577.

APPLICATION INFORMATION (continued)



Notes

1. MODE bit must always be set to 0 (direct drive)
2. BANK switching is permitted
3. BP1 must always be connected to V_{SS} and A0/OSC must be connected to either V_{DD} or V_{SS} (no LCD modulation)

Fig. 18 Use of PCF8577 as 32-bit output expander in I^2C bus application.

Purchase of Philips' I^2C components conveys a license under the Philips' I^2C patent to use the components in the I^2C-system provided the system conforms to the I^2C specifications defined by Philips.

256 x 8-BIT STATIC RAM WITH ALARM CLOCK/CALENDAR AND BCD-COUNTER

GENERAL DESCRIPTION

The PCF8583 is a low power 2048-bit static CMOS RAM organized as 256 words by 8-bits. Addresses and data are transferred serially via a two-line bidirectional bus (I^2C). The built-in word address register is incremented automatically after each written or read data byte. One address pin A0 is used for programming the hardware address, allowing the connection of two devices to the bus without additional hardware. The built-in 32,768 kHz oscillator circuit and the first 8 bytes of the RAM are used for the clock/calendar and counter functions. The next 8 bytes may be programmed as alarm registers or used as free RAM space.

Features

- I^2C bus interface operating supply voltage: 2,5 V to 6 V
- Clock operating supply voltage (0 to 70 °C): 1,0 V to 6 V
- Data retention voltage: 1,0 V to 6 V
- Low standby current: max. 15 μA
- Clock function with four year calendar
- 24 or 12 hour format
- 32,768 kHz or 50 Hz time base
- Serial input/output bus (I^2C)
- Automatic word address incrementing
- Programmable alarm, timer and interrupt function

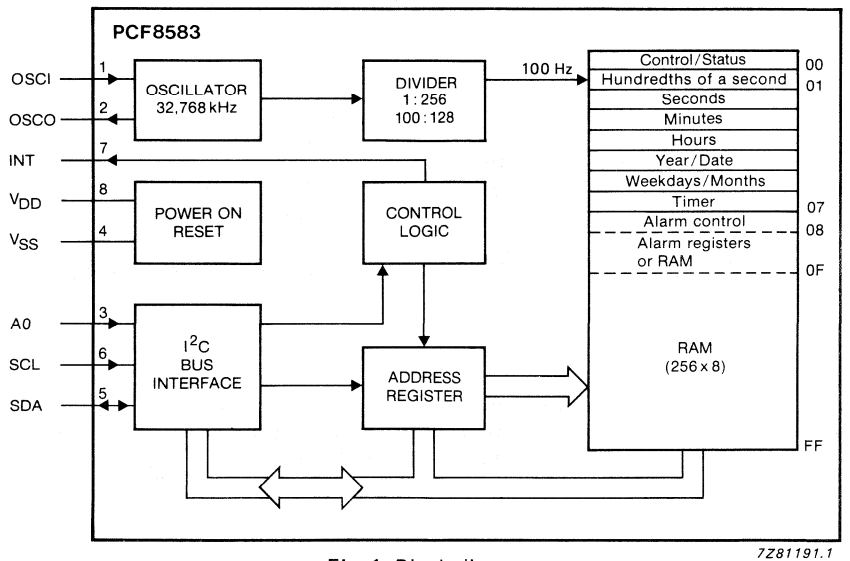


Fig. 1 Block diagram.

PACKAGE OUTLINES

PCF8583P: 8-lead DIL; plastic (SOT-97AE).
PCF8583T: 8-lead mini-pack; plastic (SO-8L; SOT-176).

PINNING

1	OSCI	oscillator input, 50 Hz or event-pulse input	
2	OSCO	oscillator output	
3	A0	address input	
4	V_{SS}	negative supply	
5	SDA	serial data line	I^2C bus
6	SCL	serial clock line	I^2C bus
7	INT	open drain interrupt output (active low)	
8	V_{DD}	positive supply	

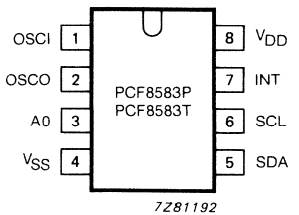


Fig. 2 Pinning diagram.

RATINGS

Limiting values in accordance with the Absolute Maximum System (IEC 134)

Supply voltage range (pin 8); note 1	V_{DD}		−0,8 to 8,0	V
Voltage range on any input	V_I		−0,8 to V_{DD} + 0,8	V
D.C. input current (any input)	I_I	max.	10	mA
D.C. output current (any output)	I_O	max.	10	mA
Supply current (pin 4 or pin 8)	I_{DD}; I_{SS}	max.	50	mA
Power dissipation per package	P_{tot}	max.	300	mW
Power dissipation per output	P	max.	50	mW
Storage temperature range	T_{stg}		−65 to + 150	°C
Operating ambient temperature range	T_{amb}		−40 to + 85	°C

Note

1. Inputs and outputs are protected against electrostatic discharges in normal handling. However, to be totally safe, it is advised to take handling precautions appropriate to handling MOS devices (see 'Handling MOS devices').

FUNCTIONAL DESCRIPTION

The PCF8583 contains a 256 by 8-bit RAM with an 8-bit auto-increment address register, an on-chip 32,768 kHz oscillator circuit, a frequency divider, a serial two-line bidirectional I^2C bus interface and a power-on reset circuit.

The first 8 bytes of the RAM (memory addresses 00 to 07) are designed as addressable 8-bit parallel registers. The first register (memory address 00) is used as a control/status register. The memory addresses 01 to 07 are used as counters for the clock function. The memory addresses 08 to 0F are free RAM locations or may be programmed as alarm registers.

Counter function modes

When the control/status register is set a 32,768 kHz clock mode, a 50 Hz clock mode or an event-counter mode can be selected.

In the clock modes the hundredths of a second, seconds, minutes, hours, date, month (four year calendar) and weekdays are stored in a BCD format. The timer register stores up to 99 days. The event-counter mode is used to count pulses applied to the oscillator input (OSCO left open). The event counter stores up to 6 digits of data.

When one of the counters is read (memory locations 01 to 07), the contents of all counters are strobed into capture latches at the beginning of a read cycle. Therefore faulty reading of the count during a carry condition is prevented.

Alarm function modes

By setting the alarm enable bit of the control/status register the alarm control register (address 08) is activated.

By setting the alarm control register a dated alarm, a daily alarm, a weekday alarm or a timer alarm may be programmed. In the clock modes, the timer register (address 07) may be programmed to count hundredths of a second, seconds, minutes, hours or days. Days are counted when an alarm is not programmed.

Whenever an alarm event occurs the alarm flag of the control/status register is set. A timer alarm event will set the alarm flag and an overflow condition of the timer will set the timer flag. The open drain interrupt output is switched on (active LOW) when the alarm or timer flag is set (enabled).

When a timer function without any alarm function is programmed the remaining alarm registers (addresses 09 to 0F) may be used as free RAM space.

Control/status register

The control/status register is defined as the memory location 00 with free access for reading and writing via the I^2C bus. All functions and options are controlled by the contents of the control/status register (see Fig. 3).

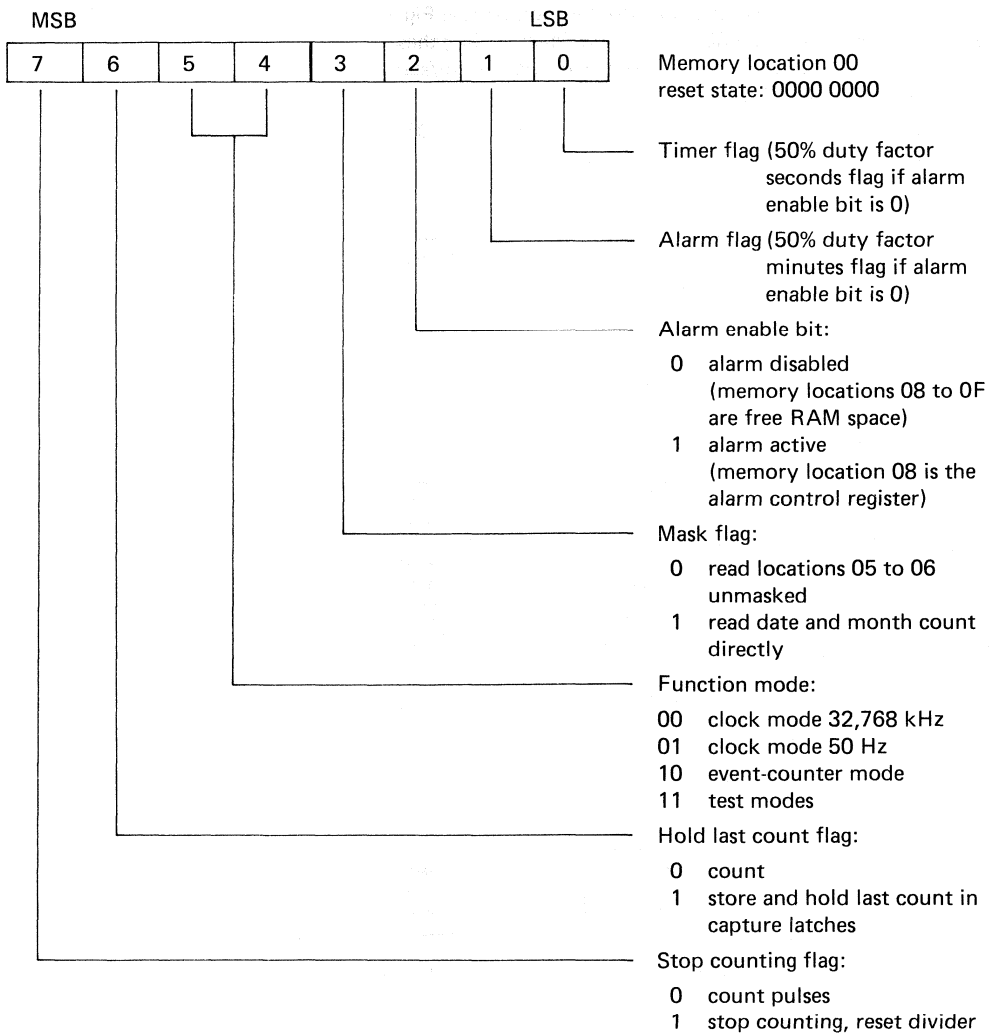


Fig. 3 Control/status register.

DEVELOPMENT DATA

Counter registers

In the different modes the counter registers are programmed and arranged as shown in Fig. 4. Counter cycles are listed in Table 1.

In the clock modes 24 h or 12 h format can be selected by setting the most significant bit of the hours counter register. The format of the hours counter is shown in Fig. 5.

The year and date are packed into memory location 05 (see Fig. 6). The weekdays and months are packed into memory location 06 (see Fig. 7). When reading these memory locations the year and weekdays are masked out when the mask flag of the control/status register is set. This allows the user to read the date and month count directly.

In the event-counter mode events are stored in BCD format. D5 is the most significant and D0 the least significant digit. The divider is by-passed.

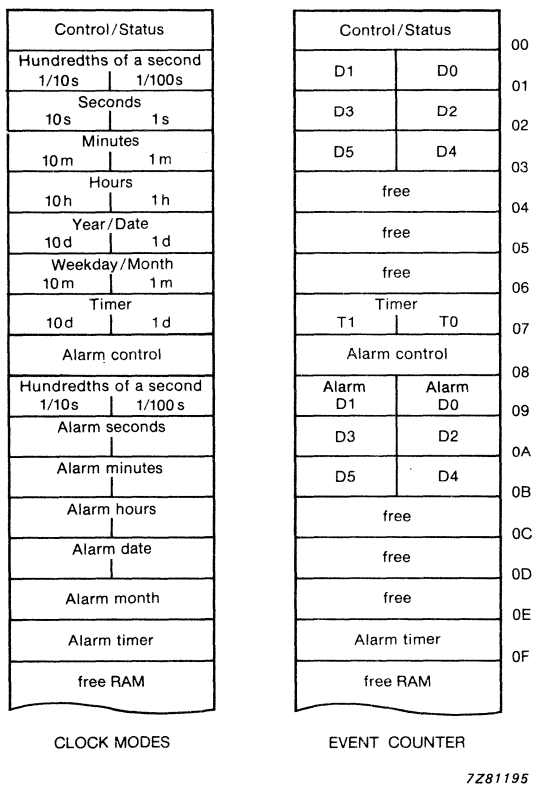
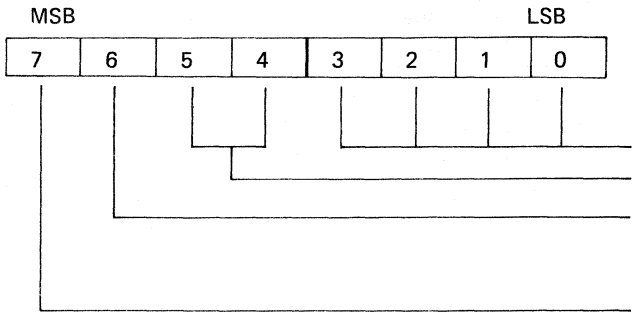


Fig. 4 Register arrangement.

Counter registers (continued)



Memory location 04 (hours counter)
reset state: 0000 0000

Unit hours BCD

Ten hours (0 to 2 binary)

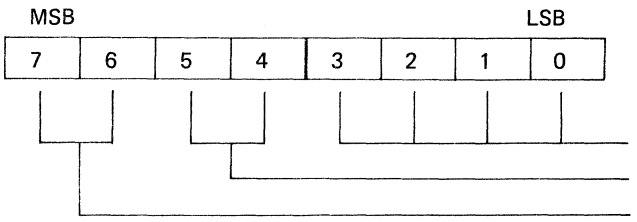
AM/PM flag:

0 AM
1 PM

Format:

0 24 h format, AM/PM flag remains unchanged
1 12 h format, AM/PM flag will be updated

Fig. 5 Format of the hours counter.



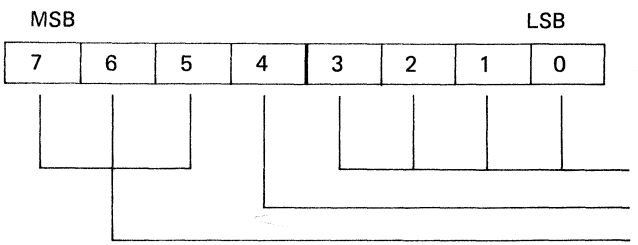
Memory location 05 (year/date)
reset state: 0000 0001

Unit days BCD

Ten days (0 to 3 binary)

Year (0 to 3 binary, read as 0 if the mask flag is set)

Fig. 6 Format of the year/date counter.



Memory location 06 (weekdays/months)
reset state: 0000 0001

Unit months BCD

Ten months

Weekdays (0 to 6 binary, read as 0 if the mask flag is set)

Fig. 7 Format of the weekdays/months counter.

Table 1 Cycle length of the time counters, clock modes

unit	counting cycle	carry to the next unit	contents of the month counter
hundredths of a second	00 to 99	99 to 00	
seconds	00 to 59	59 to 00	
minutes	00 to 59	59 to 00	
hours (24 h)	00 to 23	23 to 00	
hours (12 h)	12 AM, 01 AM to 11 AM, 12 PM, 01 PM to 11 PM	11 PM to 12 AM	
date	01 to 31	31 to 01	1, 3, 5, 7, 8, 10, 12
	01 to 30	30 to 01	4, 6, 9, 11
	01 to 29	29 to 01	2, year = 0
	01 to 28	28 to 01	2, year = 1, 2, 3
months	01 to 12	12 to 01	
year	0 to 3		
weekdays	0 to 6	6 to 0	
timer/days	00 to 99	no carry	

DEVELOPMENT DATA

Alarm control register

When the alarm enable bit of the control/status register is set the alarm control register (address 08) is activated. All alarm, timer and interrupt output functions are controlled by the contents of the alarm control register (see Figs 8a and 8b).

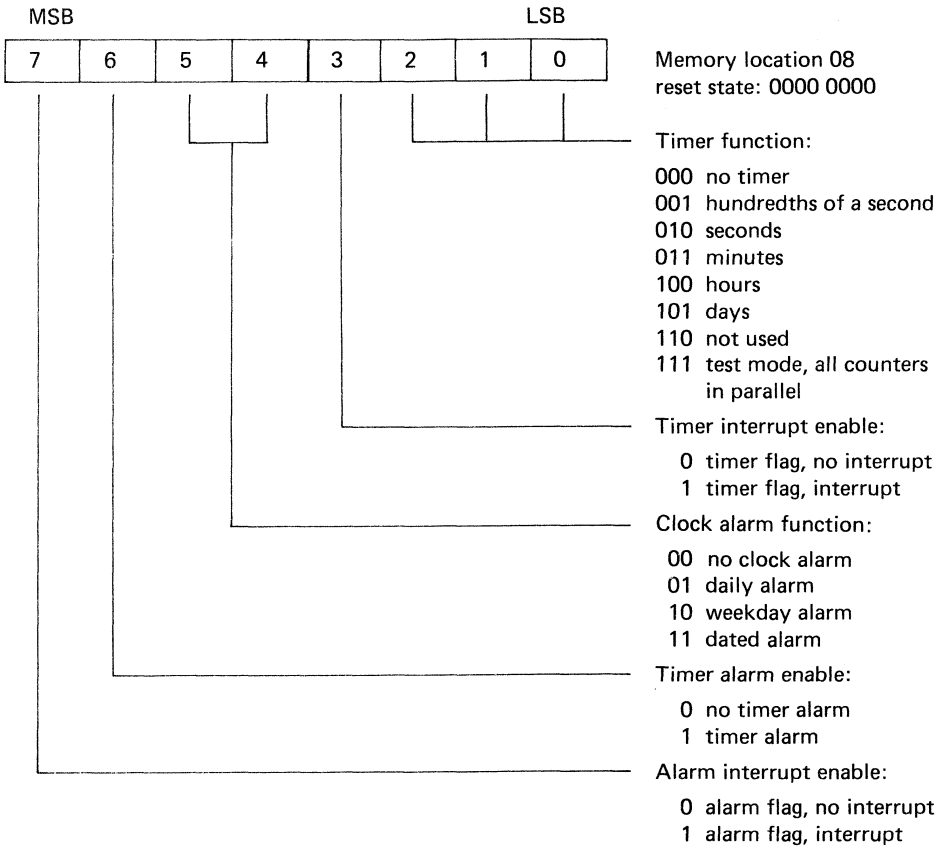
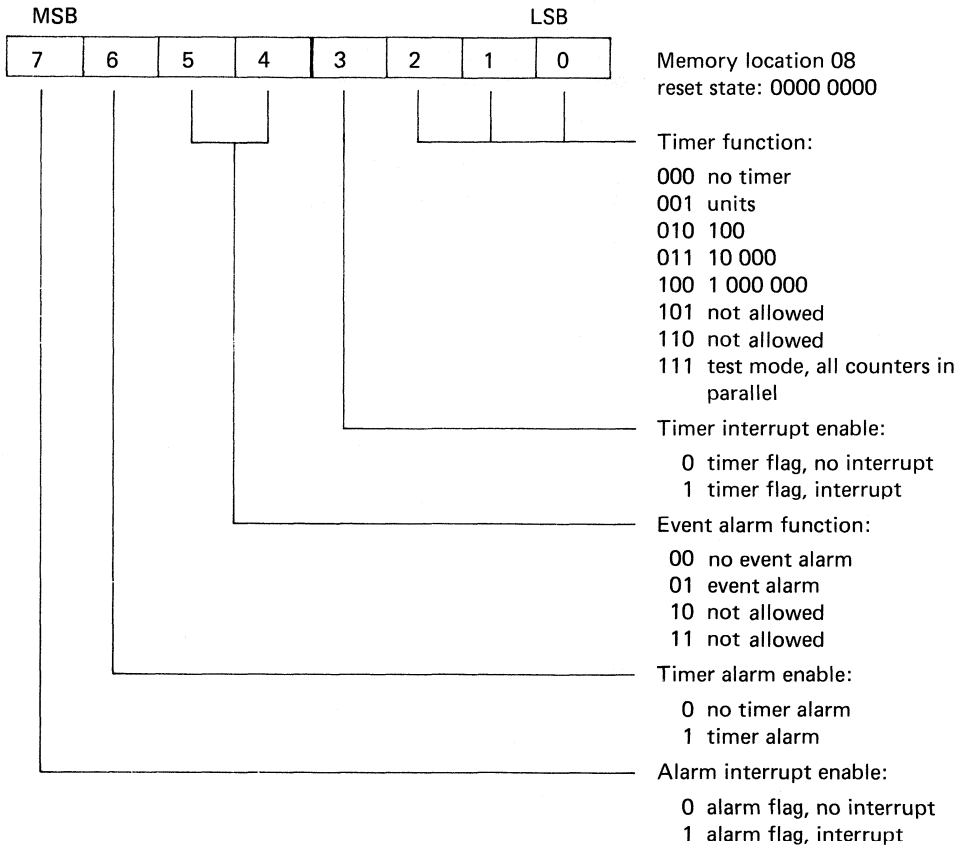


Fig. 8a Alarm control register, clock modes.



DEVELOPMENT DATA

Fig. 8b Alarm control register, event-counter mode.

Alarm registers

All alarm registers are allocated with a constant address offset of hex 08 to the corresponding counter registers.

An alarm goes off when the contents of the alarm registers matches bit-by-bit the contents of the involved counter registers. The year and weekday bits are ignored in a dated alarm. A daily alarm ignores the month and date bits. When a weekday alarm is selected, the contents of the alarm weekday/month register will select the weekdays on which an alarm is activated (see Fig. 9).

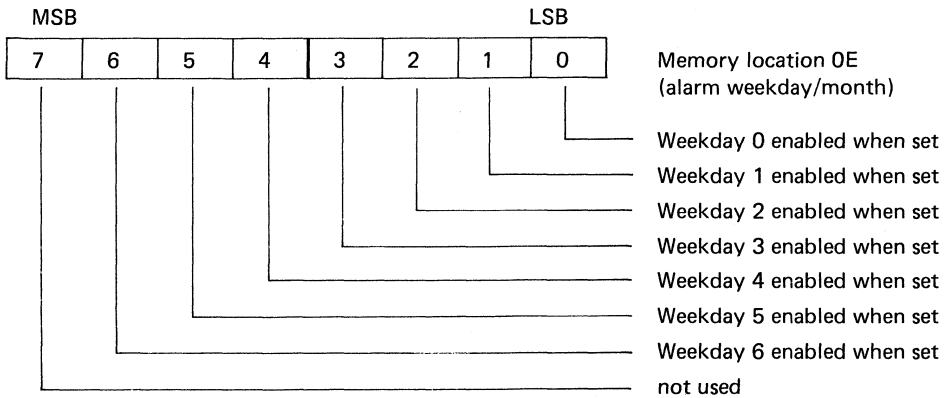


Fig. 9 Selection of alarm weekdays.

Interrupt output

The open-drain n-channel interrupt output is programmed by setting the alarm control register. It is switched on (active LOW) when the alarm flag or the timer flag is set. In the clock mode without alarm the output sequence is controlled by the timer flag. The OFF voltage of the interrupt output may exceed the supply voltage.

Oscillator and divider

A 32,768 kHz quartz crystal has to be connected to OSCI (pin 1) and OSCO (pin 2). A trimmer capacitor between OSCI and V_{DD} is used for tuning the oscillator. The oscillator frequency is scaled down to 128 Hz by the divider. A 100 Hz clock signal is derived from this signal.

In the 50 Hz clock mode or event-counter mode the oscillator is disabled and the oscillator input is switched to a high impedance state. This allows the user to feed the 50 Hz reference frequency or an external high speed event signal into the input OSCI.

Initialization

When power-up occurs the I^2C bus interface, the control/status register and all clock counters are reset. The device starts time keeping in the 32,768 kHz clock mode with the 24 h format on the first of January at 0.00.00: 00.

A second level-sensitive reset signal to the I^2C bus interface is generated as soon as the supply voltage drops below the interface reset level. This reset signal does not affect the control/status or clock counter registers.

It is recommended to set the stop counting flag of the control/status register before loading the actual time into the counters. Loading of illegal states will lead to a clock malfunction but will not latch-up the device.

CHARACTERICS OF THE I²C BUS

The I²C bus is for bidirectional, two-line communication between different ICs or modules. The two lines are a serial data line (SDA) and a serial clock line (SCL). Both lines must be connected to a positive supply via a pull-up resistor. Data transfer may be initiated only when the bus is not busy.

Bit transfer

One data bit is transferred during each clock pulse. The data on the SDA line must remain stable during the HIGH period of the clock pulse as changes in the data line at this time will be interpreted as a control signal.

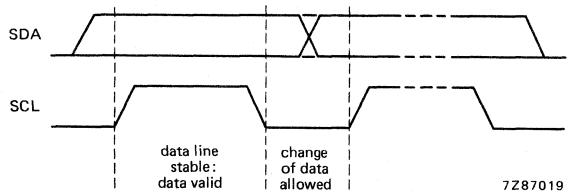


Fig. 10 Bit transfer.

Start and stop conditions

Both data and clock lines remain HIGH when the bus is not busy. A HIGH-to-LOW transition of the data line, while the clock is HIGH, is defined as the start condition (S). A LOW-to-HIGH transition of the data line while the clock is HIGH, is defined as the stop condition (P).

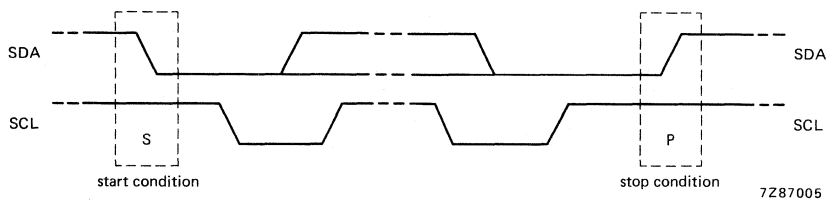


Fig. 11 Definition of start and stop condition.

System configuration

A device generating a message is a "transmitter", a device receiving a message is the "receiver". The device that controls the message is the "master" and the devices which are controlled by the master are the "slaves".

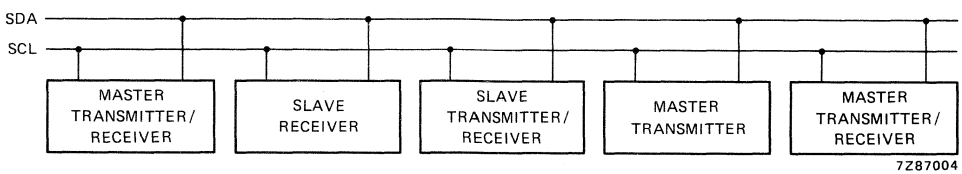


Fig. 12 System configuration.

Acknowledge

The number of data bytes transferred between the start and stop conditions from transmitter to receiver is not limited. Each data byte of eight bits is followed by one acknowledge bit. The acknowledge bit is a HIGH level put on the bus by the transmitter whereas the master also generates an extra acknowledge related clock pulse. A slave receiver which is addressed must generate an acknowledge after the reception of each byte. Also a master must generate an acknowledge after the reception of each byte that has been clocked out of the slave transmitter. The device that acknowledge has to pull down the SDA line during the acknowledge clock pulse, so that the SDA line is stable LOW during the HIGH period of the acknowledge related clock pulse. A master receiver must signal an end of data to the transmitter by *not* generating an acknowledge on the last byte that has been clocked out of the slave. In this event the transmitter must leave the data line HIGH to enable the master to generate a stop condition.

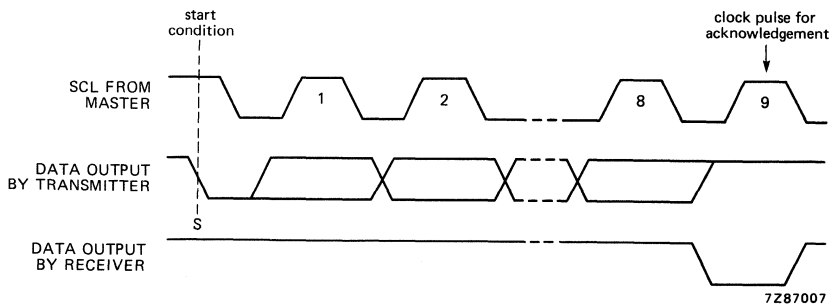


Fig. 13 Acknowledgement on the I²C bus.

DEVELOPMENT DATA

Timing specifications

All the timing values are valid within the operating supply voltage and ambient temperature range and refer to V_{IL} and V_{IH} with an input voltage swing of V_{SS} to V_{DD}.

parameter	symbol	min.	typ.	max.	unit
SCL clock frequency	f_{SCL}	–	–	100	kHz
Tolerable spike width on bus	t_{SW}	–	–	100	ns
Bus free time	t_{BUF}	4,0	–	–	μs
Start condition set-up time	$t_{SU;STA}$	4,0	–	–	μs
Start condition hold time	$t_{HD;STA}$	4,7	–	–	μs
SCL LOW time	t_{LOW}	4,7	–	–	μs
SCL HIGH time	t_{HIGH}	4,0	–	–	μs
SCL and SDA rise time	t_R	–	–	1,0	μs
SCL and SDA fall time	t_F	–	–	0,3	μs
Data set-up time	$t_{SU;DAT}$	250	–	–	ns
Data hold time	$t_{HD;DAT}$	0	–	–	ns
SCL LOW to data out valid	$t_{VD;DAT}$	–	–	3,4	μs
Stop condition set-up time	$t_{SU;STO}$	4,0	–	–	μs

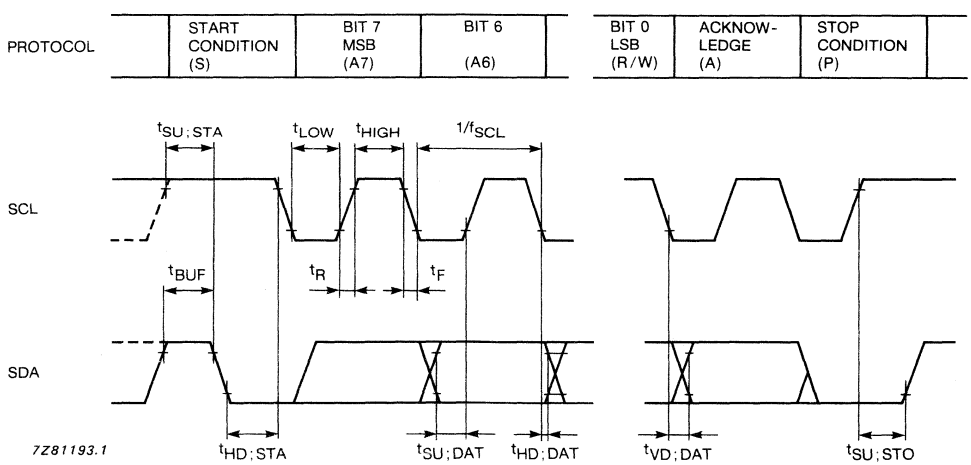


Fig. 14 I²C bus timing diagram.

I²C bus protocol

Before any data is transmitted on the I²C bus, the device which should respond is addressed first. The addressing is always done with the first byte transmitted after the start procedure. The I²C bus configuration for the different PCF8583 READ and WRITE cycles is shown in Fig. 15.

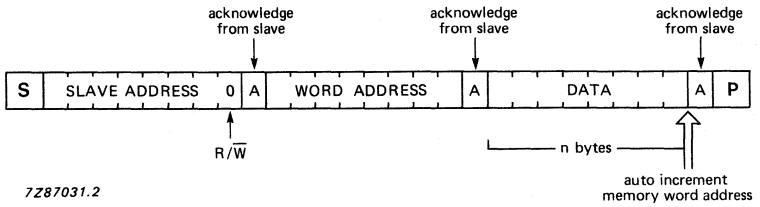


Fig. 15a Master transmits to slave receiver (WRITE mode).

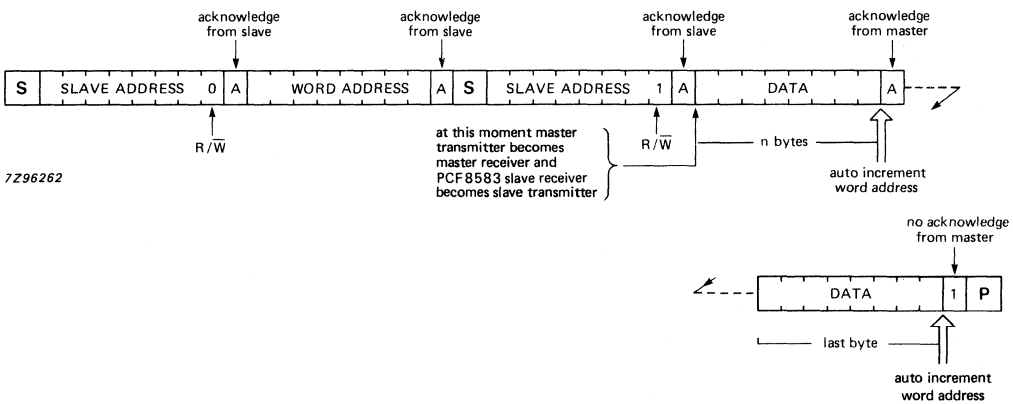


Fig. 15b Master reads after setting word address (WRITE word address; READ data).

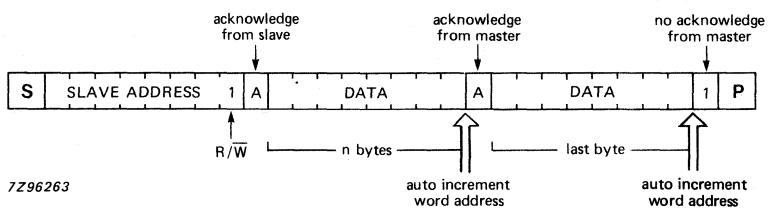


Fig. 15c Master reads slave immediately after first byte (READ mode).

DEVELOPMENT DATA

CHARACTERISTICS

V_{DD} = 2,5 to 6,0 V; V_{SS} = 0 V; T_{amb} = −40 to + 85 °C unless otherwise specified

parameter	symbol	min.	typ.	max.	unit
Supply					
Supply voltage	V_{DD}	2,5	–	6	V
Supply current at $V_I = V_{SS}$ or V_{DD}					
operating at f_{SCL} = 100 kHz	I_{DD}	–	–	200	μA
standby at f_{SCL} = 0 kHz	I_{DDO}	–	–	15	μA
standby at T_{amb} = −25 to + 70 °C	I_{DDO}	–	–	5	μA
Power-on reset voltage level (note 1)	V_{POR}	1,5	1,9	2,3	V
Inputs; input/output SDA					
Input voltage LOW (note 2)	V_{IL}	−0,8	–	0,3 x V_{DD}	V
Input voltage HIGH (note 2)	V_{IH}	0,7 x V_{DD}	–	V_{DD} + 0,8	V
Output current LOW at V_{OL} = 0,4 V	I_{OL}	3	–	–	V
Output leakage current HIGH at $V_{OH} = V_{DD}$	I_{OH}	–	–	250	nA
Input leakage current at $V_I = V_{DD}$ or V_{SS}	$\pm I_I$	–	–	250	nA
Input capacitance (SCL, SDA) at $V_I = V_{SS}$	C_I	–	–	7	pF
LOW V_{DD} data retention					
Supply voltage for data retention	V_{DDR}	1	–	6	V
Supply current at V_{DDR} = 1 V	I_{DDR}	–	–	5	μA
Supply current at V_{DDR} = 1 V; T_{amb} = −25 to + 70 °C	I_{DDR}	–	–	2	μA
Oscillator					
Integrated oscillator capacitance	C_{OSC}	–	40	–	pF
Oscillator stability for: ΔV_{DD} = 100 mV at V_{DD} = 1,5 V; T_{amb} = 25 °C	f/f_{OSC}	–	2×10^{-6}	–	–

CHARACTERISTICS (continued)

parameter	symbol	min.	typ.	max.	unit
Quartz crystal parameters					
Frequency = 32,768 kHz					
Series resistance	R_S	–	–	40	KΩ
Parallel capacitance	C_L	–	9	–	pF
Trimmer capacitance	C_T	5	–	25	pF

Notes to characteristics

1. The power-on reset circuit resets the I^2C bus logic when $V_{DD} < V_{POR}$.
2. When the voltages are a diode voltage above or below the supply voltage V_{DD} or V_{SS} an input current will flow; this current must not exceed ± 0,5 mA.

APPLICATION INFORMATION

The PCF8583 slave address has a fixed combination 1010 as group 1.

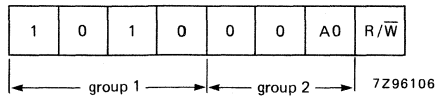


Fig. 16 PCF8583 address.

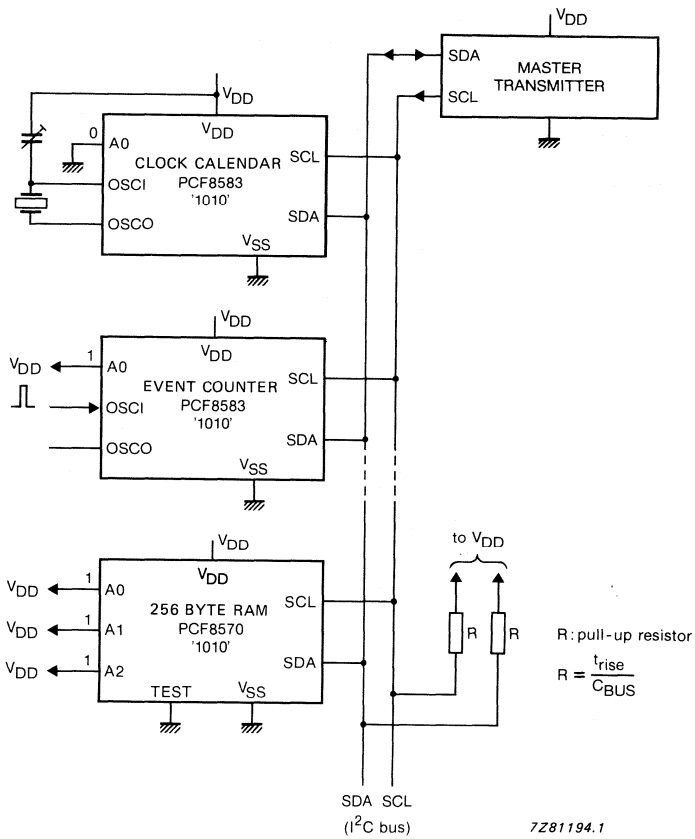


Fig. 17 PCF8583 application diagram.

DEVELOPMENT DATA

8-BIT A/D AND D/A CONVERTER

GENERAL DESCRIPTION

The PCF8591 is a single chip, single supply low power 8-bit CMOS data acquisition device with four analogue inputs, one analogue output and a serial I^2C bus interface. Three address pins A0, A1 and A2 are used for programming the hardware address, allowing the use of up to eight devices connected to the I^2C bus without additional hardware. Address, control and data to and from the device are transferred serially via the two-line bidirectional bus (I^2C).

The functions of the device include analogue input multiplexing, on-chip track and hold function, 8-bit analogue-to-digital conversion and an 8-bit digital-to-analogue conversion. The maximum conversion rate is given by the maximum speed of the I^2C bus.

FEATURES

- Single power supply
- Operating supply voltage 2,5 V to 6 V
- Low standby current
- Serial input/output via I^2C bus
- Address by 3 hardware address pins
- Sampling rate given by I^2C bus speed
- 4 analogue inputs programmable as single-ended or differential inputs
- Auto-incremented channel selection
- Analogue voltage range from V_{SS} to V_{DD}
- On-chip track and hold circuit
- 8-bit successive approximation A/D conversion
- Multiplying DAC with one analogue output

APPLICATIONS

Closed loop control systems; low power converter for remote data acquisition; battery operated equipment; acquisition of analogue values in automotive, audio and TV applications.

PACKAGE OUTLINES

PCF8591P:16-lead DIL; plastic (SOT-38).
PCF8591T:16-lead mini-pack; plastic (SO-16L; SOT-162A).

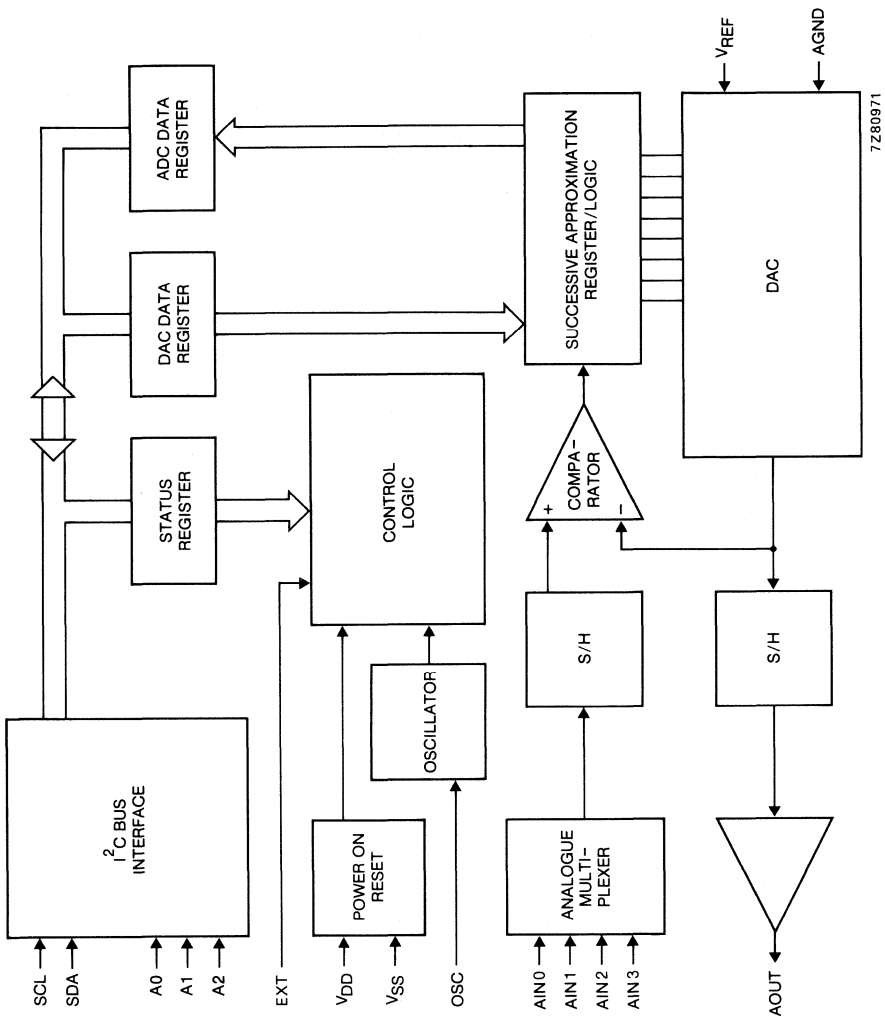


Fig. 1 Block diagram.

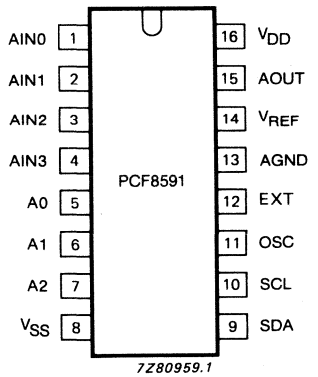


Fig. 2 Pinning diagram.

PINNING

1.	AIN0	analogue inputs (A/D converter)
2.	AIN1	
3.	AIN2	
4.	AIN3	
5.	A0	hardware address
6.	A1	
7.	A2	
8.	V_{SS}	negative supply voltage
9.	SDA	I^2C bus data input/output
10.	SCL	I^2C bus clock input/output
11.	OSC	oscillator input/output
12.	EXT	external/internal switch for oscillator input
13.	AGND	analogue ground
14.	V_{REF}	voltage reference input
15.	AOUT	analogue output (D/A converter)
16	V_{DD}	positive supply voltage

FUNCTIONAL DESCRIPTION

DEVELOPMENT DATA

Adressing

Each PCF8591 device in an I^2C bus system is activated by sending a valid address to the device. The address consists of a fixed part and a programmable part. The programmable part must be set according to the address pins A0, A1 and A2. The address always has to be sent as the first byte after the start condition in the I^2C bus protocol. The last bit of the address byte is the read/write-bit which sets the direction of the following data transfer (see Figs 3 and 10).

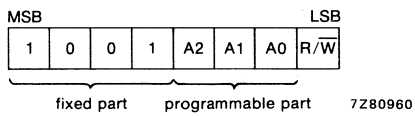


Fig. 3 Address byte.

Control byte

The second byte sent to a PCF8591 device will be stored in its control register and is required to control the device function.

The upper nibble of the control register is used for enabling the analogue output, and for programming the analogue inputs as single-ended or differential inputs. The lower nibble selects one of the analogue input channels defined by the upper nibble (see Fig. 4). If the auto-increment flag is set the channel number is incremented automatically after each A/D conversion.

The selection of a non-existing input channel results in the highest available channel number being allocated. Therefore, if the auto-increment flag is set, the next selected channel will be always channel 0. The most significant bits of both nibbles are reserved for future functions and have to be set to 0. After a power-on reset condition all bits of the control register are reset to 0. The D/A converter and the oscillator are disabled for power saving. The analogue output is switched to a high impedance state.

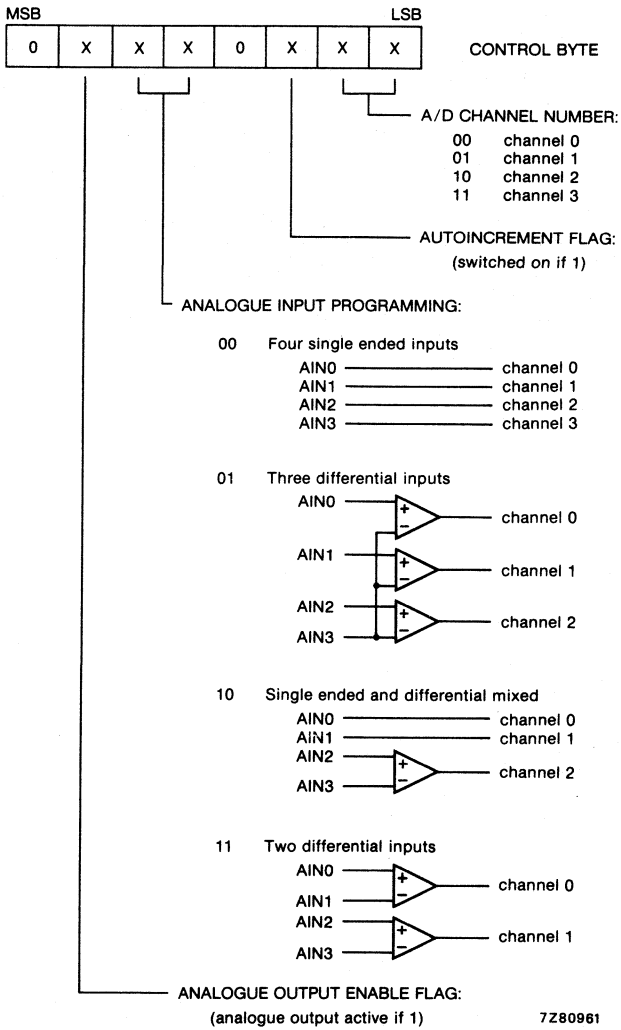


Fig. 4 Control byte.

DEVELOPMENT DATA

D/A conversion

The third byte sent to a PCF8591 device is stored in the DAC data register and is converted to the corresponding analogue voltage using the on-chip D/A converter. This D/A converter consists of a resistor divider chain connected to the external reference voltage with 256 taps and selection switches. The tap-decoder switches one of these taps to the DAC output line (see Fig. 5).

The analogue output voltage is buffered by an auto-zeroed unity gain amplifier. This buffer amplifier may be switched on or off by setting the analogue output enable flag of the control register. In the active state the output voltage is held until a further data byte is sent.

The on-chip D/A converter is also used for successive approximation A/D conversion. In order to release the DAC for an A/D conversion cycle the unity gain amplifier is equipped with a track and hold circuit. This circuit holds the output voltage while executing the A/D conversion.

The output voltage supplied to the analogue output AOUT is given by the formula shown in Fig. 6. The waveforms of a D/A conversion sequence are shown in Fig. 7.

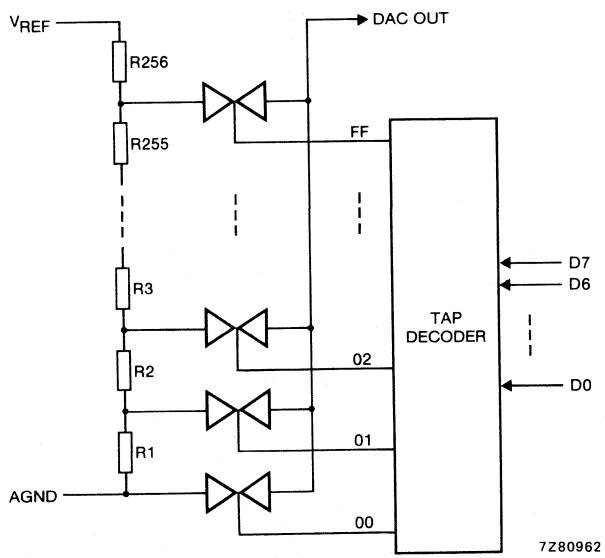


Fig. 5 DAC resistor divider chain.

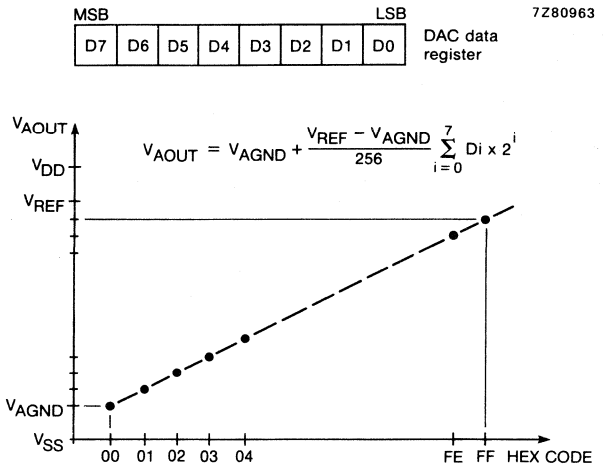


Fig. 6 DAC data and d.c. conversion characteristics.

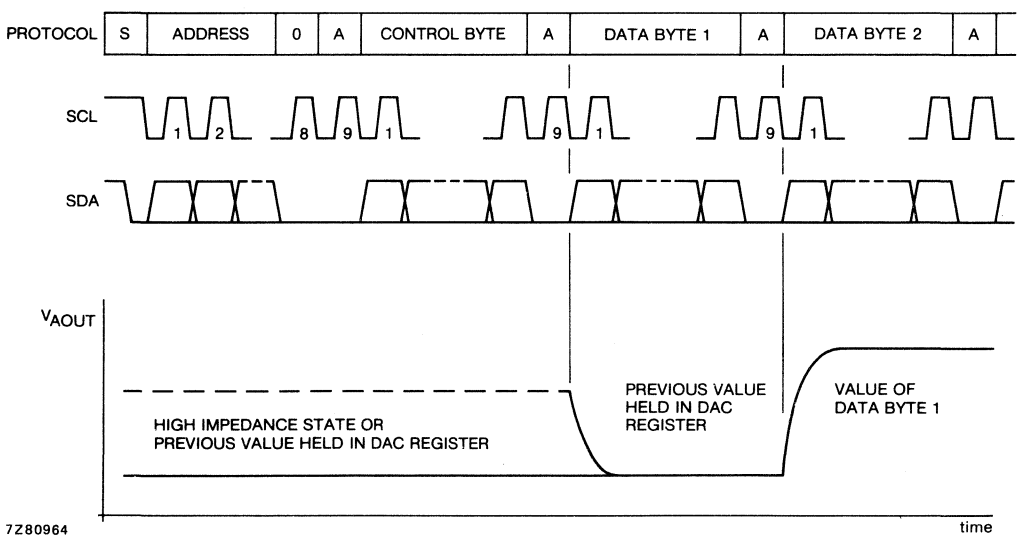


Fig. 7 D/A conversion sequence.

A/D conversion

The A/D converter makes use of the successive approximation conversion technique. The on-chip D/A converter and a high gain comparator are used temporarily during an A/D conversion cycle.

An A/D conversion cycle is always started after sending a valid read mode address to a PCF8591 device. The A/D conversion cycle is triggered at the trailing edge of the acknowledge clock pulse and is executed while transmitting the result of the previous conversion (see Fig. 8).

Once a conversion cycle is triggered an input voltage sample of the selected channel is stored on the chip and is converted to the corresponding 8-bit binary code. Samples picked up from differential inputs are converted to an 8-bit two's complement code (see Fig. 9). The conversion result is stored in the ADC data register and awaits transmission. If the auto-increment flag is set the next channel is selected.

The first byte transmitted in a read cycle contains the conversion result code of the previous read cycle. After a power-on reset condition the first byte read is a hexadecimal 80. The protocol of an I^2C bus read cycle is shown in Fig. 10.

The maximum A/D conversion rate is given by the actual speed of the I^2C bus.

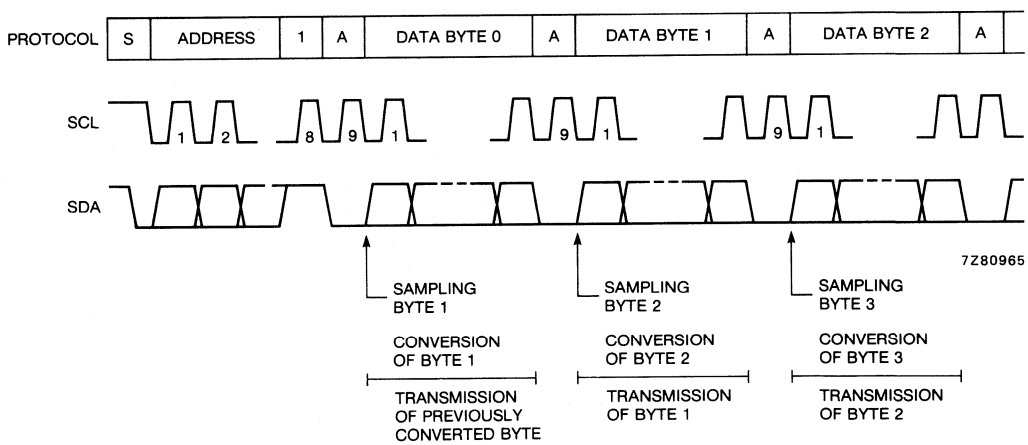


Fig. 8 A/D conversion sequence.

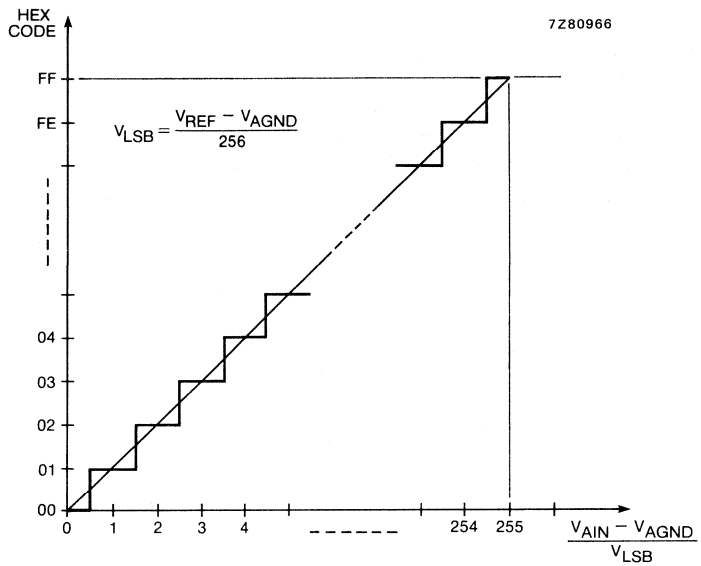


Fig. 9a A/D conversion characteristics of single-ended inputs.

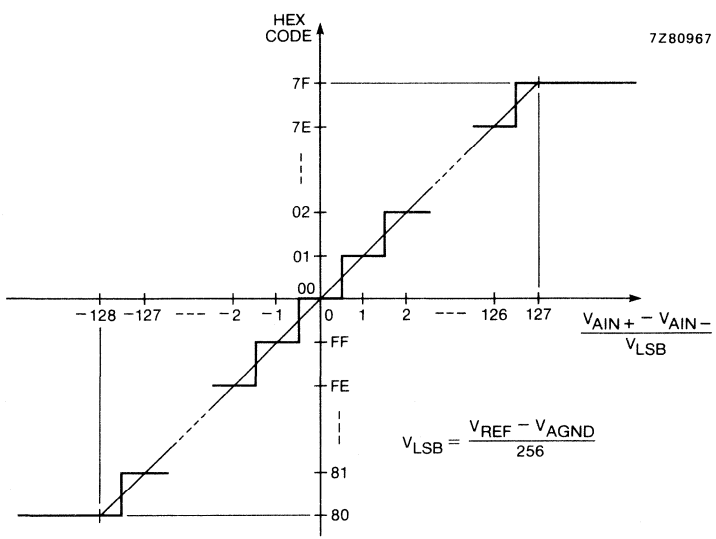


Fig. 9b A/D conversion characteristics of differential inputs.

Reference voltage

For the D/A and A/D conversion either a stable external voltage reference or the supply voltage has to be applied to the resistor divider chain (pins V_{REF} and AGND). The AGND pin has to be connected to the system analogue ground and may have a d.c. off-set with reference to V_{SS}.

A low frequency may be applied to the V_{REF} and AGND pins. This allows the use of the D/A converter as a one-quadrant multiplier; see Application Information and Fig. 6.

The A/D converter may also be used as a one or two quadrant analogue divider. The analogue input voltage is divided by the reference voltage. The result is converted to a binary code. In this application the user has to keep the reference voltage stable during the conversion cycle.

Oscillator

An on-chip oscillator generates the clock signal required for the A/D conversion cycle and for refreshing the auto-zeroed buffer amplifier. When using this oscillator the EXT pin has to be connected to V_{SS}. At the OSC pin the oscillator frequency is available.

If the EXT pin is connected to V_{DD} the oscillator output OSC is switched to a high impedance state allowing the user to feed an external clock signal to OSC.

Bus protocol

After a start condition a valid hardware address has to be sent to a PCF8591 device. The read/write bit defines the direction of the following single or multiple byte data transfer. For the format and the timing of the start condition (S), the stop condition (P) and the acknowledge bit (A) refer to the I^2C bus characteristics. In the write mode a data transfer is terminated by sending either a stop condition or the start condition of the next data transfer.

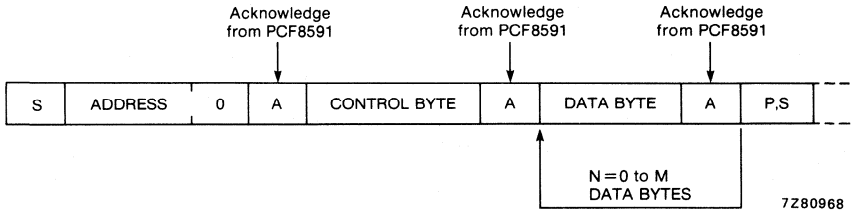


Fig. 10a Bus protocol for write mode, D/A conversion.

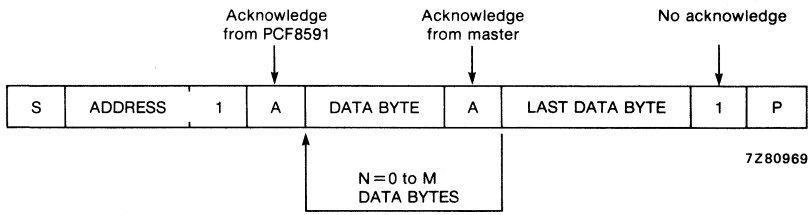


Fig. 10b Bus protocol for read mode, A/D conversion.

CHARACTERICS OF THE I²C BUS

The I²C bus is for bidirectional, two-line communication between different ICs or modules. The two lines are a serial data line (SDA) and a serial clock line (SCL). Both lines must be connected to a positive supply via a pull-up resistor. Data transfer may be initiated only when the bus is not busy.

Bit transfer

One data bit is transfered during each clock pulse. The data on the SDA line must remain stable during the HIGH period of the clock pulse as changes in the data line at this time will be interpreted as a control signal.

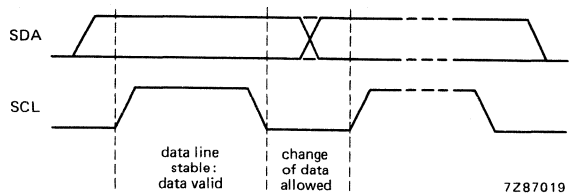


Fig. 11 Bit transfer.

Start and stop conditions

Both data and clock lines remain HIGH when the bus is not busy. A HIGH-to-LOW transition of the data line, while the clock is HIGH, is defined as the start condition (S). A LOW-to-HIGH transition of the data line while the clock is HIGH, is defined as the stop condition (P).

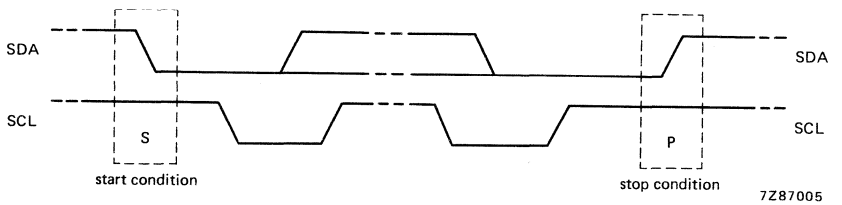


Fig. 12 Definition of start and stop condition.

System configuration

A device generating a message is a "transmitter", a device receiving a message is the "receiver". The device that controls the message is the "master" and the devices which are controlled by the master are the "slaves".

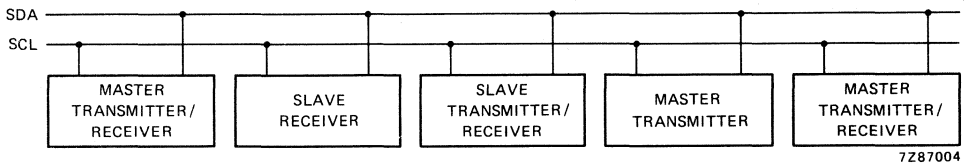


Fig. 13 System configuration.

Acknowledge.

The number of data bytes transfered between the start and stop conditions from transmitter to receiver is not limited. Each data byte of eight bits is followed by one acknowledge bit. The acknowledge bit is a HIGH level put on the bus by the transmitter whereas the master also generates an extra acknowledge related clock pulse. A slave receiver which is addressed must generate an acknowledge after the reception of each byte. Also a master must generate an acknowledge after the reception of each byte that has been clocked out of the slave transmitter. The device that acknowledges has to pull down the SDA line during the acknowledge clock pulse, so that the SDA line is stable LOW during the HIGH period of the acknowledge related clock pulse. A master receiver must signal an end of data to the transmitter by *not* generating an acknowledge on the last byte that has been clocked out of the slave. In this event the transmitter must leave the data line HIGH to enable the master to generate a stop condition.

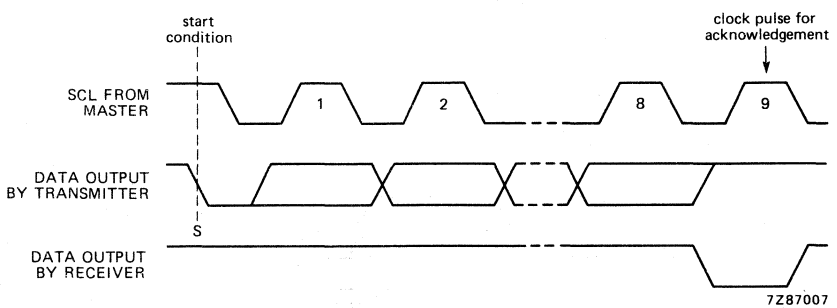


Fig. 14 Acknowledgement on the I^2C bus.

DEVELOPMENT DATA

Timing specifications

All the timing values are valid within the operating supply voltage and ambient temperature range and refer to V_{IL} and V_{IH} with an input voltage swing of V_{SS} to V_{DD}.

parameter	symbol	min.	typ.	max.	unit
SCL clock frequency	f_{SCL}	–	–	100	kHz
Tolerable spike width on bus	t_{SW}	–	–	100	ns
Bus free time	t_{BUF}	4,0	–	–	μs
Start condition set-up time	$t_{SU;STA}$	4,0	–	–	μs
Start condition hold time	$t_{HD;STA}$	4,7	–	–	μs
SCL LOW time	t_{LOW}	4,7	–	–	μs
SCL HIGH time	t_{HIGH}	4,0	–	–	μs
SCL and SDA rise time	t_R	–	–	1,0	μs
SCL and SDA fall time	t_F	–	–	0,3	μs
Data set-up time	$t_{SU;DAT}$	250	–	–	ns
Data hold time	$t_{HD;DAT}$	0	–	–	ns
SCL LOW to data out valid	$t_{VD;DAT}$	–	–	3,4	μs
Stop condition set-up time	$t_{SU;STO}$	4,0	–	–	μs

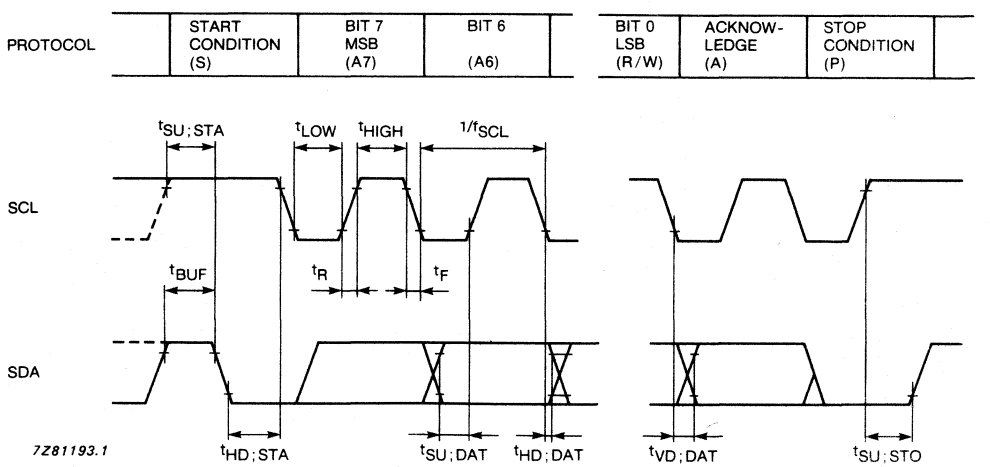


Fig. 15 I^2C bus timing diagram.

RATINGS

Limiting values in accordance with the Absolute Maximum System (IEC 134)

Supply voltage range	V_{DD}		−0,5 to +8,0 V
Voltage on any pin	V_I		−0,5 to V_{DD} +0,5 V
Input current d.c.	I_I	max.	10 mA
Output current d.c.	I_O	max.	20 mA
V_{DD} or V_{SS} current	I_{DD}, I_{SS}	max.	50 mA
Power dissipation per package	P_{tot}	max.	300 mW
Power dissipation per output	P	max.	100 mW
Storage temperature range	T_{stg}		−65 to +150 °C
Operating ambient temperature range	T_{amb}		−40 to +85 °C

Note:

Inputs and outputs are protected against electrostatic discharges in normal handling. However, to be totally safe, it is advised to take handling precautions appropriate to handling MOS devices (see 'Handling MOS devices').

CHARACTERISTICS

V_{DD} = 2,5 V to 6 V; V_{SS} = 0 V; T_{amb} = −40 °C to +85 °C unless otherwise specified

parameter	conditions	symbol	min.	typ.	max.	unit
Supply						
Supply voltage	operating	V_{DD}	2,5	–	6,0	V
Supply current	standby V_I = V_{SS} or V_{DD}; no load	I_{DD0}	–	1	15	μA
Supply current	operating; AOUT off; f_{SCL} = 100 kHz	I_{DD1}	–	125	250	μA
Supply current	AOUT active; f_{SCL} = 100 kHz	I_{DD2}	–	0,45	1,0	mA
Power-on reset level	note 1	V_{POR}	0,8	–	2,0	V
Digital inputs/output	SCL, SDA, A0, A1, A2					
Input voltage	LOW	V_{IL}	0	–	0,3 x V_{DD}	V
Input voltage	HIGH	V_{IH}	0,7 x V_{DD}	–	V_{DD}	V
Input current	leakage; V_I = V_{SS} to V_{DD}	I_I	–	–	250	nA
Input capacitance		C_I	–	–	5	pF
SDA output current	leakage; HIGH at V_{OH} = V_{DD}	I_{OH}	–	–	250	nA
SDA output current	LOW at V_{OL} = 0,4 V	I_{OL}	3,0	–	–	mA

parameter	conditions	symbol	min.	typ.	max.	unit
Reference voltage inputs	V_{REF}, AGND					
Voltage range	reference	V_{REF}	V_{AGND}	–	V_{DD}	V
Voltage range	analogue ground	V_{AGND}	V_{SS}	–	V_{REF}	V
Input current	leakage	I_I	–	–	250	nA
Input resistance	V_{REF} to AGND	R_{REF}	–	100	–	kΩ
Oscillator	OSC, EXT					
Input current	leakage	I_I	–	–	250	nA
Oscillator frequency		f_{OSC}	0,75	–	1,25	MHz

D/A CHARACTERISTICS

V_{DD} = 5,0 V; V_{SS} = 0 V; V_{REF} = 5,0 V; V_{AGND} = 0 V; R_{load} = 10 kΩ; C_{load} = 100 pF; T_{amb} = −40 °C to +85 °C unless otherwise specified

parameter	conditions	symbol	min.	typ.	max.	unit
Analogue output						
Output voltage range	no resistive load	V_{OA}	V_{SS}	–	V_{DD}	V
Output voltage range	R_{load} = 10 kΩ	V_{OA}	V_{SS}	–	0,9xV_{DD}	V
Output current	leakage; AOUT disabled	I_{LO}	–	–	250	nA
Accuracy						
Offset error	T_{amb} = 25 °C	OS_e	–	–	50	mV
Linearity error		L_e	–	–	±1,5	LSB
Gain error	no resistive load	G_e	–	–	1	%
Settling time	to ½ LSB full scale step	t_{DAC}	–	–	90	µs
Conversion rate		f_{DAC}	–	–	11,1	kHz
Supply noise rejection	at f = 100 Hz; V_{DD} = 0,1 V_{PP}	SNRR	–	40	–	dB

DEVELOPMENT DATA

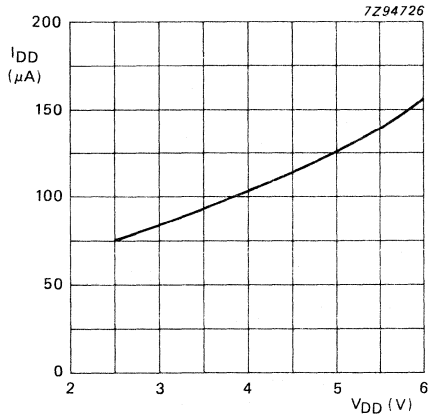
A/D CHARACTERISTICS

V_{DD} = 5,0 V; V_{SS} = 0 V; V_{REF} = 5,0 V; V_{AGND} = 0 V; R_{source} = 10 kΩ; T_{amb} = −40 °C to +85 °C unless otherwise specified

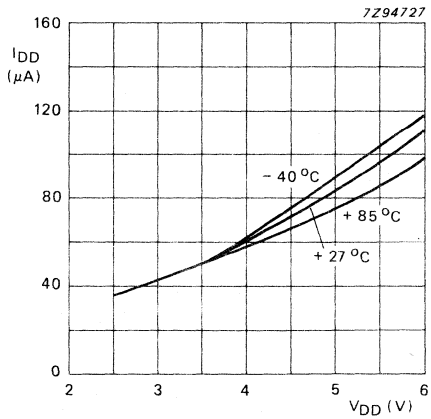
parameter	conditions	symbol	min.	typ.	max.	unit
Analogue inputs						
Input voltage range		V_{IA}	V_{SS}	–	V_{DD}	V
Input current	leakage	I_{IA}	–	–	100	nA
Input capacitance		C_{IA}	–	10	–	pF
Input capacitance	differential	C_{ID}	–	10	–	pF
Single-ended voltage	measuring range	V_{IS}	V_{AGND}	–	V_{REF}	V
Differential voltage	measuring range; $V_{FS} = V_{REF} - V_{AGND}$	V_{ID}	$\frac{-V_{FS}}{2}$	–	$\frac{+V_{FS}}{2}$	V
Accuracy						
Offset error	T_{amb} = 25 °C	OS_e	–	–	20	mV
Linearity error		L_e	–	–	±1,5	LSB
Gain error		G_e	–	–	1	%
Gain error	small-signal; ΔV_{IN} = 16 LSB	GS_e	–	–	5	%
Rejection ratio	common-mode	CMRR	–	60	–	dB
Supply noise rejection	at f = 100 Hz; V_{DDN} = 0,1xV_{PP}	SNRR	–	40	–	dB
Conversion time		t_{ADC}	–	–	90	μs
Sampling/conversion rate		f_{ADC}	–	–	11,1	kHz

Note

1. The power on reset circuit resets the I^2C bus logic when V_{DD} is less than V_{POR}.



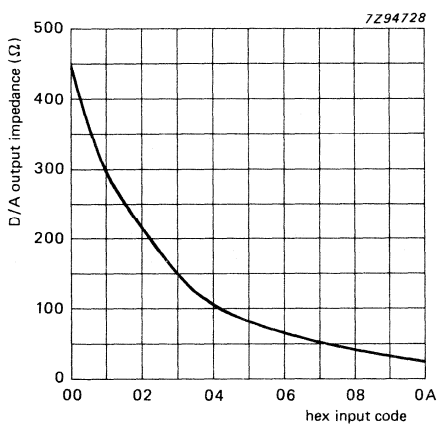
(a) internal oscillator; T_{amb} = + 27 °C.



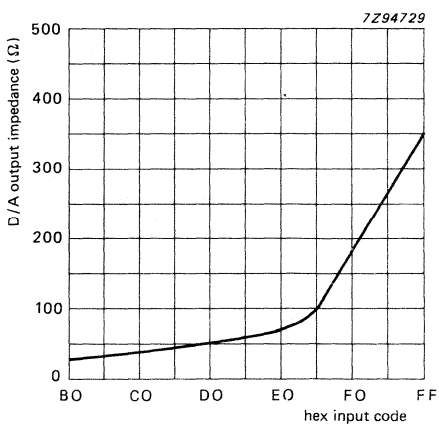
(b) external oscillator.

Fig. 16 Operating supply current against supply voltage (analogue output disabled).

DEVELOPMENT DATA



(a) output impedance near negative power rail; T_{amb} = + 27 °C.



(b) output impedance near positive power rail; T_{amb} = + 27 °C.

Fig. 17 Output impedance of analogue output buffer (near power rails).

The x-axis represents the hex input-code equivalent of the output voltage.

APPLICATION INFORMATION

Inputs must be connected to V_{SS} or V_{DD} when not in use. Analogue inputs may also be connected to AGND or V_{REF}.

In order to prevent excessive ground and supply noise and to minimize cross-talk of the digital to analogue signal paths the user has to design the printed-circuit board layout very carefully. Supply lines common to a PCF8591 device and noisy digital circuits and ground loops should be avoided. Decoupling capacitors (> 10 μF) are recommended for power supply and reference voltage inputs.

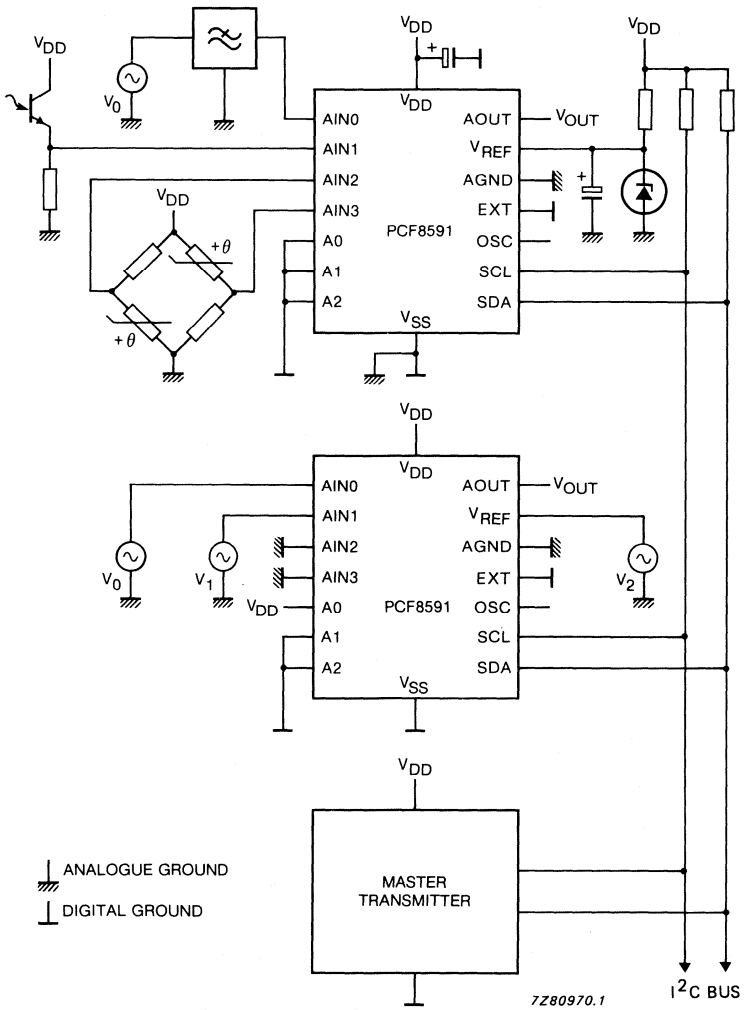


Fig. 18 Application diagram.

Purchase of Philips' I²C components conveys a license under the Philips' I²C patent to use the components in the I²C-system provided the system conforms to the I²C specification defined by Philips.

PACKAGE INFORMATION

PACKAGE OUTLINES

For products: 2960, 2964B, 9401, 9403 and for products with prefixes: SCN, 8T, 8X

GENERAL INFORMATION

INTRODUCTION

The following information applies to all packages unless otherwise specified on individual package outline drawings.

General

1. Dimensions shown are metric units (millimeters), except those in parentheses which are English units (inches).
2. Lead spacing shall be measured within this zone.
   a. Shoulder and lead tip dimensions are to centerline of leads.
3. Tolerances non-cumulative
4. Thermal resistance values are determined by utilizing the linear temperature dependence of the forward voltage drop across the substrate diode in a digital device to monitor the junction temperature rise during known power application across V_{CC} and ground. The values are based upon 120 mils square die for plastic packages and a 90 mils square die in the smallest available cavity for hermetic packages. All units were solder mounted to P.C. boards, with standard stand-off, for measurement.

Plastic Only

5. Lead material: Olin 194 (copper alloy) or equivalent, solder dipped.
6. Body material: Plastic (epoxy)
7. Round hole in top corner denotes lead No. 1.
8. Body dimensions do not include molding flash.
9. SO packages–microminiature packages:
   a. Lead material: Olin 194 (copper alloy) or equivalent, solder dipped.
   b. Body material: Plastic (epoxy).

Hermetic Only

10. Lead material
    a. ASTM alloy F-15 (KOVAR) or equivalent—gold plated, tin plated, or solder dipped.
    b. ASTM alloy F-30 (Alloy 42) or equivalent—tin plated, gold plated or solder dipped.
    c. ASTM alloy F-15 (KOVAR) or equivalent—gold plated.
11. Body Material
    a. Eyelet, ASTM alloy F-15 or equivalent—gold or tin plated, glass body.
    b. Ceramic with glass seal at leads.
    c. BeO ceramic with glass seal at leads.
    d. Ceramic with ASTM alloy F-30 or equivalent.
12. Lid Material
    a. Nickel or tin plated nickel, weld seal
    b. Ceramic, glass seal.
    c. ASTM alloy F-15 or equivalent, gold plated, alloy seal.
    d. BeO Ceramic with glass seal
13. Signetics symbol, angle cut, or lead tab denotes Lead No. 1.
14. Recommended minimum offset before lead bend.
15. Maximum glass climb .010 inches.
16. Maximum glass climb or lid skew is .010 inches.
17. Typical four places.
18. Dimension also applies to seating plane.

PLASTIC PACKAGES

NO. OF LEADS	PACKAGE CODE	Θ_{ja}/Θ_{jc} (°C/W)	DESCRIPTION
Plastic Dual-In-Line			
14	NHA	95/33	Cu Lead Frame
16	NJA	73/33	Cu Lead Frame
18	NKA	69/26	Cu Lead Frame
20	NLA	65/26	Cu Lead Frame
24	NNA	60/23	Cu Lead Frame
28	NQA	56/21	Cu Lead Frame
SO Packages			
16	DJ	tbd	SO-16L
20	DL	tbd	SO-20L

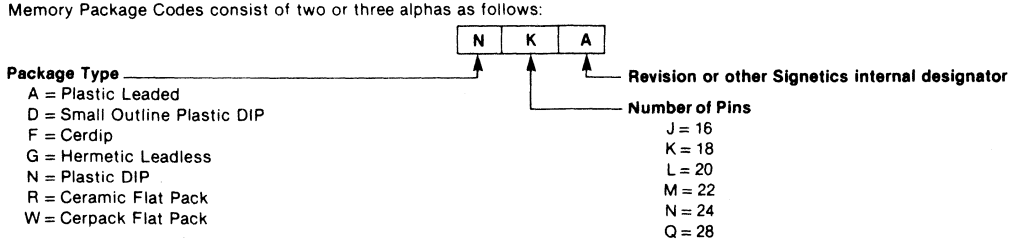
GENERAL INFORMATION (continued)

HERMETIC PACKAGES

NO. OF LEADS	PACKAGE CODE	Θ_{ja}/Θ_{jc} (°C/W)	DESCRIPTION[1]
Flat Packs			
10	QF	230/55	Flat Ceramic
10	WF	240/50	Flat Ceramic
14	QHA	185/45	Flat Ceramic Laminate
14	WH	205/50	Flat Ceramic
16	QJA	170/45	Flat Ceramic Laminate
16	RJA	133/30	Flat Ceramic, BeO
16	WJ	200/50	Flat Ceramic
18	RKA	107/22	Flat Ceramic, BeO
24	QNA	155/44	Flat Ceramic Laminate
24	RNA	107/22	Flat Ceamic, BeO
24	WN	155/40	Flat Ceramic
28	RQA	107/22	Flat Ceramic, BeO
40	RWA	95/20	Flat Ceramic, BeO
Cerdip Family			
8	FE	110/30	Dual-In-Line Ceramic
14	FH	110/30	Dual-In-Line Ceramic
16	FJ	77/30	Dual-In-Line Ceramic
18	FK	73/27	Dual-In-Line Ceramic
20	FL	72/25	Dual-In-Line Ceramic
22	FM	66/27	Dual-In-Line Ceramic
24	FN	63/26	Dual-In-Line Ceramic
28	FQ	57/27	Dual-In-Line Ceramic

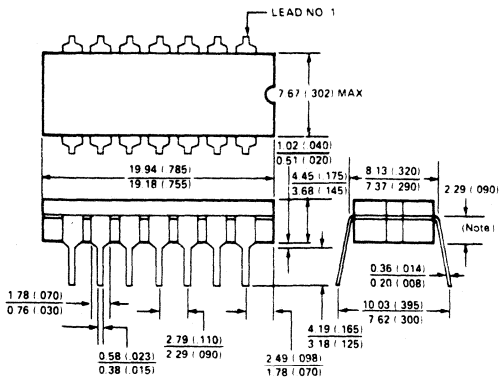
NOTE
1. Dual-In-Line packages unless otherwise described

Memory Package Codes consist of two or three alphas as follows:

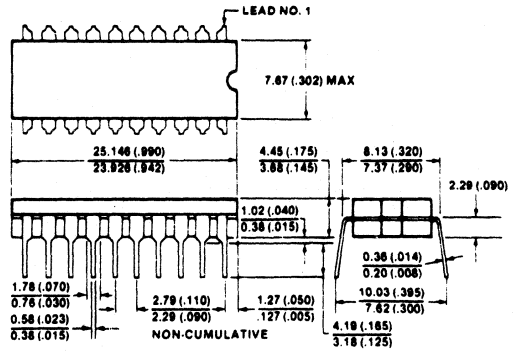


HERMETIC: Cerdip

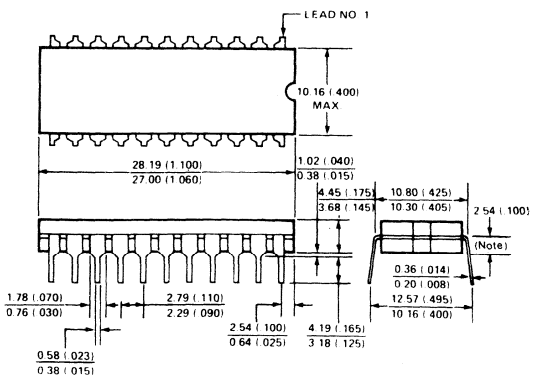
FH 14



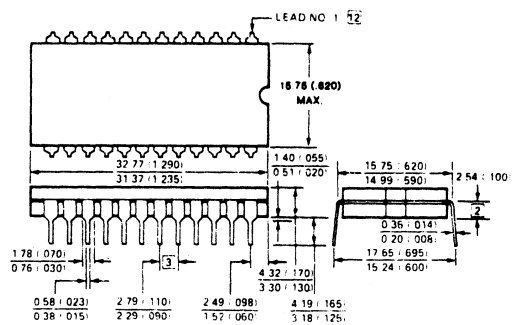
FL 20



FM 22

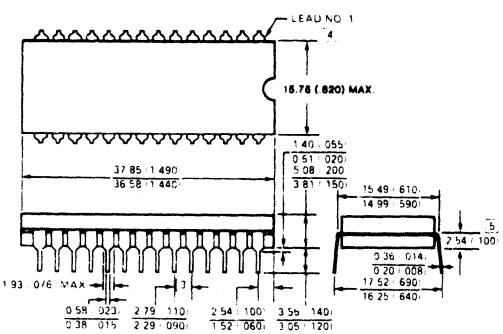


FN 24



CONSTRUCTION NOTES 9c 10d 11c

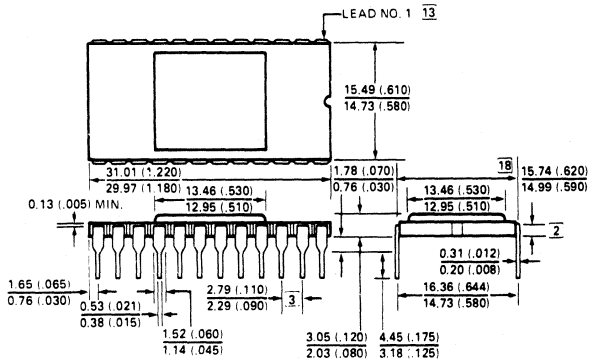
FQ 28



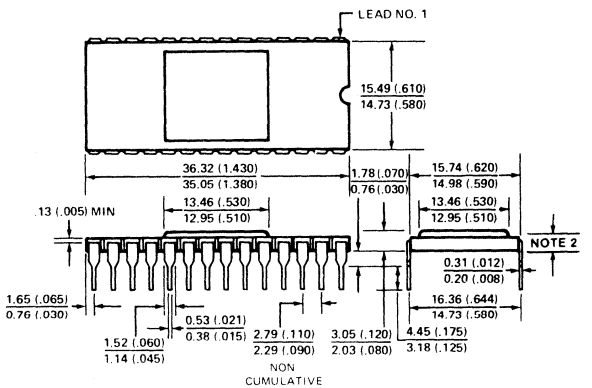
CONSTRUCTIONS NOTES 9c 10d 11c

HERMETIC: Cerdil

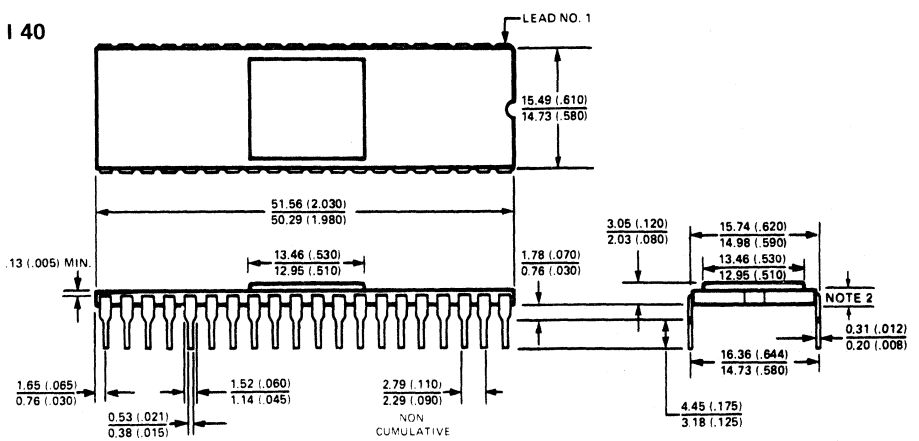
I 24



I 28

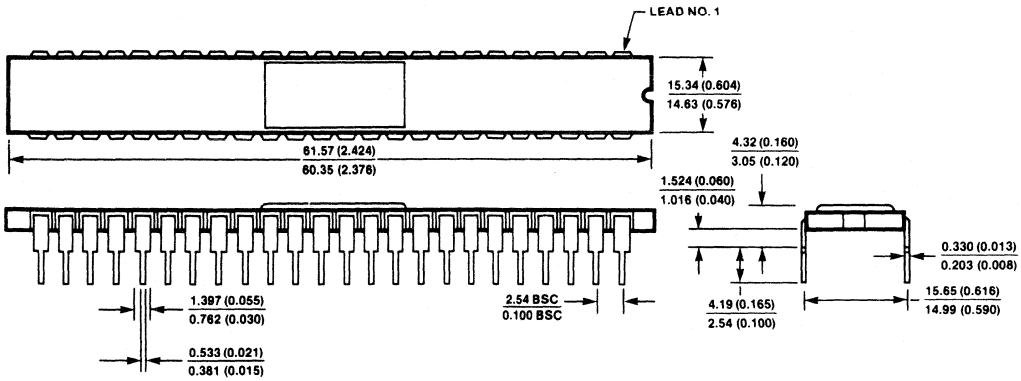


I 40

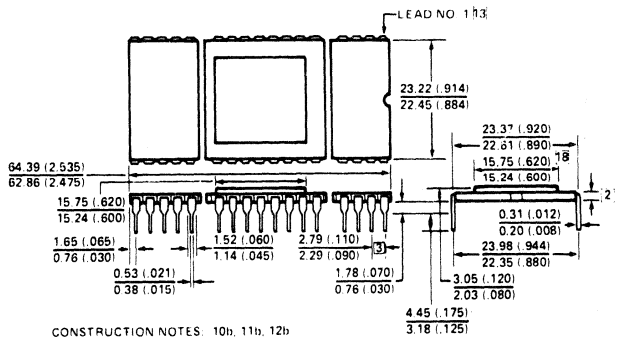


HERMETIC: Cerdil (continued)

I 48



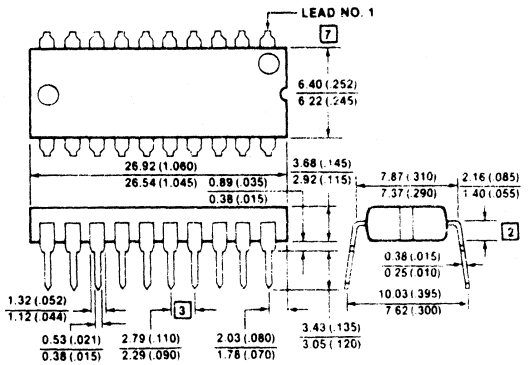
I 50



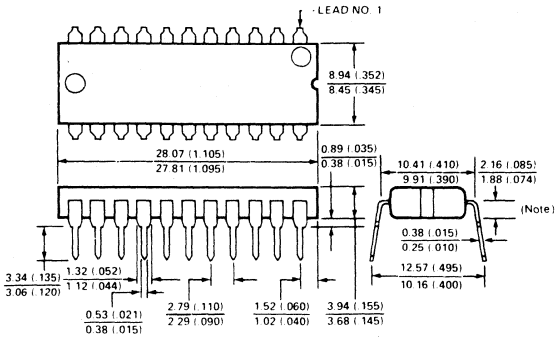
NOTE:
1. If solder dipped terminals used, terminal dimension tolerances may be increased by .001.

PLASTIC: Dual-in-line

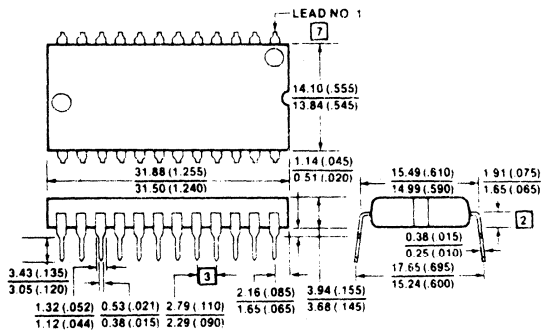
NLA 20



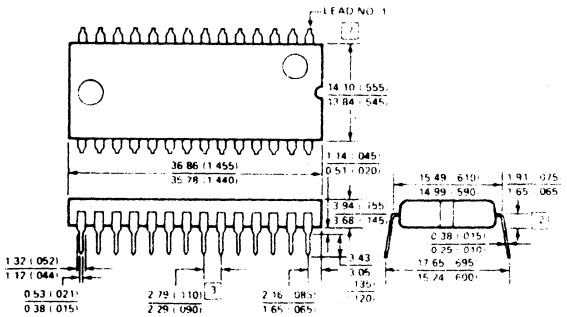
N 22



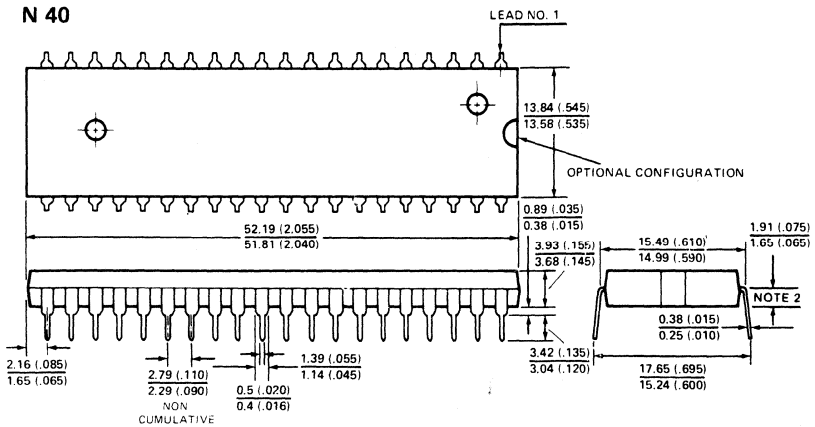
NNA 24



NQA 28



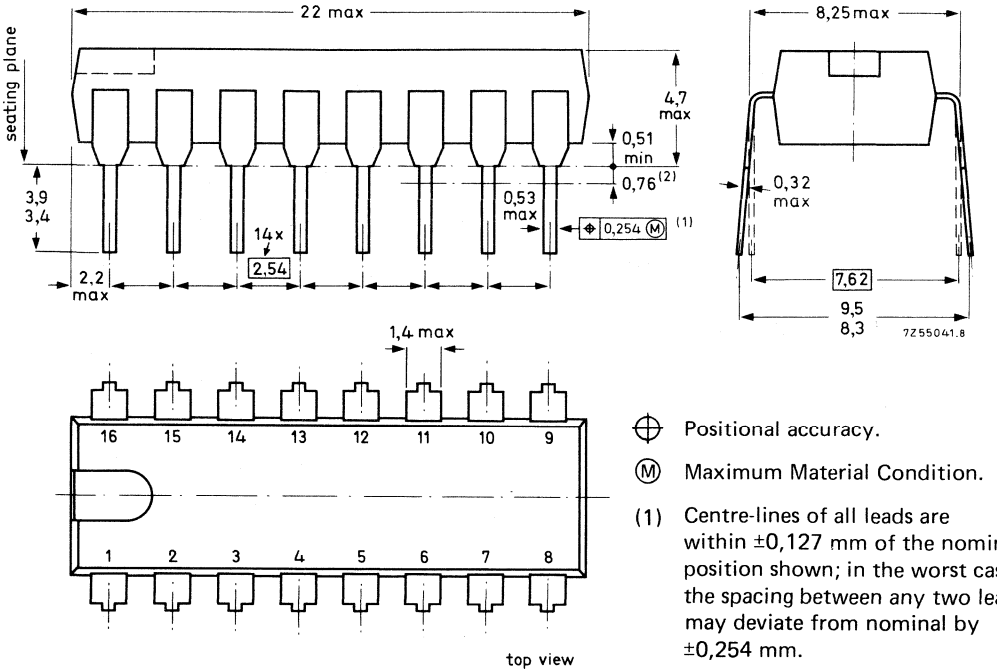
N 40



N 14, N 48, N 50, N 64 For current information contact local sales offices.

For products with prefixes: MAB, MAF, MEA, OM, PCD, PCF, SAA, TEA

16-LEAD DUAL IN-LINE; PLASTIC (SOT-38)



⊕ Positional accuracy.

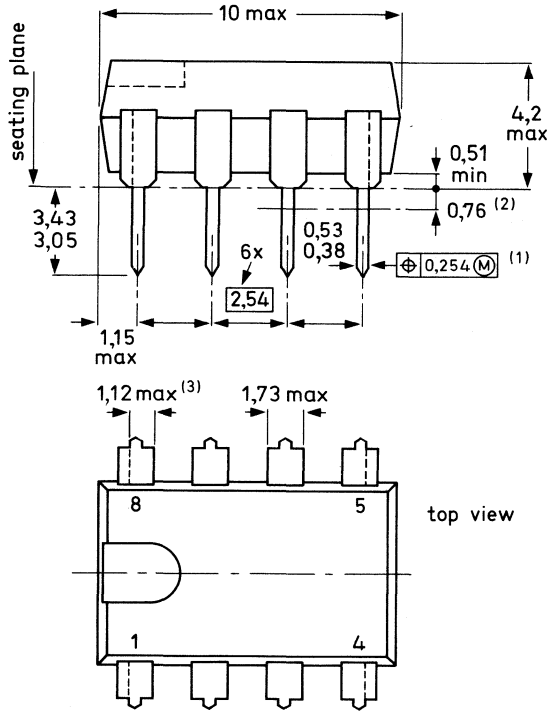
Ⓜ Maximum Material Condition.

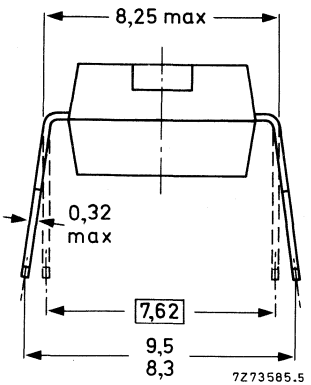
(1) Centre-lines of all leads are within ±0,127 mm of the nominal position shown; in the worst case, the spacing between any two leads may deviate from nominal by ±0,254 mm.

(2) Lead spacing tolerances apply from seating plane to the line indicated.

Dimensions in mm

8-LEAD DUAL IN-LINE; PLASTIC (SOT-97AE, DE, EE)





Dimensions in mm

⊕ Positional accuracy.

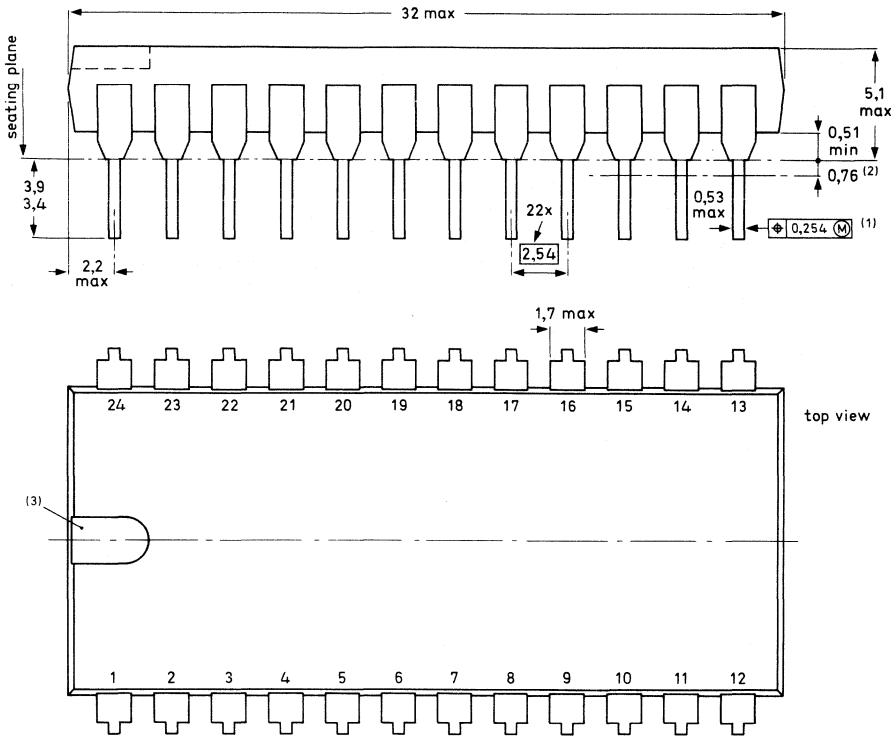
Ⓜ Maximum Material Condition.

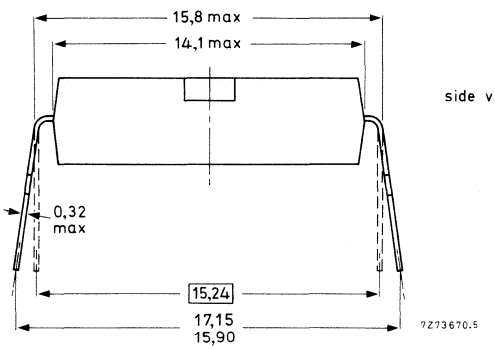
(1) Centre-lines of all leads are within ±0,127 mm of the nominal position shown; in the worst case, the spacing between any two leads may deviate from nominal by ±0,254 mm.

(2) Lead spacing tolerances apply from seating plane to the line indicated.

(3) Only for devices with asymmetrical end-leads.

24-LEAD DUAL IN-LINE; PLASTIC (SOT-101A,B,F,G,L)





⊕ Positional accuracy.

Ⓜ Maximum Material Condition.

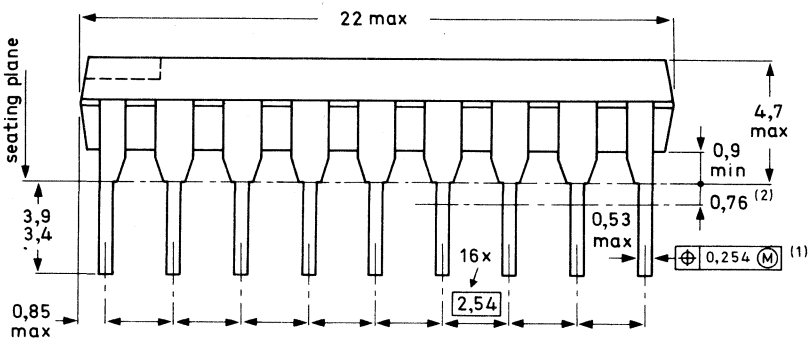
(1) Centre-lines of all leads are within ±0,127 mm of the nominal position shown; in the worst case, the spacing between any two leads may deviate from nominal by ±0,254 mm.

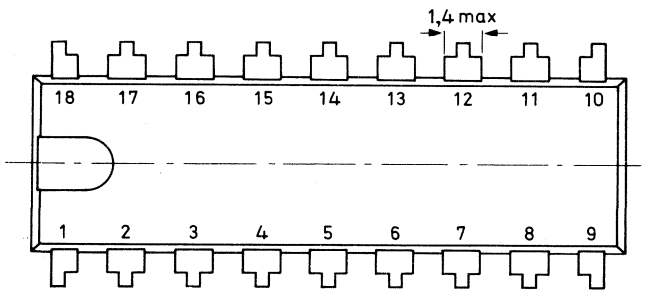
(2) Lead spacing tolerances apply from seating plane to the line indicated.

(3) Index may be horizontal as shown, or vertical.

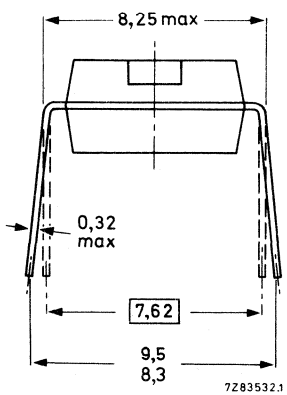
Dimensions in mm

18-LEAD DUAL IN-LINE; PLASTIC (SOT-102HE, HG, KE, ME, PG)





top view



side view

⊕ Positional accuracy.

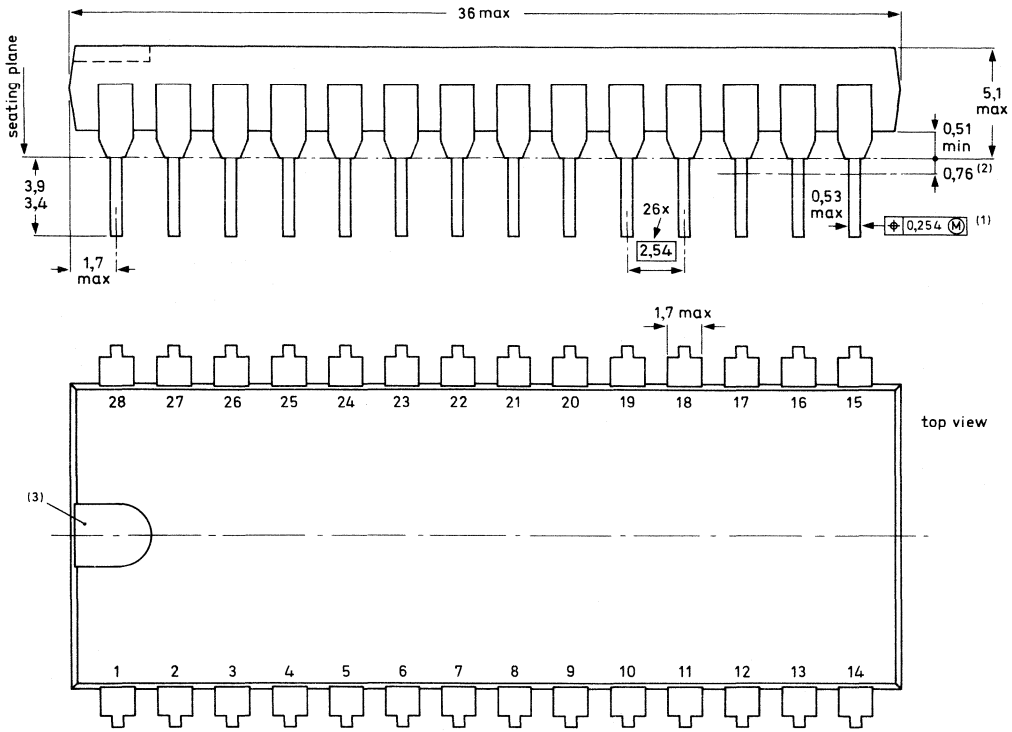
Ⓜ Maximum Material Condition.

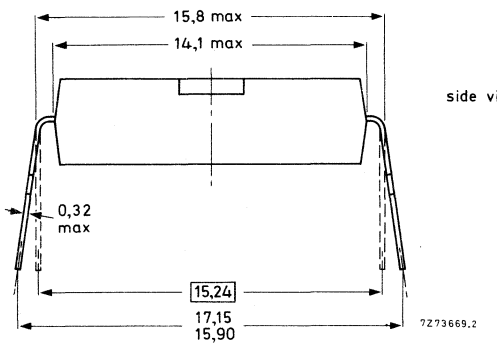
(1) Centre-lines of all leads are within ±0,127 mm of the nominal position shown; in the worst case, the spacing between any two leads may deviate from nominal by ±0,254 mm.

(2) Lead spacing tolerances apply from seating plane to the line indicated.

Dimensions in mm

28-LEAD DUAL IN-LINE; PLASTIC (SOT-117)





⊕ Positional accuracy.

Ⓜ Maximum Material Condition.

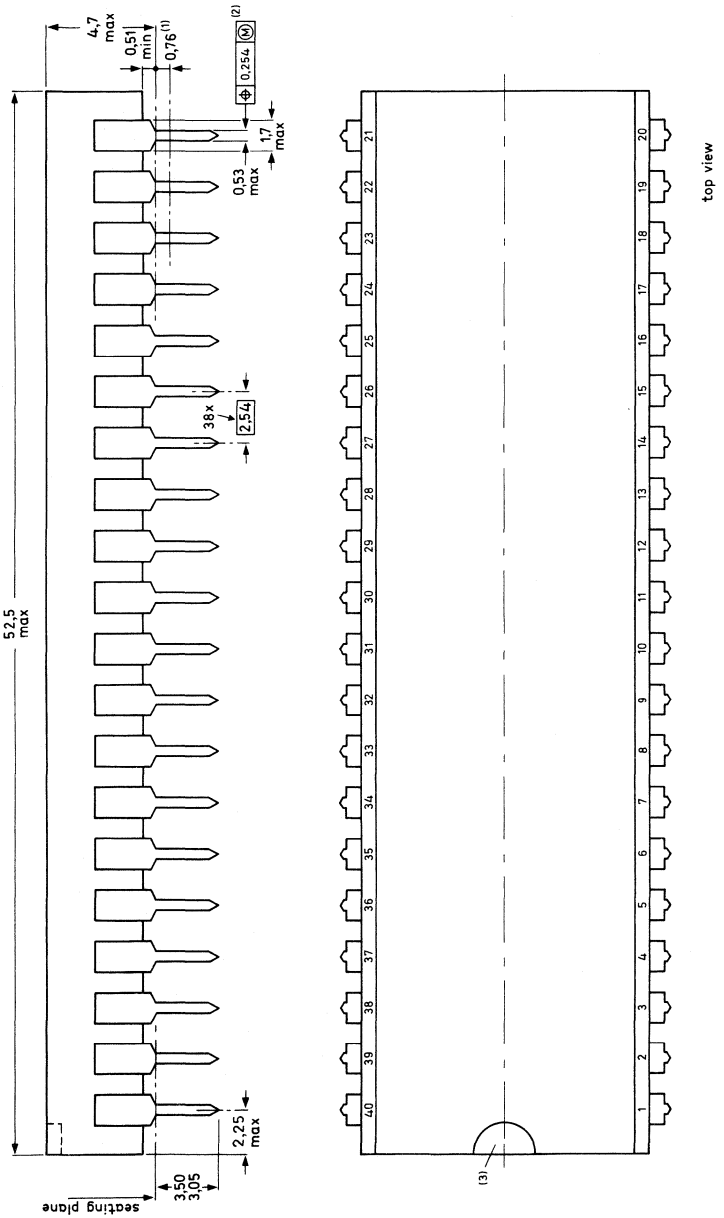
(1) Centre-lines of all leads are within ±0,127 mm of the nominal position shown; in the worst case, the spacing between any two leads may deviate from nominal by ±0,254 mm.

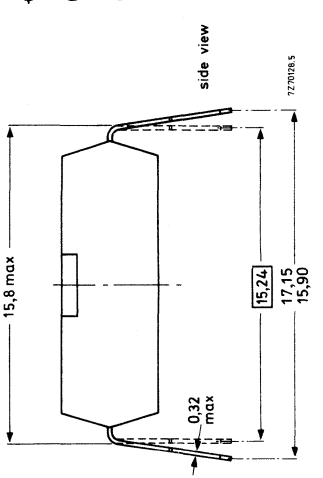
(2) Lead spacing tolerances apply from seating plane to the line indicated.

(3) Index may be horizontal as shown, or vertical.

Dimensions in mm

40-LEAD DUAL IN-LINE; PLASTIC (SOT-129)





⊕ Positional accuracy.

Ⓜ Maximum Material Condition.

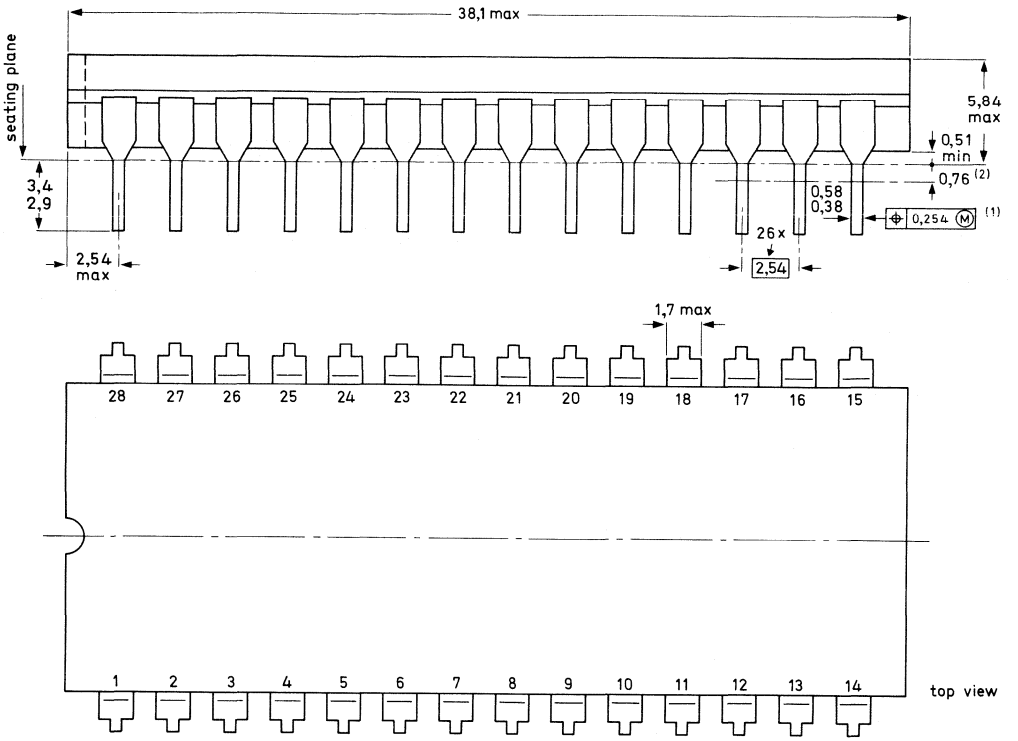
(1) Centre-lines of all leads are within ±0,127 mm of the nominal position shown; in the worst case, the spacing between any two leads may deviate from nominal by ±0,254 mm.

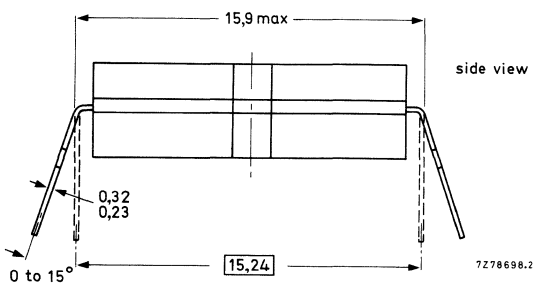
(2) Lead spacing tolerances apply from seating plane to the line indicated.

(3) Index may be horizontal as shown, or vertical.

Dimensions in mm

28-LEAD DUAL IN-LINE; CERAMIC (CERDIP) (SOT-135A)





⊕ Positional accuracy.

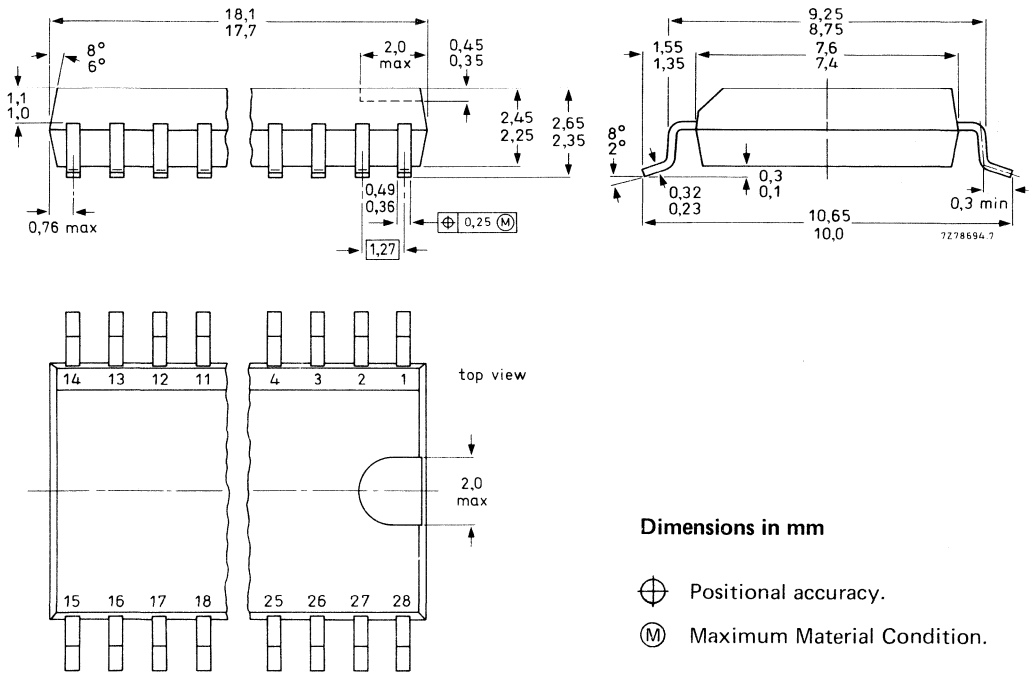
Ⓜ Maximum Material Condition.

(1) Centre-lines of all leads are within ±0,127 mm of the nominal position shown; in the worst case, the spacing between any two leads may deviate from nominal by ±0,254 mm.

(2) Lead spacing tolerances apply from seating plane to the line indicated.

Dimensions in mm

28-LEAD MINI-PACK; PLASTIC (SO-28; SOT-136A)

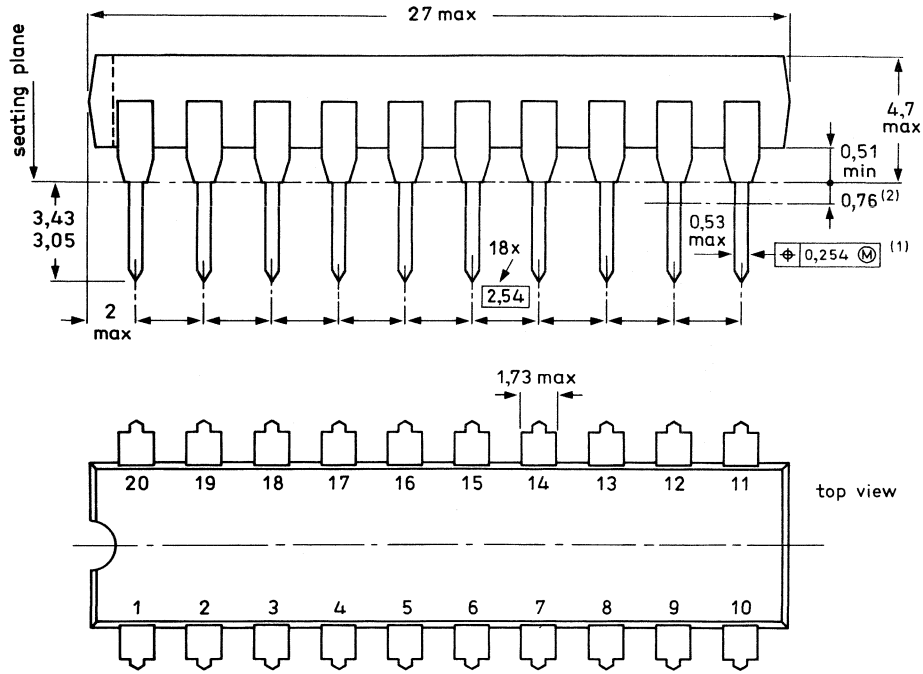


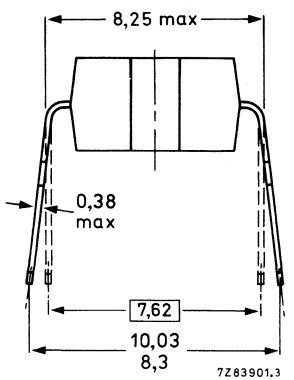
Dimensions in mm

⊕ Positional accuracy.

Ⓜ Maximum Material Condition.

20-LEAD DUAL IN-LINE; PLASTIC (SOT-146; 146C1)





side view

⊕ Positional accuracy.

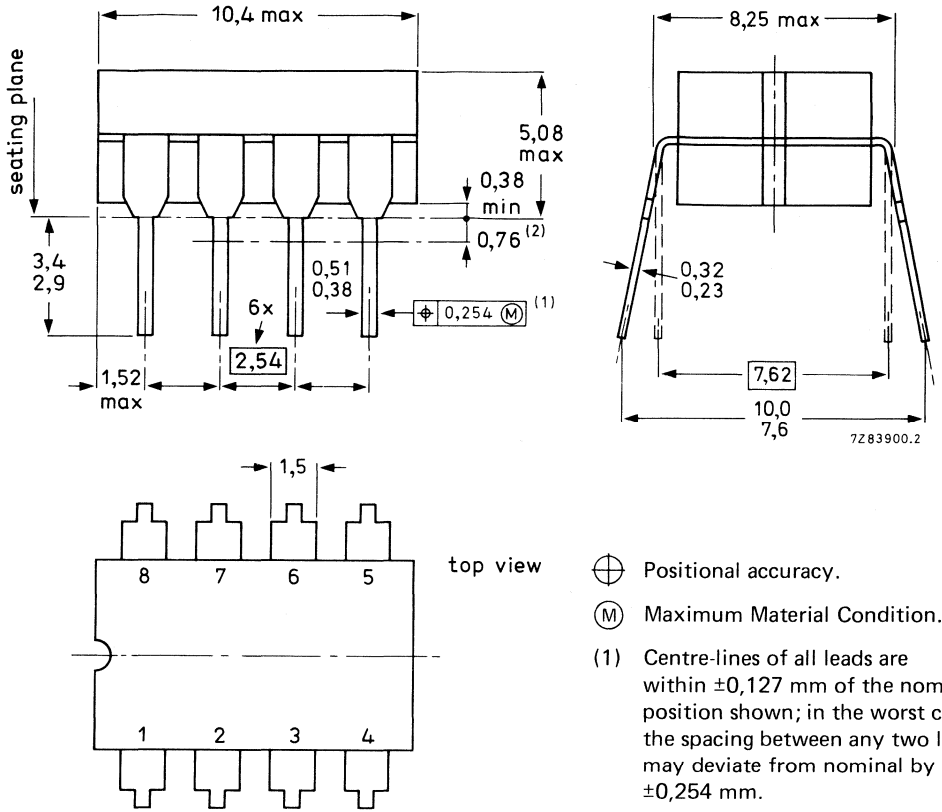
Ⓜ Maximum Material Condition.

(1) Centre-lines of all leads are within ±0,127 mm of the nominal position shown; in the worst case, the spacing between any two leads may deviate from nominal by ±0,254 mm.

(2) Lead spacing tolerances apply from seating plane to the line indicated.

Dimensions in mm

8-LEAD DUAL IN-LINE; CERAMIC (CERDIP) (SOT-151A)



⊕ Positional accuracy.

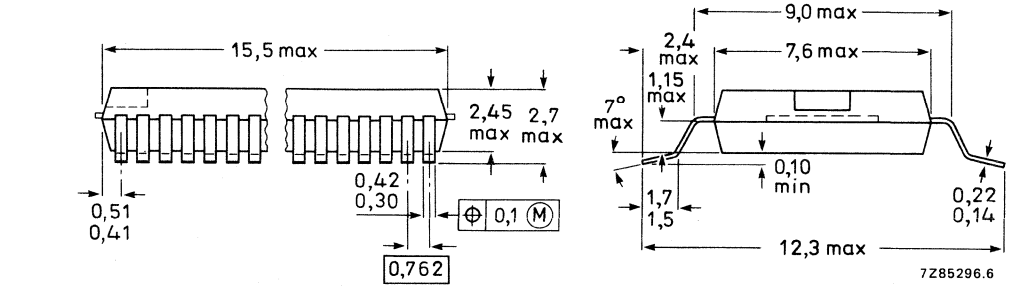
Ⓜ Maximum Material Condition.

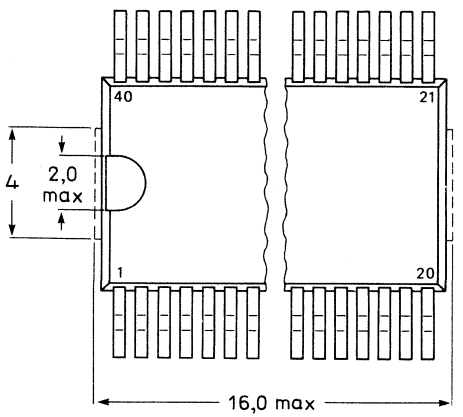
(1) Centre-lines of all leads are within ±0,127 mm of the nominal position shown; in the worst case, the spacing between any two leads may deviate from nominal by ±0,254 mm.

(2) Lead spacing tolerances apply from seating plane to the line indicated.

Dimensions in mm

40-LEAD MINI-PACK; PLASTIC (VSO-40; SOT-158A)





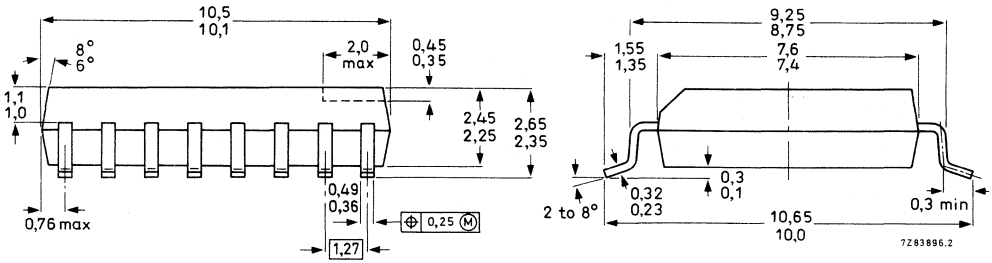
top view

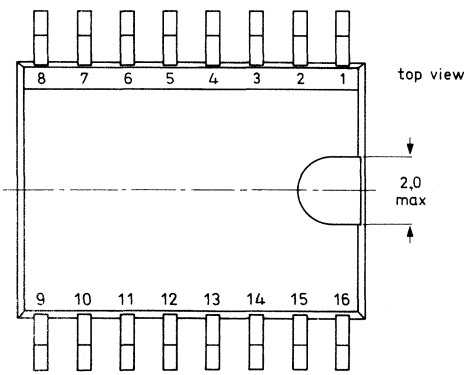
Dimensions in mm

⊕ Positional accuracy.

Ⓜ Maximum Material Condition.

16-LEAD MINI-PACK; PLASTIC (SO-16L; SOT-162A)



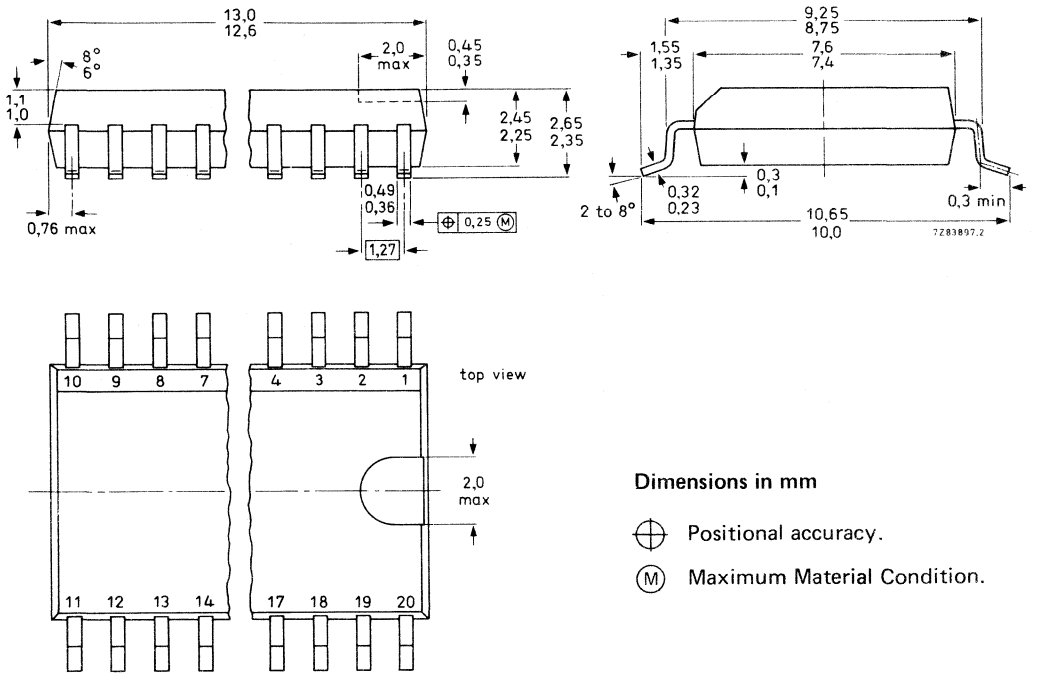


Dimensions in mm

⊕ Positional accuracy.

Ⓜ Maximum Material Condition.

20-LEAD MINI-PACK; PLASTIC (SO-20; SOT-163A)

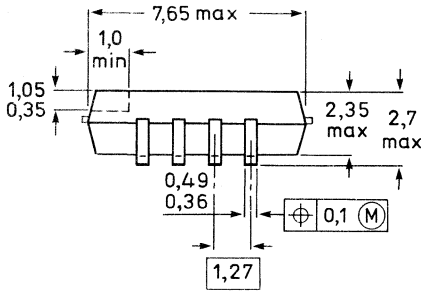


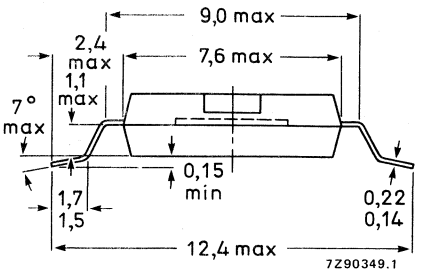
Dimensions in mm

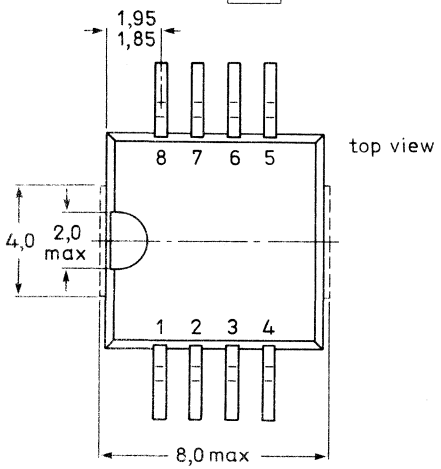
⊕ Positional accuracy.

Ⓜ Maximum Material Condition.

8-LEAD MINI-PACK; PLASTIC (SO-8L; SOT-176)





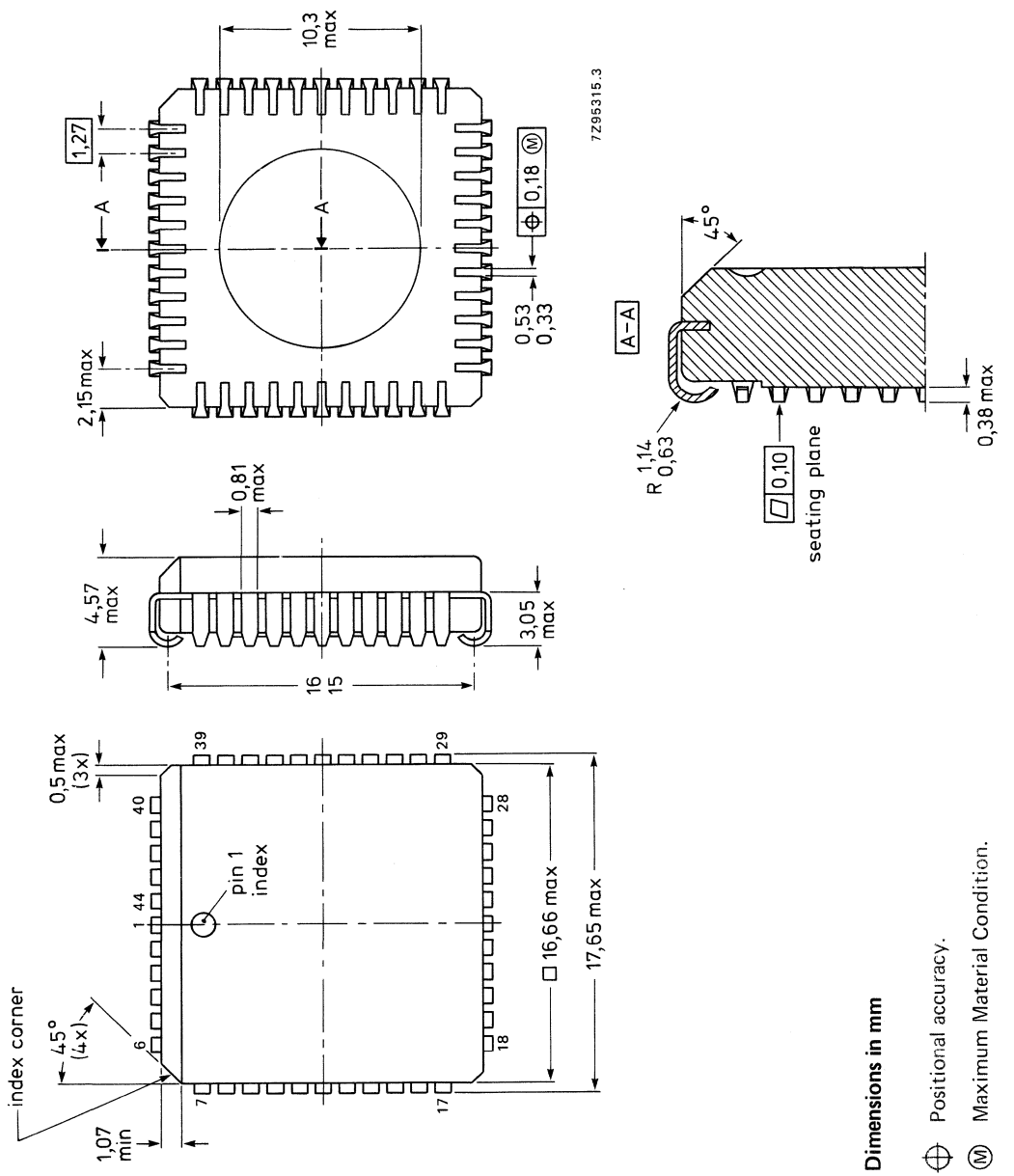


Dimensions in mm

⊕ Positional accuracy.

Ⓜ Maximum Material Condition.

44-LEAD PLASTIC LEADED CHIP-CARRIER (PLCC); SOT-187A

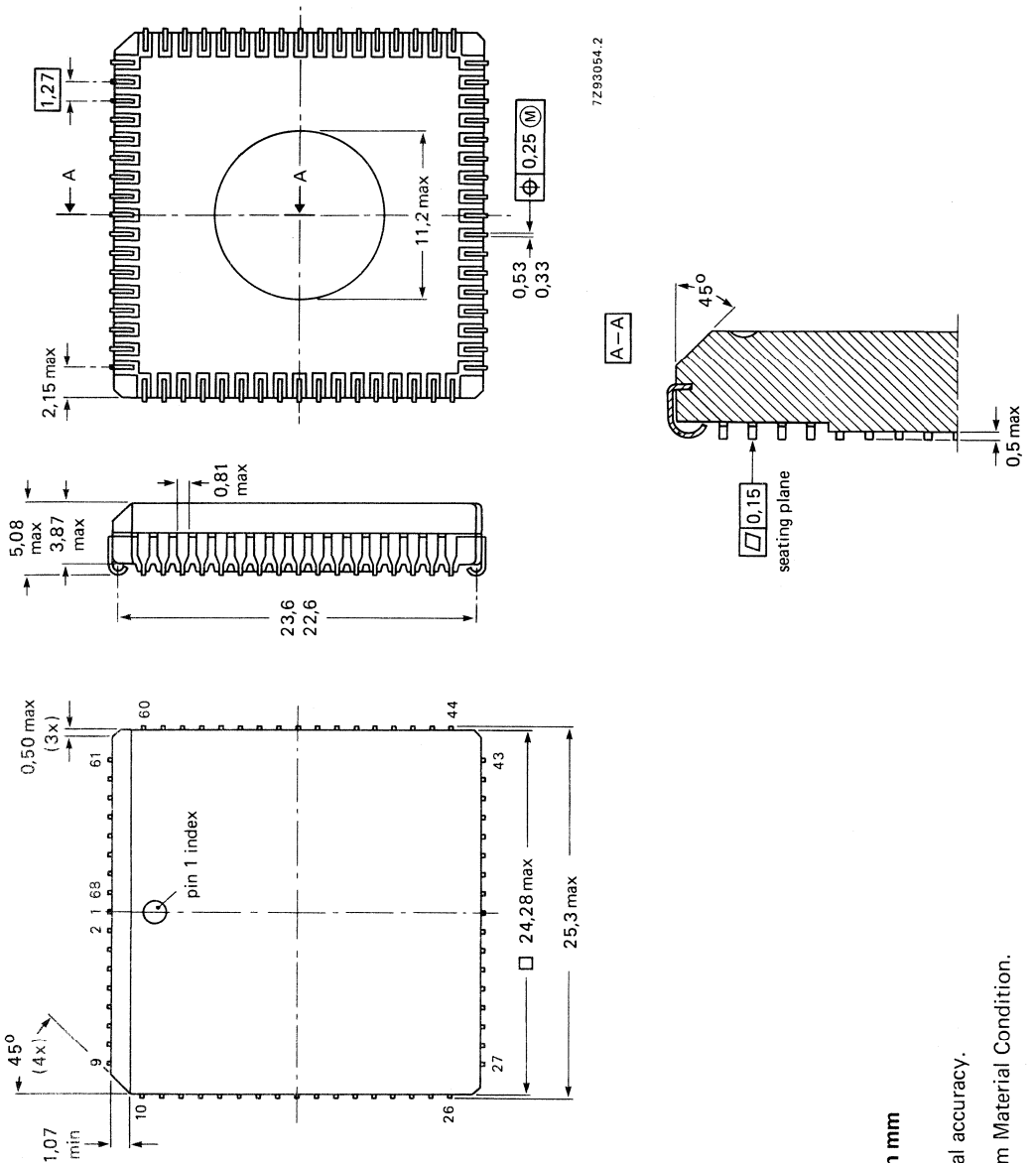


Dimensions in mm

⊕ Positional accuracy.

Ⓜ Maximum Material Condition.

68-LEAD PLASTIC LEADED CHIP-CARRIER (PLCC); SOT-188A

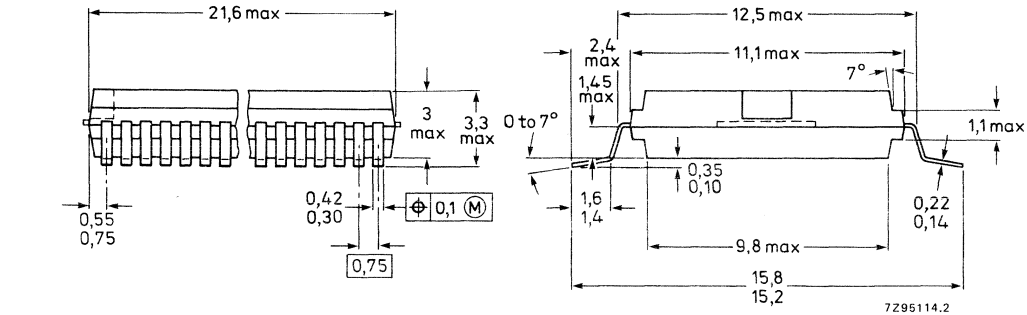


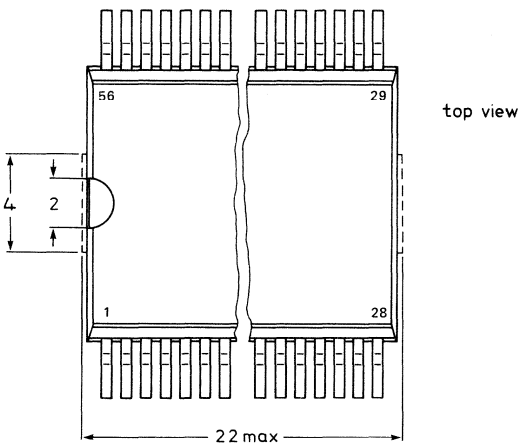
Dimensions in mm

⊕ Positional accuracy.

Ⓜ Maximum Material Condition.

56-LEAD MINI-PACK; PLASTIC (VSO-56; SOT-190)





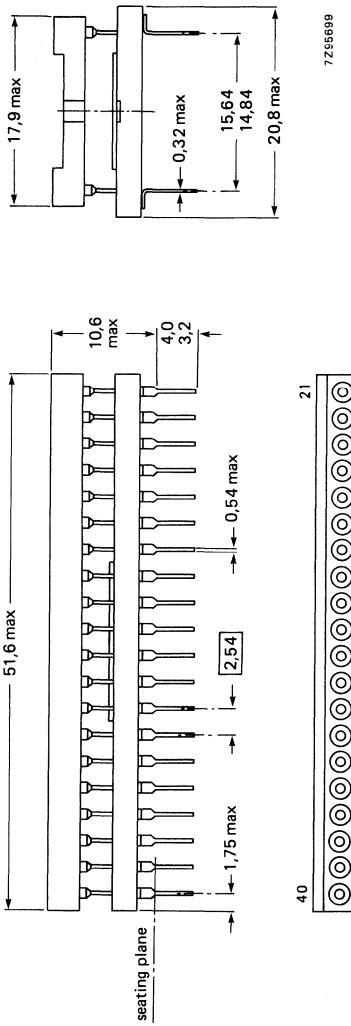
top view

Dimensions in mm

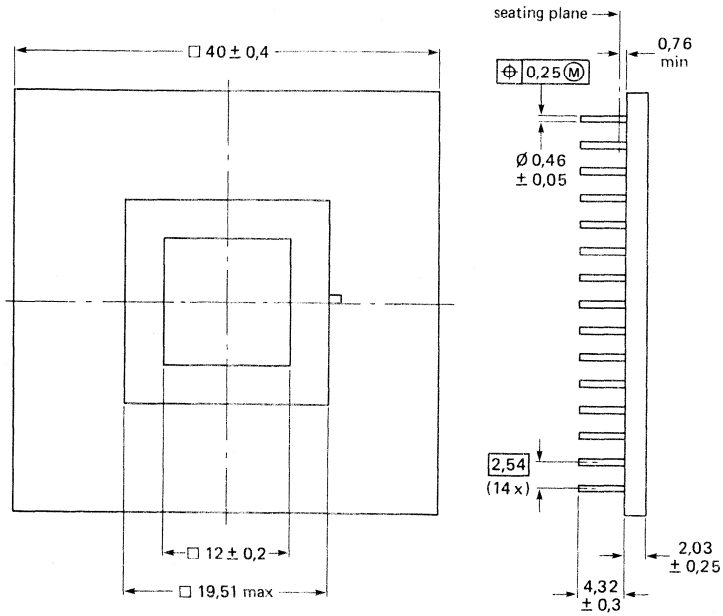
⊕ Positional accuracy.

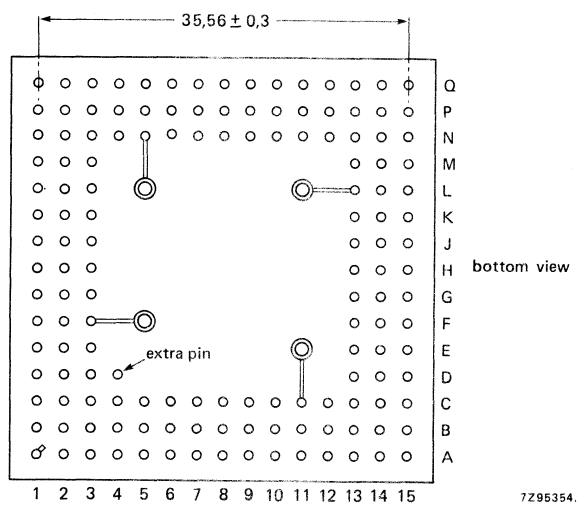
Ⓜ Maximum Material Condition.

40-LEAD DUAL IN-LINE PIGGY BACK; PLASTIC (SOT-215)



144-PIN GRID ARRAY (PGA)





Soldering

Plastic dual in-line (DIL) packages

1. By hand

Apply the soldering iron below the seating plane (or not more than 2 mm above it).
If its temperature is below 300 °C it must not be in contact for more than 10 seconds; if between 300 °C and 400 °C, for not more than 5 seconds.

2. By dip or wave

The maximum permissible temperature of the solder is 260 °C; this temperature must not be in contact with the joint for more than 5 seconds. The total contact time of successive solder waves must not exceed 5 seconds.
The device may be mounted up to the seating plane, but the temperature of the plastic body must not exceed the specified storage maximum. If the printed-circuit board has been pre-heated, forced cooling may be necessary immediately after soldering to keep the temperature within the permissible limit.

3. Repairing soldered joints

The same precautions and limits apply as in (1) above.

Plastic mini-pack (SO) packages

1. By hand-held soldering iron or pulse-heated solder tool

Apply the heating tool to the flat part of the lead only. Contact time must be limited to 10 seconds at up to 300 °C. When using proper tools, all leads can be soldered in one operation within 2 to 5 seconds at between 270 and 320 °C. (Pulse-heated soldering is not recommended for SO packages).

For pulse-heated solder tool (resistance) soldering of VSO packages, solder is applied to substrate by dipping or by an extra thick tin/lead plating before package placement.

2. By wave

Maximum permissible solder temperature is 260 °C, and maximum duration of package immersion in solder bath is 10 seconds, if allowed to cool to less than 150 °C within 6 seconds. Typical dwell time is 4 seconds at 250 °C.

A modified wave soldering technique is recommended, using two solder waves (dual-wave); a first turbulent wave with high upward pressure is followed by a smooth, laminar wave. A mildly activated flux will eliminate the need for removal of corrosive residues in most applications.

3. By solder paste reflow

Reflow soldering requires the solder paste (a suspension of fine solder particles, flux and binding agent) to be applied to the substrate by screen printing or pressure-syringe dispensing before device placement.

Several techniques exist for reflowing, for example, thermal conduction by heated belt, infrared, and vapour-phase reflow. Dwell times vary between 8 and 60 seconds according to method. Typical reflow temperatures range from 215 to 250 °C.

Pre-heating is necessary to dry paste and evaporate binding agent, and to reduce thermal shock on entry to reflow zone.

4. Repairing soldered joints

The same precautions and limits apply as in (1) above.

Electronic components and materials for professional, industrial and consumer uses from the world-wide Philips Group of Companies

Argentina: PHILIPS ARGENTINA S.A., Div. Elcoma, Vedia 3892, 1430 BUENOS AIRES, Tel. (01) 541 - 7141 to 7747.
Australia: PHILIPS INDUSTRIES LTD., Elcoma Division, 11 Waltham Street, ARTARMON, N.S.W. 2064, Tel. (02)439 3322.
Austria: ÖSTERREICHISCHE PHILIPS INDUSTRIE G.m.b.H., UB Bauelemente, Triester Str. 64, A-1101 WIEN, Tel. (0222) 62 91 11-0.
Belgium: N.V. PHILIPS & MBLE ASSOCIATED, 80 Rue Des Deux Gares, B-1070 BRUXELLES, Tel. (02) 525-61-11.
Brazil: CONSTANTA-IBRAPE; (Active Devices): Av. Brigadeiro Faria Lima, 1735-SAO PAULO-SP, Tel. (011) 211-2600,
(Passive Devices & Materials): Av. Francisco Monteiro, 702 - RIBEIRAO PIRES-SP, Tel. (011) 459-8211.
Canada: PHILIPS ELECTRONICS LTD., Elcoma Division, 601 Milner Ave., SCARBOROUGH, Ontario, M1B 1M8, Tel. (416) 292-5161.
Chile: PHILIPS CHILENA S.A., Av. Santa Maria 0760, SANTIAGO, Tel. (02) 77 38 16.
Colombia: IND. PHILIPS DE COLOMBIA S.A., c/o IPRELENSO LTD., Cra. 21, No. 56-17, BOGOTA, D.E., Tel. (01) 2 49 76 24.
Denmark: MINIWATT A/S, Strandlodsvej 2, P.O. Box 1919, DK 2300 COPENHAGEN S, Tel. (01) 54 11 33.
Finland: OY PHILIPS AB, Elcoma Division, Kaivokatu 8, SF-00100 HELSINKI 10, Tel. (90) 1 72 71.
France: RTC-COMPELEC, 130 Avenue Ledru Rollin, F-75540 PARIS 11, Tel. (01) 43 38 80 00.
Germany (Fed. Republic): VALVO, UB Bauelemente der Philips G.m.b.H., Valvo Haus, Burchardstrasse 19, D-2 HAMBURG 1, Tel. (040) 3296-0.
Greece: PHILIPS HELLENIQUE S.A., Elcoma Division, No. 15, 25th March Street, GR 17778 TAVROS, Tel. (01) 48 94 339/48 94 911.
Hong Kong: PHILIPS HONG KONG LTD., Elcoma Div., 15/F Philips Ind. Bldg., 24-28 Kung Yip St., KWAI CHUNG, Tel. (0)-24 51 21.
India: PEICO ELECTRONICS & ELECTRICALS LTD., Elcoma Dept., Band Box Building,
254-D Dr. Annie Besant Rd., BOMBAY - 400 025, Tel. (022) 4930311/4930590.
Indonesia: P.T. PHILIPS-RALIN ELECTRONICS, Elcoma Div., Setiabudi II Building, 6th Fl., Jalan H.R. Rasuna Said (P.O. Box 223/KBY) Kuningan,
JAKARTA 12910, Tel. (021) 51 79 95.
Ireland: PHILIPS ELECTRICAL (IRELAND) LTD., Elcoma Division, Newstead, Clonskeagh, DUBLIN 14, Tel. (01) 69 33 55.
Italy: PHILIPS S.p.A., Div. Componenti Elcoma, Piazza IV Novembre 3, I-20124 MILANO, Tel. (02) 6752.1.
Japan: NIHON PHILIPS CORP., Shuwa Shinagawa Bldg., 26-33 Takanawa 3-chome, Minato-ku, TOKYO (108), Tel. (03) 448-5611.
(IC Products) SIGNETICS JAPAN LTD., 8-7 Sanbancho Chiyoda-ku, TOKYO 102, Tel. (03) 230-1521.
Korea (Republic of): PHILIPS ELECTRONICS (KOREA) LTD., Elcoma Div., Philips House, 260-199 Itaewon-dong, Yongsan-ku, SEOUL,
Tel. (02) 794-5011.
Malaysia: PHILIPS MALAYSIA SDN BHD, Elcoma Div., 345 Jalan Gelugor, 11700 PULAU PINANG, Tel. (04) 87 00 44.
Mexico: ELECTRONICA, S.A de C.V., Carr. México-Toluca km. 62.5, TOLUCA, Edo. de México 50140, Tel. Toluca 91 (721) 613-00.
Netherlands: PHILIPS NEDERLAND, Marktgroep Elonco, Postbus 90050, 5600 PB EINDHOVEN, Tel. (040) 78 37 49.
New Zealand: PHILIPS NEW ZEALAND LTD., Elcoma Division, 110 Mt. Eden Road, C.P.O. Box 1041, AUCKLAND, Tel. (09) 605-914.
Norway: NORSK A/S PHILIPS, Electronica Dept., Sandstuveien 70, OSLO 6, Tel. (02) 68 02 00.
Pakistan: PHILIPS ELECTRICAL CO. OF PAKISTAN LTD., Philips Markaz, M.A. Jinnah Rd., KARACHI-3, Tel. (021) 72 57 72.
Peru: CADESA, Av. Alfonso Ugarte 1268, LIMA 5, Tel. (014) 326070.
Philippines: PHILIPS INDUSTRIAL DEV. INC., 2246 Pasong Tamo, P.O. Box 911, Makati Comm. Centre, MAKATI-RIZAL 3116, Tel. (02) 86 89 51 to 59.
Portugal: PHILIPS PORTUGUESA S.A.R.L., Av. Eng. Duarte Pacheco 6, 1009 LISBOA Codex, Tel. (019) 68 31 21.
Singapore: PHILIPS PROJECT DEV. (Singapore) PTE LTD., Elcoma Div., Lorong 1, Toa Payoh, SINGAPORE 1231, Tel. 35 02 000.
South Africa: S.A. PHILIPS (Pty) LTD., EDAC Div., 3rd Floor Rainer House, Upper Railway Rd. & Ove St., New Doornfontein, JOHANNESBURG 2001,
Tel. (011) 402-4600/07.
Spain: MINIWATT S.A., Balmes 22, BARCELONA 7, Tel. (03) 301 63 12.
Sweden: PHILIPS KOMPONENTER A.B., Lidingövägen 50, S-11584 STOCKHOLM 27, Tel. (08) 7821000.
Switzerland: PHILIPS A.G., Elcoma Dept., Allmendstrasse 140-142, CH-8027 ZÜRICH, Tel. (01) 488 22 11.
Taiwan: PHILIPS TAIWAN LTD., 150 Tun Hua North Road, P.O. Box 22978, TAIPEI, Taiwan, Tel. (02) 7120500.
Thailand: PHILIPS ELECTRICAL CO. OF THAILAND LTD., 283 Silom Road, P.O. Box 961, BANGKOK, Tel. (02) 233-6330-9.
Turkey: TÜRK PHILIPS TICARET A.S., Elcoma Department, Inönü Cad., No. 78-80, 80090 Ayazpasa ISTANBUL, Tel. (01) 143 59 10.
United Kingdom: MULLARD LTD., Mullard House, Torrington Place, LONDON WC1E 7HD, Tel. (01) 580 6633.
United States: (Active Devices & Materials) AMPEREX SALES CORP., Providence Pike, SLATERSVILLE, R.I. 02876, Tel. (401) 762-9000.
(Passive & Electromech. Dev.) MEPCO/CENTRALAB, INC., 2001 West Blue Heron Blvd, RIVIERA BEACH, Florida 33404,
Tel. (305) 881-3200.
(IC Products) SIGNETICS CORPORATION, 811 East Arques Avenue, SUNNYVALE, CA 94088-3409, Tel. (408) 991-2000.
Uruguay: LUZILECTRON S.A., Avda Uruguay 1287, P.O. Box 907, MONTEVIDEO, Tel. (02) 98 53 95.
Venezuela: IND. VENEZOLANAS PHILIPS S.A., c/o MAGNETICA S.A., Calle 6, Ed. Las Tres Jotas, App. Post. 78117, CARACAS, Tel. (02) 239 39 31.

For all other countries apply to: Philips Electronic Components and Materials Division, International Business Relations, P.O. Box 218,
5600 MD EINDHOVEN, The Netherlands, Telex 35000 phtcnl

AB54

Printed in The Netherlands

9398 146 40011