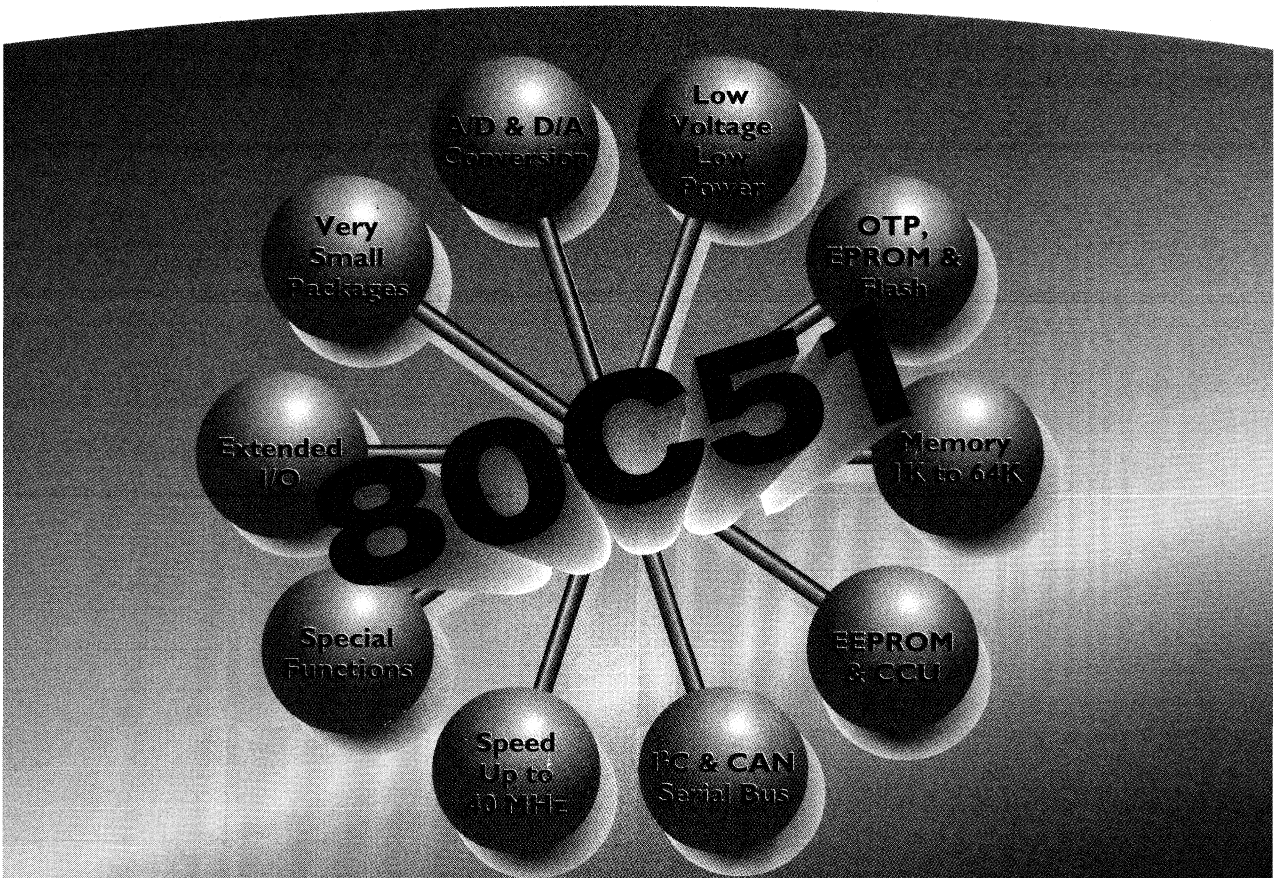
TEGRATED CIRCUITS

80C51-Based 8-Bit Microcontrollers

1998
Data Handbook IC20
CD-ROM included



PHILIPS

Let's make things better.

ttp://www.semiconductors.philips.com

QUALITY ASSURED

Our quality system focuses on the continuing high quality of our components and the best possible service for our customers. We have a three-sided quality strategy: we apply a system of total quality control and assurance; we operate customer-oriented dynamic improvement programmes; and we promote a partnering relationship with our customers and suppliers.

PRODUCT SAFETY

In striving for state-of-the-art perfection, we continuously improve components and processes with respect to environmental demands. Our components offer no hazard to the environment in normal use when operated or stored within the limits specified in the data sheet.

Some components unavoidably contain substances that, if exposed by accident or misuse, are potentially hazardous to health. Users of these components are informed of the danger by warning notices in the data sheets supporting the components. Where necessary the warning notices also indicate safety precautions to be taken and disposal instructions to be followed. Obviously users of these components, in general the set-making industry, assume responsibility towards the consumer with respect to safety matters and environmental demands.

All used or obsolete components should be disposed of according to the regulations applying at the disposal location. Depending on the location, electronic components are considered to be 'chemical', 'special' or sometimes 'industrial' waste. Disposal as domestic waste is usually not permitted.

80C51-Based 8-Bit Microcontrollers

CONTENTS

		page
SECTION 1	GENERAL INFORMATION	5
SECTION 2	80C51 TECHNICAL DESCRIPTION	41
SECTION 3	80C51 FAMILY DERIVATIVES	141
SECTION 4	HIGH PERFORMANCE 16-BIT 80C51 XA (eXtended Architecture)	247
SECTION 5	INTER–INTEGRATED CIRCUIT (I^2C) BUS	249
SECTION 6	I^2C SERIAL BUS APPLICATION NOTES & ARTICLES	251
SECTION 7	87C750, 8XC751, 8XC752 APPLICATION NOTES	257
SECTION 8	80C51 APPLICATION NOTES AND ARTICLES	263
SECTION 9	CONTROL AREA NETWORK (CAN) BUS	283
SECTION 10	80C51 SUPPORT CHIPS	297
SECTION 11	DEVELOPMENT SUPPORT	301
SECTION 12	PACKAGE OUTLINE	307
SECTION 13	DATA HANDBOOK SYSTEM	333
APPENDIX	PIN CONFIGURATION/COMPATIBILITIES	337

DEFINITIONS

Data Sheet Identification	Product Status	Definition (Note)
Objective Specification	Formative or in Design	This data sheet contains the design target or goal specifications for product development. Specifications may change in any manner without notice.
Preliminary Specification	Preproduction Product	This data sheet contains preliminary data, and supplementary data will be published at a later date. Philips Semiconductors reserves the right to make changes at any time without notice in order to improve design and supply the best possible product.
Product Specification	Full Production	This data sheet contains Final Specifications. Philips Semiconductors reserves the right to make changes at any time without notice, in order to improve design and supply the best possible product.
Short-form specification	—	The data in this specification is extracted from a full data sheet with the same type number and title. For detailed information see the relevant data sheet or data handbook.
Limiting values		
Limiting values given are in accordance with the Absolute Maximum Rating System (IEC 134). Stress above one or more of the limiting values may cause permanent damage to the device. These are stress ratings only and operation of the device at these or at any other conditions above those given in the Characteristics sections of the specification is not implied. Exposure to limiting values for extended periods may affect device reliability.		
Application information		
Applications that are described herein for any of these products are for illustrative purposes only. Philips Semiconductors makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification		

LIFE SUPPORT APPLICATIONS

These products are not designed for use in life support appliances, devices, or systems where malfunction of these products can reasonably be expected to result in personal injury. Philips Semiconductors customers using or selling these products for use in such applications do so at their own risk and agree to fully indemnify Philips Semiconductors for any damages resulting from such improper use or sale.

PURCHASE OF PHILIPS I^2C COMPONENTS

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. This specification can be ordered using the code 9398 393 40011.

DISCLAIMER

Philips Semiconductors reserves the right to make changes, without notice, in the products, including circuits, standard cells, and/or software, described or contained herein in order to improve design and/or performance. Philips Semiconductors assumes no responsibility or liability for the use of any of these products, conveys no license or title under any patent, copyright, or mask work right to these products, and makes no representations or warranties that these products are free from patent, copyright, or mask work right infringement, unless otherwise specified.

NOTE: Always check with your local Philips Semiconductors Sales Office to be certain that you have the latest data sheet(s) before completing a design.

Preface

Microcontrollers from Philips Semiconductors

Philips Semiconductors continues to innovate its product offering with new and exciting 8 and 16-bit microcontrollers. The result has been thicker and thicker data books to hold all of these products. Rather than continue this progression, we have taken a different tack which combines the use of the Internet and computer CD ROMs with the basic information available in data book form. The IC20 Data Book contains the descriptions, feature listing, pinouts for each of the devices, and includes more extensive selection guides, so that you can identify the device that interests you. Once you find this device, you can print a copy of the datasheet from the CD ROM that is enclosed, download a copy from the Internet, or have one faxed to you from our FAX-on-DEMAND system. Datasheets on the Internet and FAX-on-DEMAND are constantly updated. We hope that you will find this a much more efficient and up-to-date method for getting the various device information that you require.

Philips Semiconductors 8 and 16-bit microcontrollers are based on the widely-accepted 8048, 8051 and XA architectures. We offer most of the 'industry standard' products in these architectures as well as a large selection of powerful derivative products. These derivatives offer a wide assortment of features, including: additional memory, A/D, PWM, additional timers, DTMF, OSD, OTP, EMC and EMI, plus many others. The variety of product derivatives allows Philips Semiconductors to support a broad range of functions in consumer, telecom, EDP, multi media, automotive and industrial applications.

For details, see:

- 8048 'industry standard' architecture types (PCF84CXXX family) in *"Data Handbook IC14"*.
  The PCD33XX family covers telecom terminal family devices based on the 8048 core and instruction set, in *"Data Handbook IC03"*.
- 8051 'industry standard' architecture types in *"Data Handbook IC20"*.
- XA types in *"Data Handbook IC25"*.

Systems are requiring lower power along with lower operating voltages. In addition, applications are becoming more memory intensive. Philips "New and Improved" family of 80C51 Microcontrollers provide a solution for these applications. These devices are now specified for operation from 2.7V to 5.5V@16MHz, and can operate over a wider operating frequency range because the CPU core uses a static design, and consumes 50% less supply current than the previous devices. Memory sizes are available with up to 1K RAM and 64K ROM/EPROM.

The Low Power 80CL51 family of derivatives can be found in *"Data Handbook IC20"*. These devices operate over the wider voltage range of 1.8 to 6.0V and are ideal for portable and battery operations.

Many of Philips Semiconductors ICs offer on-board UART serial ports and I^2C-bus. The I^2C-bus allows easy connection to over 100 other devices, thereby increasing system capabilities even further. For automotive and industrial applications, we also offer the CAN and the VAN serial bus. The CAN standard, developed by Bosch, and VAN concepts offer high noise immunity and error correction.

Philips Semiconductors 16-bit microcontroller family is based on the XA architecture. The XA is upwards compatible with the 80C51 and offers users an easy migration path to higher performance. The XA's compatibility with the 80C51 has in no way limited the performance of the XA, which is one of the highest performance 16-bit microcontrollers available.

Section 1
General Information

80C51-Based
8-Bit Microcontrollers

CONTENTS

Contents 7

Quality 11

Selection guide for the 80C51 microcontroller family 12

80C51 microcontroller family features guide 24

8051 microcontroller cross-reference guide 28

Low power / low voltage microcontroller family 29

CMOS and NMOS 8-bit microcontroller family 31

CMOS 16-bit microcontroller family 34

FAX-on-DEMAND System 35

Microcontroller internet and bulletin board access 36

Ordering Information 38

CONTENTS

IC20: 80C51-BASED 8-BIT MICROCONTROLLERS

Preface 3

Section 1 – General Information

Contents 7
Quality 11
Selection guide for the 80C51 microcontroller family 12
80C51 microcontroller family features guide 24
8051 microcontroller cross-reference guide 28
Low power / low voltage microcontroller family 29
CMOS and NMOS 8-bit microcontroller family 31
CMOS 16-bit microcontroller family 34
FAX-on-DEMAND System 35
Microcontroller internet and bulletin board access 36
Ordering Information 38

Section 2 – 80C51 Technical Description

80C51 family architecture 43
80C51 family hardware description 58
80C51 family programmer's guide and instruction set 82
80C51 family EPROM products 137

Section 3 – 80C51 Family Derivatives

8XC51/80C31	8-bit CMOS (low voltage, low power, and high speed) microcontroller families	143
8XC52/54/58/80C32 8XC51FA/FB/FC/80C51FA 8XC51RA+/RB+/RC+/RD+/ 80C51RA+	8-bit CMOS (low voltage, low power and high speed) microcontroller families	146
89C52/54/58 89C51RA+/RB+/RC+/RD+	8-bit CMOS microcontroller families with FLASH program memory	151
83C145; 83C845; 83C055; 87C055	Microcontrollers for TV and video (MTV)	154
P83Cx80; P87C380	Microcontrollers for monitors with DDC interface, auto-sync detection and sync proc.	156
P80CL31; P80CL51	Low voltage 8-bit microcontrollers with UART	159
P80CL410; P83CL410	Low voltage 8-bit microcontrollers with I^2C-bus	162
P83C434; P83C834	8-bit microcontrollers with LCD-driver	164
80C451/83C451/87C451	CMOS single-chip 8-bit microcontrollers	166
83C453/87C453	CMOS single-chip 8-bit microcontrollers	169
87C524	CMOS single-chip 8-bit microcontroller	171
87C528	CMOS single-chip 8-bit microcontroller	173
P83C524; P80C528; P83C528	8-bit microcontrollers	175
80C550/83C550/87C550	CMOS single-chip 8-bit microcontroller with A/D and watchdog timer	179
80C552/83C552	Single-chip 8-bit microcontroller with 10-bit A/D, capture/compare timer, high-speed outputs, PWM	181
87C552	Single-chip 8-bit microcontroller with 10-bit A/D, capture/compare timer, high-speed outputs, PWM	184
P83CE558/P80CE558/P89CE558	Single-chip 8-bit microcontroller	186
P83CE559/P80CE559	Single-chip 8-bit microcontroller	188
P8xCE560	8-bit microcontroller	190
P83C562; P80C562	8-bit microcontroller	193
80C575/83C575/87C575	CMOS single-chip 8-bit microcontrollers	195
83C576/87C576	CMOS single-chip 8-bit microcontrollers	197
P80CL580; P83CL580	Low voltage 8-bit microcontrollers with uart, I^2C-bus and ADC	199
P8XC592	8-bit microcontroller with on-chip CAN	(See Section 9)
P8XCE598	8-bit microcontroller with on-chip CAN	(See Section 9)
80C652/83C652	CMOS single-chip 8-bit microcontrollers	201

87C652	CMOS single-chip 8-bit microcontroller	205
83C654	CMOS single-chip 8-bit microcontroller	207
87C654	CMOS single-chip 8-bit microcontroller	211
P89C738; P89C739	8-bit microcontrollers	213
83C748/87C748	CMOS single-chip 8-bit microcontrollers	215
83C749/87C749	CMOS single-chip 8-bit microcontrollers	217
TPM749	Microcontroller with TrackPoint™ microcode from IBM	219
83C750/87C750	CMOS single-chip 8-bit microcontrollers	221
83C751/87C751	CMOS single-chip 8-bit microcontrollers	223
83C752/87C752	CMOS single-chip 8-bit microcontrollers with A/D, PWM	225
83C754/87C754	CMOS single-chip 8-bit microcontrollers	227
TPM754	Microcontroller with TrackPoint™ microcode from IBM	229
P83CL781; P83CL782	Low voltage 8-bit microcontrollers with UART and I^{2}C-bus	231
80C851/83C851	CMOS single-chip 8-bit microcontroller with on-chip EEPROM	234
TELX family	Low voltage 8-bit microcontrollers	236
P83CL883; P87CL883; P83CL884; P87CL884	TELX microcontrollers for CT0 handset/basestation applications	238
P83CL886; P87CL886	TELX microcontrollers for CT0 handset/basestation applications	241
P83CL887; P87CL887	TELX microcontrollers for CT0 handset/basestation applications	243
P87CL881	Low voltage 8-bit microcontroller	245

Section 4 – High Performance 16-bit 80C51 XA (eXtended Architecture) (See CDROM)

80C51XA Architectural overview		(See CDROM)
XA-G1	CMOS single-chip 16-bit microcontroller	(See CDROM)
XA-G2	CMOS single-chip 16-bit microcontroller	(See CDROM)
XA-G3	CMOS single-chip 16-bit microcontroller	(See CDROM)
XA-S3	Single-chip 16-bit microcontroller	(See CDROM)

Section 5 – Inter-Integrated Circuit (I^{2}C) Bus (See CDROM)

The I^{2}C-bus and how to use it (including specifications)		(See CDROM)
I^{2}C peripheral selection guide		(See CDROM)
82B715	I^{2}C bus extender	(See CDROM)

Section 6 – I^{2}C Serial Bus Application Notes and Articles

AN422	Using the 8XC751 microcontroller as an I^{2}C bus master	253
AN425	Interfacing the PCF8584 I^{2}C-bus controller to 80C51 family microcontrollers	253
AN430	Using the 8XC751/752 in multimaster I^{2}C applications	253
AN433	I^{2}C slave routines for the 83C751	254
AN434	Connecting a PC keyboard to the I^{2}C-bus	254
AN435	I^{2}C byte oriented system driver	254
AN438	I^{2}C routines for 8XC528	255
AN444	Using the P82B715 I^{2}C extender on long cables	255
AN452	One mile long I^{2}C communication using the P82B715	255
EIE/AN91007	I^{2}C driver routines for 8XC751/2 microcontrollers	256
ETV/AN89004	PLM51 I^{2}C software interface IIC51 (version 0.5)	256

CONTENTS

Section 7 – 87C750, 8XC751, 8XC752 Application Notes

AN422 Using the 8XC751 microcontroller as an I²C bus master .... (See Section 6)
AN423 Software driven serial communication routines for the 83C751 and 83C752 microcontrollers ... 259
AN426 Controlling air core meters with the 87C751 and SA5775 259
AN427 Timer I for the 83/87C748/749 and the 83/87C751/752 (non-I²C applications) microcontrollers . 259
AN428 Using the ADC and PWM of the 83C752/87C752 260
AN429 Airflow measurement using the 83/87C752 and "C" 260
AN430 Using the 8XC751/752 in multimaster I²C applications .... (See Section 6)
AN433 I²C slave routines for the 83C751 .... (See Section 6)
AN435 I²C byte oriented system driver .... (See Section 6)
AN436 "Opti-Mizer" power management for notebook computers using the 8XC752 microcontroller . 260
AN439 87C751 fast NiCad charger 261
AN442 (BCM) 87C751: Specification for a bus-controlled monitor 261
AN446 A software duplex UART for the 751/752 261
AN453 Using the 87C751 microcontroller to gang program PCF8582/PCF8581 EEPROMs 262
AN454 Interfacing the 83C576/87C576 to the ISA bus 262
AN459 8XC750 watering controller 262
EIE/AN91007 I²C driver routines for 8XC751/2 microcontrollers .... (See Section 6)

Section 8 – 80C51 Application Notes and Articles

AN408 80C451 operation of port 6 265
AN417 256k Centronics printer buffer using the 87C451 microcontroller 265
AN418 Counter/timer 2 of the 83C552 microcontroller 265
AN420 Using up to 5 external interrupts on 80C51 family microcontrollers 266
AN424 8051 family warm boot determinations 266
AN440 RAM loader program for 80C51 family applications 266
AN443 IEEE Micro Mouse using the 87C751 microcontroller 267
AN447 Automatic baud rate detection for the 80C51 267
AN448 Determining baud rates for 8051 UARTs and other UART issues 267
AN449 Measure resistance and capacitance without an A/D 268
AN455 In-circuit programming of the Philips 87C576 microcontroller 268
AN456 Using LC oscillator circuits with Philips microcontrollers 268
AN457 80C51 External Memory Interfacing 269
AN458 Dual data pointers for '51 family 269
EIE/AN91001 Workbench EMC evaluation method 269
EIE/AN91006 A/D conversion with P83CL410 PCF1252-x 270
EIE/AN91009 Driver for 8xC851 E2PROM 270
EIE/AN92001 Low RF-emission applications with a P83CE654 microcontroller 270
EIE/AN93017 Using the analog-to-digital converter of the 8XC552 microcontroller 271
ESG89001 Electro magnetic compatibility and printed circuit board (PCB) constraints 271
Chips push CAN bus into embedded world 272
Add Text Overlay to Any Video Display 274

Section 9 – Control Area Network (CAN) Bus

Control Area Network (CAN) overview 285
P82C150 CAN Serial Linked I/O device (SLIO) with digital and analog port functions 286
PCA82C250 CAN controller interface 288
SJA1000 Stand-alone CAN controller 290
P8xC592 8-bit microcontroller with on-chip CAN 292
P8xCE598 8-bit microcontroller with on-chip CAN 294

CONTENTS

Section 10 – 80C51 Support Chips

PCF1252-X family 299

Section 11 – Development Support

Microcontroller development tool contact information 303

8-bit microcontroller demonstration and evaluation boards 305

Section 12 – Package Outlines

Soldering Package information 309

Plastic Dual In-Line Package

DIP8: plastic dual in-line package; 8 leads (300 mil) SOT97-1 311

DIP24: plastic dual in-line package; 24 leads (300 mil) .... SOT222-1 ... 312

DIP28: plastic dual in-line package; 28 leads (600 mil); long body .... SOT117-2.... 313

DIP40: plastic dual in-line package; 40 leads (600 mil) .... SOT129-1 ... 314

Plastic Shrink Dual In-Line Package

SDIP42: plastic shrink dual in-line package; 42 leads (600 mil) .... SOT270-1 ... 315

Plastic Leaded Chip Carrier

PLCC28: plastic leaded chip carrier; 28 leads; pedestal .... SOT261-3 ... 316

PLCC44: plastic leaded chip carrier; 44 leads .... SOT187-2 ... 317

PLCC68: plastic leaded chip carrier; 68 leads .... SOT188-2 ... 318

PLCC68: plastic leaded chip carrier; 68 leads; pedestal .... SOT188-3 ... 319

Plastic Quad Flat Package

QFP44: plastic quad flat package; 44 leads (lead length 1.3 mm); body 10 x 10 x 1.75 mm .... SOT307-2 ... 320

QFP44: plastic quad flat package; 44 leads (lead length 2.35 mm); body 14 x 14 x 2.2 mm .... SOT205-1 ... 321

LQFP44: plastic low profile quad flat package; 44 leads; body 10 x 10 x 1.4 mm .... SOT389-1 ... 322

QFP64: plastic quad flat package; 64 leads (lead length 1.95mm); body 14 x 20 x 2.8 mm .... SOT319-2 ... 323

QFP80: plastic quad flat package; 80 leads (lead length 1.95 mm); body 14 x 20 x 2.7 mm; high stand-off height .... SOT318-1 ... 324

QFP80: plastic quad flat package; 80 leads (lead length 1.95 mm); body 14 x 20 x 2.8 mm .... SOT318-2 ... 325

Plastic Small Outline Package

SO8: plastic small outline package; 8 leads; body width 3.9mm SOT96-1 326

VSO40: plastic very small outline package; 40 leads .... SOT158-1 ... 327

VSO56: plastic very small outline package; 56 leads .... SOT190-1 ... 328

SO28: plastic small outline package; 28 leads; body width 7.5mm .... SOT136-1 ... 329

Plastic Shrink Small Outline Package

SSOP24: plastic shrink small outline package; 24 leads; body width 5.3 mm .... SOT340-1 ... 330

SSOP28: plastic shrink small outline package; 28 leads; body width 5.3mm .... SOT341-1 ... 331

Section 13 – Data Handbook System

Data handbook system 335

Appendix

Pin Configuration/Compatibilities 337

TOTAL QUALITY MANAGEMENT

Philips Semiconductors is a Quality Company, renowned for the high quality of our products and service. We keep alive this tradition by constantly aiming towards one ultimate standard, that of zero defects. This aim is guided by our Total Quality Management (TQM) system, the basis of which is described in the following paragraphs.

Quality assurance

Based on ISO 9000 standards, customer standards such as Ford TQE and IBM MDQ. Our factories are certified to ISO 9000 by external inspectorates.

Partnerships with customers

PPM co-operations, design-in agreements, ship-to-stock, just-in-time and self-qualification programmes, and application support.

Partnerships with suppliers

Ship-to-stock, statistical process control and ISO 9000 audits.

Quality improvement programme

Continuous process and system improvement, design improvement, complete use of statistical process control, realization of our final objective of zero defects, and logistics improvement by ship-to-stock and just-in-time agreements.

ADVANCED QUALITY PLANNING

During the design and development of new products and processes, quality is built-in by advanced quality planning. Through failure-mode-and-effect analysis the critical parameters are detected and measures taken to ensure good performance on these parameters. The capability of process steps is also planned in this phase.

PRODUCT CONFORMANCE

The assurance of product conformance is an integral part of our quality assurance (QA) practice. This is achieved by:

- Incoming material management through partnerships with suppliers.
- In-line quality assurance to monitor process reproducibility during manufacture and initiate any necessary corrective action. Critical process steps are 100% under statistical process control.
- Acceptance tests on finished products to verify conformance with the device specification. The test results are used for quality feedback and corrective actions. The inspection and test requirements are detailed in the general quality specifications.
- Periodic inspections to monitor and measure the conformance of products.

PRODUCT RELIABILITY

With the increasing complexity of Original Equipment Manufacturer (OEM) equipment, components reliability must be extremely high. Our research laboratories and development departments study the failure mechanisms of semiconductors. Their studies result in design rules and process optimization for the highest built-in product reliability. Highly accelerated tests are applied to the product reliability evaluation. Rejects from reliability tests and from customer complaints are submitted to failure analysis, to result in corrective action.

CUSTOMER RESPONSES

Our quality improvement depends on joint action with our customer. We need our customer's inputs and we invite constructive comments on all aspects of our performance. Please contact our local sales representative.

RECOGNITION

The high quality of our products and services is demonstrated by many Quality Awards granted by major customers and international organizations.

Selection guide for the 80C51 microcontroller family

Now you can choose among the largest selection of feature-rich 80C51 derivatives in the world.

Philips has developed the broadest line of derivative devices in the industry, all based on the 80C51 core architecture. Designed for real-time applications, these 80C51 derivative products are used in applications ranging from consumer products and computer peripherals to automotive systems and aerospace designs.

When you need a full range of memory options for your microcontroller applications, turn to Philips 80C51 derivatives for: FLASH, OTP-EPROM, and ROM from 1K to 64K, and ROMless versions that can address up to 64K bytes of external memory.

For applications operating on battery power, low-power devices are essential. Philips has a growing family of 80C51 microcontrollers that operate as low as 2.7V. For applications requiring lower voltage, the 8XCL51 family operates as low as 1.8V-significantly lower than most other 80C51microcontrollers on the market. Philips' low-power microcontrollers also have idle and power-down modes that can reduce total power consumption and allow you to get maximum battery life from your application. Philips has also reduced the supply current of the 80C51/52/54/58 and FX series by 50%. So, Philips' low-power microcontrollers give you an important advantage over your competitors.

Because today's 8-bit designs require increasing performance, Philips has added 80C51 devices with increased performance and additional features. However, many designs require a dramatic performance increase. These designs can only be satisfied by a powerful 16-bit core. Recognizing the fact that thousands of customers currently use 80C51 in their applications and have a considerable investment in code libraries and development tools, Philips created a 16-bit architecture that enables customers to leverage their investment and move to 16-bit performance along an upwardly compatible growth path. The Philips eXtended Architecture (XA) enables the migration of 80C51 devotees to 16-bit microcontroller designs. The XA is much more than a simple 16-bit extension of the 80C51; it provides four to five times the performance of an 8-bit 80C51 running at the same clock frequency, with the same source code. The XA provides a 16-bit performance punch that many of today's applications demand while maintaining the 80C51 development investment.

To make your selection of Philips 80C51 derivative microcontrollers easy, the following pages summarize the available devices in seven key categories. A complete listing of the full family of Philips 80C51 microcontrollers, divided by memory size, may be found on the last four pages of this brochure. Each category features key design solutions and highlights individual product features.

These feature categories include:

Memory and Speed

Devices listed in this section by memory size include the 8XC750, the smallest 80C51 with 1K of ROM/OTP, to the largest memory device 8XCE560 and P8XC51RD+ with 64K ROM/OTP/FLASH. For applications requiring a small package footprint, Philips offers the PQFP and the 7XX series in the very small SSOP. Speeds are on the increase with most devices specified for 33MHz and 750 at 40MHz.

Counters/Timers

Philips offers the widest selection of derivatives with built-in counters and timers including Watchdog timers and Programmable Counter Arrays(PCA).

Serial Interface

Offering more serial interface options than any other supplier, Philips microcontrollers feature I^2C, UART, enhanced UART, and CAN bus interface options to meet your needs.

Low Voltage Operation

Philips offers a full family of low voltage derivatives including low volt OTP devices optimized at 16MHz.

Analog Features

With a variety of devices featuring 8- and 10-bit A/D converters, Philips has the products to meet your specific analog-to-digital application needs.

Protection Capabilities and Reduced EMI/RFI

Philips offers several built-in protection circuits such as Watchdog timers, oscillator failure detection, and low V_{CC} detection capabilities. And, as the demand for reduced EMI products rises, Philips family of devices with low EMI/RFI grows to meet that demand.

Additional Features

Many of Philips unique additional features are detailed in this section including: On-Screen Display, Universal Peripheral Interface, and Smart Card.

Microcontroller Characteristic Guide

Philips created the *Microcontroller Characteristic Guide* as a quick-reference for the full line of Philips 80C51 derivatives.

Selection guide for the 80C51 microcontroller family

MEMORY AND SPEED

The Philips family of 80C51 derivative microcontrollers provides the broadest range of memory and speed capabilities offered by any supplier.

OTP-EPROM

Philips offers more OTPs than any other microcontroller supplier, providing effective solutions for prototyping, low- and high-volume production runs, and applications requiring unique codes.

At each step of the design cycle, Philips' broad range of user-programmable devices provides outstanding flexibility and choice. From the software development stage through device emulation and the system debug stage, our FLASH devices are the most effective solutions to your design challenges. Because these devices are electrically erasable, you can reuse them until your design is perfected.

Early in the production phase, OTP microcontrollers are ideal because they provide the flexibility to make software changes quickly and easily. In many applications, OTP devices are being used for volume production runs because they allow you to respond quickly to changes in production requirements. And, if you have to make a software change during production, OTP devices allow you to put the software changes into production at a lower cost-because there are no NRE (Non-Recurring Engineering) charges, no work-in-process charges, no need to scrap large amounts of inventory, and none of the long lead times required by mask devices. Best of all, OTP microcontrollers mean you can respond to your market quickly.

FLASH

FLASH microcontrollers offer maximum flexibility in user programmability. FLASH microcontrollers can be electrically erased and reprogrammed again and again. As a user you can determine when in the production flow the devices should be programmed. And if revisions are needed, the device can be reprogrammed with new updated software.

The FLASH devices can be programmed with the standard parallel programmer or depending on the device they can be serially programmed. Parallel programmable devices are primarily programmed out of the application. The device is probably mounted on the PC board in a socket. To upgrade the devices software, the device is removed from the socket, reprogrammed, and then put back into the socket. Programming the device out of the application eliminates the requirement to have circuitry on the PC board to enable programming. The P89C51RX+ Family, P89C738, P89C138, P89C238, and P89C132 can be parallel programmed.

Serial programming devices makes it simpler to program devices permanently mounted on the PC board and reduces the amount of programming hardware when compared to the Parallel method. There are two Serial programming methods. First method is controlled external to the application. This normally is done through the use of an external connecter mounted on the PC board. The P89C138/238 uses this method.

The other serial programming method uses the device to control the programming. The code is updated through the standard UART or I/O configured as the serial port. In this method the processor handles the communication to call the program updates and then erases and programs the FLASH memory. A minimum of programming hardware has to be added to accomplish the erase and reprogram. Small sections of code can be changed in the P89C51RX+ and P89C132. The boot handler can be customized by the user depending on the application requirements.

EEPROM

EEPROM memory means you decide when you want to program-either in circuit or in a programmer.

With the Philips 8XC864, 8XC858, 8XC855, 83C852 and 8XC851 we are now offering a range of 80C51 microcontrollers that have EEPROM data memory. The EEPROM data memory that stores critical data even when the power is turned off.

Speed and Performance

Philips has extended its line of high performance 80C51 microcontrollers that run at speeds up to 33MHz and, additionally, offers the 8XC750 which operates at 40MHz. This means you may not have to move up to a more expensive microcontroller architecture to get the processing power required. These new microcontrollers which operate at speeds up to 33MHz require 50% less supply currents than previous designs.

If your design requires low power dissipation and speed is not that important, you can select from devices that operate down to 0 frequency, feature static operation, and operate down to 2.7V or 1.8V. Philips developed processing capability that yields high speed and low voltage operation. Philips is the first to offer OTPs with 2.7V operation without sacrificing speed. Philips has extended the performance features of standard devices (8XC51/52/54/58 and FX series) to operate down to 2.7V.

If your design requires more performance than you can obtain from the 80C51 product line, you can select from a compatible, high-performance, 16-bit microcontroller family. The Philips eXtended Architecture (XA) family of microcontrollers offers 10 to 100 times the performance of the 80C51 allowing an easy migration up the performance curve.

Selection guide for the 80C51 microcontroller family

MEMORY AND SPEED

Memory Size	Part Number	OTP-FLASH EPROM (bytes)	ROM (bytes)	RAM (bytes)	External Memory Capability	Speed (MHz) Min–Max	Comments/Special Features
1K	87C750/83C750	1K	1K	64	No	3.5–40	40MHz, 24-Pin Skinny DIP and SSOP
2K	87C748/83C748	2K	2K	64	No	3.5–16	24-Pin Skinny DIP Package and SSOP
	87C751/83C751	2K	2K	64	No	3.5–16	24-Pin Skinny DIP Package and SSOP
	87C749/83C749	2K	2K	64	No	3.5–16	Pulse Width Modulation and SSOP
	87C752/83C752	2K	2K	64	No	3.5–16	Pulse Width Modulation and SSOP
4K	87C754/83C754	4K	4K	256	No	3.5–16	8-bit DAC, Comparator, Reference and MUX Input
	87C51/80C51/80C31	4K	4K	128	Yes	0–33	ROM/ROMless: 2.7V to 5.5V @ 16MHz; Freq. to 0Hz
	80CL51/80CL31	4K	4K	128	Yes	0–16	Low Voltage/Power (1.8V to 6V)
	83CL410/80CL410	4K	4K	128	Yes	0–12	Low Voltage/Power (1.8V to 6V)
	87C451/83C451/80C451	4K	4K	128	Yes	3.5–16	7 I/O Ports, Processor Bus Interface
	87C550/83C550/80C550	4K	4K	128	Yes	3.5–16	Watchdog Timer to reset the microcontroller
	83C851/80C851	4K	4K	128	Yes	1.2–16	256 Bytes EEPROM, 80C51 Pin-Compatible
6K	83C852/80C852	6K	6K	256	No	1–6	Smart Card, 2K EEPROM, CCU
	83CL580/80CL580	6K	6K	256	Yes	0–12	Watchdog Timer, Pulse Width Modulation, Low Voltage (2.5V to 6V)
8K	89C52/87C52/80C52/80C32	8K	8K	256	Yes	0–33	2.7V to 5.5V @ 16MHz; Freq. to 0Hz
	83C845		8K	256	No	3.5–12	OSD, 9 Pulse Width Modulation output
	S87C652/P83C652/P80C652	8K	8K	256	Yes	1.2–24	I^2C Serial Bus
	87C51FA/83C51FA/80C51FA	8K	8K	256	Yes	0–33	Industry Standard; 2.7V to 5.5V @ 16MHz; Freq. to 0 Hz
	89C51RA+/87C51RA+/ 83C51RA+/80C51RA+	8K	8K	512	Yes	0–33	2.7 to 5.5V @ 16MHz; Freq. to 0Hz; FLASH (89) 5V only
	87C575/83C575/80C575	8K	8K	256	Yes	4–16	8XC575/8XC576 extended Feature Set includes Low V_{CC} Detect, Low Active Reset, Oscillator Fail Detect, Reduced EMI and 4 Analog Comparators.
	87C576/83C576/80C576	8K	8K	256	Yes	6–16	UPI, on-board programming, 8XC575/8XC576 extended Feature Set includes Low V_{CC} Detect, Low Active Reset, Oscillator Fail Detect, Reduced EMI and 4 Analog Comparators.
	83C562/80C562	—	8K	256	Yes	8.5–16	Pulse Width Modulation, Watchdog Timer, Capture/Compare Counter/Timer with High Speed Output
	87C552/83C552/80C552	8K	8K	256	Yes	1.2–30	Pulse Width Modulation, Watchdog Timer, Capture/Compare Counter/Timer with High Speed Output
	87C453/83C453	8K	8K	256	Yes	3.5–16	Processor bus interface
12K	83C145		12K	256	No	3.5–12	OSD, 9 Pulse Width Modulation output

Selection guide for the 80C51 microcontroller family

MEMORY AND SPEED (continued)

Memory Size	Part Number	OTP-FLASH EPROM (bytes)	ROM (bytes)	RAM (bytes)	External Memory Capability	Speed (MHz) Min–Max	Comments/Special Features
16K	87C055/83C055	16K	16K	256	No	3.5–12	Reduced EMI, OSD, 9 PWM Outputs
	89C54/87C54/80C54	16K	16K	256	Yes	0–33	2.7V to 5.5V @ 16MHz; Freq. to 0Hz (Ex. (89C))
	S87C654/P83C654	16K	—	256	Yes	1.2–24	I^{2}C Serial Bus
	83CL781	—	16K	256	Yes	0–12	Low Voltage/Power (1.8V to 6V)
	83CL782	—	16K	256	Yes	0–12	83CL781 optimized for 12MHz @ 3.1V
	87C51FB/83C51FB	16K	16K	256	Yes	0–33	Industry Standard; 2.7V to 5.5V @ 16MHz; Freq. to 0Hz
	89C51RB+/87C51RB+/ 83C51RB+	16K	16K	512	Yes	0–33	2.7V to 5.5V @ 16MHz; Freq. to 0Hz; FLASH 5V only
	87C524/83C524	16K	—	512	Yes	1.2–16	512 RAM, Watchdog Timer
	83C858	—	16K	512	No	1.2–10	8K Bytes EEPROM, Smart Card
	83C592/80C592	16K	16K	512	Yes	1.2–16	CAN 2.0A, Watchdog Timer, Capture/Compare Counter/Timer with High Speed Output, PWM
32K	89C58/87C58/80C58	32K	32K	256	Yes	0–33	2.7V to 5.5V @ 16MHz; Freq. to 0Hz
	87C51FC/83C51FC	32K	32K	256	Yes	0–33	Industry Standard, 2.7V to 5.5V @ 16MHz;
	89C51RC+/87C51RC+/ 83C51RC+	32K	32K	512	Yes	0–33	Freq. to 0Hz; FLASH (89C) 5V only
	87C528/83C528/80C528	32K	32K	512	Yes	2.5–20	Large memory for high level languages, Watchdog Timer
	83CE598/80CE598	32K	32K	512	Yes	1.2–16	CAN bus, Watchdog Timer, T2, PWM, reduced EMI
	89CE558/83CE558/80CE558	32K		1024	Yes	3.5–16	Watchdog Timer, Pulse Width Modulation, Low EMI
64K	89C51RD+/87C51RD+/ 83C51RD+	64K	6K	1024	Yes	0–33	2.7V to 5.5V; Freq. to 0Hz; FLASH 5V only
XA Family							
32K	P51XAG30/G37/G33	32K	32K	512	Yes	0–30	High performance, 16-bit 80C51 compatible
	P51XAS3		32K	1024	Yes	30	16MByte Address Range

COUNTERS/TIMERS

To better meet your design needs, Philips offers a wide range of timer configurations on many of its microcontrollers, including the three standard 16-bit 80C51 timers: Timer 0, Timer 1, and Timer 2. We also offer enhancements to these standard timers on selected products that feature other specialized timers, such as PCAs and hardware Watchdog timers.

PCA

The Philips PCA (Programmable Counter Array) is identical in function to the industry-standard PCA and offers significant timing advantages over standard 80C51 timers.

A PCA is a special timer with up to five associated 16-bit capture/compare modules. Offered as an integral part of the device to enable you to simplify your software design, each module of the PCA can be programmed individually to operate in one of four modes: rising- and/or falling-edge capture, software timer, high-speed output, or pulse width modulator. And each PCA module has an associated pin in Port 1 which you can use as input or output for that module.

All of our microcontrollers that feature a PCA, such as the 8XC51FX and 8XC51RX+ family, the 8XC754, the 8XC575 and the 8XC576, are ideal for applications requiring pulse width modulation, high-speed I/O, and counting capabilities. Some key applications most likely to benefit from the PCA are automotive control, cellular and cordless phones, high-end storage device controls, instrumentation, medical diagnostics, power supply control and sequencing, and monitoring equipment.

Hardware Watchdog Timer

The Philips hardware Watchdog timer assures you that the microcontroller in your application will continue to operate properly.

Watchdog timers protect your design from disruptions to microcontroller execution, which can be caused by voltage spiking or other external conditions. If a Watchdog timer is allowed to overflow, it will reset the microcontroller. To prevent an overflow, a Watchdog timer must be cleared before it overflows. A specific load sequence is used to clear the timer so it can't be cleared accidentally. The time-out period for the Philips Watchdog timer is adjustable from 2,000 instruction cycles to 512,000. (See also Protection Capabilities section for more information on the special features of the Philips Watchdog timer.)

Selection guide for the 80C51 microcontroller family

COUNTERS/TIMERS

Device		Timer Types					Capture	Compare	High-speed Outputs
1 Timer	87C750/83C750	ET0							
	87C751/83C751	ET0							
	87C752/83C752	ET0							
2 Timers	87C748/83C748	ET0	TR						
	87C749/83C749	ET0	TR						
	87C51/80C51/80C31	T0	T1						
	80CL51/80CL31	T0	T1						
	83CL410/80CL410	T0	T1						
	87C451/83C451/80C451	T0	T1						
	87C550/83C550/80C550	T0	T1						
	83C851/80C851	T0	T1						
	83C852/80C852	T0	T1						
	S87C652/P83C652/P80C652	T0	T1						
	87C453/83C453/80C453	T0	T1			Watchdog			
	87C055/83C055	T0	T1						
	S87C654/P83C654	T0	T1						
	83C845	TO	TI						
	83C145	TO	TI						
3 Timers	83CL580/80CL580	T0	T1	T2		Watchdog	X		
	87C52/80C52/80C32	T0	T1	T2			X		
	89C52/87C552/83C562/80C562	T0	T1	ST2		Watchdog	X	X	X
	87C552/83C552/80C552	T0	T1	ST2		Watchdog	X	X	X
	89C54/87C54/80C54	T0	T1	T2					
	83CL781	T0	T1	T2			X		
	83CL782	T0	T1	T2			X		
	87C524/83C524	T0	T1	T2		Watchdog	X		
	87C592/83C592/80C592	T0	T1	T2		Watchdog	X	X	X
	89C51RB+/87C51FC/83C51FC	T0	T1	T2	PCA	X	X	X	
	87C528/83C528/80C528	T0	T1	T2		Watchdog	X		
	83CE598/80CE598	T0	T1	T2		Watchdog	X	X	X
	87CE558/83CE558/80CE558	T0	T1	ST2		Watchdog	X	X	X
	87C754/83C754	T0	T1	PCA	X	X	X		
4 Timers	87C51FA/83C51FA/80C51FA	T0	T1	ET2	PCA	X	X	X	
	89C51RA+/87C51RA+/ 83C51RA+/80C51RA+	T0	T1	ET2	PCA	Watchdog	X	X	
	87C575/83C575/80C575	T0	T1	ET2	PCA	Watchdog	X	X	X
	87C576/83C576/80C576	T0	T1	ET2	PCA	Watchdog	X	X	X
	87C51FB/83C51FB	T0	T1	ET2	PCA	X	X	X	
	87C51FC/83C51FC	T0	T1	ET2	PCA	X	X	X	
4 Timers	89C51RB+/87C51RB+/83C51RB+	T0	T1	ET2	PCA	Watchdog	X	X	
	89C51RC+/87C51RC+/83C51RC+	T0	T1	ET2	PCA	Watchdog	X	X	
	89C51RD+/87C51RD+/83C51RD+	T0	T1	ET2	PCA	Watchdog	X	X	
XA Family									
4 Timers	P51XAG30/G37/G33	ET0	ET1	ET2		Watchdog	X		X
	P51XAS3	ET0	ET1	ET2	PCA	Watchdog	X		X

SERIAL INTERFACE

Philips microcontrollers offer a broad range of serial communication protocols. These serial interfaces are designed to meet all the different performance and cost requirements of your design.

UART/Enhanced UART

The 80C51 UART is an interface for microcontrollers that provides all the features you need to implement simple serial communication. The 80C51 UART is bi-directional and can simultaneously send and receive data. The 80C51 UART is the serial I/O most often used on 80C51 microcontrollers.

Philips also offers the Enhanced 80C51 UART on selected microcontrollers. Enhanced UART has all the standard UART functions plus framing error detection and automatic address recognition.

I^{2}C

Philips I^{2}C, or Inter-Integrated Circuit, is a 2-wire serial communication with unique start-and-stop conditions, bi-directional communication, full synchronization, and multimaster operation.

I^{2}C is primarily used to communicate between two ICs, or multiple ICs in close proximity (13 feet, at 100K bit/sec). By using Philips I^{2}C Bus Extender, P82B715, the communication is increased by a factor of 10 (excess of 130 feet). This enables I^{2}C to be used in applications between a central control box and remote locations.

The Philips proprietary I^{2}C serial bus protocol is an industry standard found in more than 100 products and licensed to more than 30 IC manufacturers.

CAN bus

The CAN (Control Area Network) bus operates well in noisy, harsh environments, such as in industrial and heavy equipment applications, and in numeric control equipment, such as lathes and mills.

The CAN bus is ideal for designs requiring reliable serial communication or multiplexed wiring. The CAN 2-wire serial bus in our 8XC592 and 8XCE598 microcontrollers is able to transmit data at speeds up to 1 million bits per second and over distances of up to 5,000 meters.

The 8XC592 and 8XCE598 are the first members of a growing family of Philips products that support CAN, and we are continually developing new products to add to this family of powerful serial interface parts.

High-Speed Serial Interface

For very high-speed data transmission between the CPU and other peripheral devices, the 8-bit, 3-wire serial I/O features a maximum speed of 1MHz baud rate. Other features include an interrupt generated whenever a complete byte has been sent or received, baud rate clocks, Schmitt trigger inputs on clock and data-in ports, and software-select of LSB or MSB first.

Selection guide for the 80C51 microcontroller family

SERIAL INTERFACE

Device	UART	Enhanced UART	I^2C Bit	I^2C Byte	CAN Bus	High-speed Serial	Comments/Special Features
87C751/83C751			X				24-Pin Skinny DIP Package and SSOP
87C752/83C752			X				Pulse Width Modulation and SSOP
87C754/83C754		X					8BIT DAC, Comparator, Reference and MUX Input
87C51/80C51/80C31		X					Operation at 2.7V to 5.5V @ 16MHz
80CL51/80CL31	X						Low Voltage/Power (1.8V to 6V)
83CL410/80CL410				X			Low Voltage/Power (1.8V to 6V)
87C451/83C451/80C451	X						7 I/O Ports, Processor Bus Interface
87C550/83C550/80C550	X						Watchdog Timer to reset the microcontroller
83C851/80C851	X						256 Bytes EEPROM, 80C51 Pin-Compatible
83CL580/80CL580	X			X			Watchdog Timer, Pulse Width Modulation, Low Voltage (2.5V to 6V)
89C52/87C52/80C52/80C32		X					Operation at 2.7V to 5.5V @ 16MHz
S87C652/P83C652/P80C652	X						I^2C Serial Bus
87C51FA/83C51FA/80C51FA		X					Industry Standard
89C51RA+/87C51RA+ /83C51RA+/80C51RA+		X					512 RAM & WD
87C575/83C575/80C575		X					See Extended Feature Set below*
87C576/83C576/80C576		X					See Extended Feature Set below*, plus UPI, on-board programming
83C562/80C562	X						Pulse Width Modulation, Watchdog Timer, Capture/Compare Counter/Timer
87C552/83C552/80C552	X			X			Pulse Width Modulation, Watchdog Timer, Capture/Compare Counter/Timer
87C453/83C453/80C453		X					Processor bus interface
89C54/87C54/80C54		X					Operation at 2.7V to 5.5V @ 16MHz
S87C654/P83C654	X			X			I^2C Serial Bus
83CL781	X			X			Low Voltage/Power (1.8V to 6V)
83CL782	X			X			83CL781 optimized for 12MHz @ 3.1V
87C51FB/83C51FB		X					Industry Standard
89C51RB+/87C51RB+/83C51RB+		X					512 RAM & WD
87C524/83C524	X		X				512 RAM, Watchdog Timer
87C592/83C592/80C592	X					X	CAN 2.0A, Watchdog Timer, Capture/ Compare Counter/Timer, Pulse Width Modulation
89C58/87C58/80C58		X					Operation at 2.7V to 5.5V @ 16MHz
87C51FC/83C51FC		X					Industry Standard
89C51RC+/87C51RC+/83C51RC+		X					512 RAM & WD
87C528/83C528/80C528	X		X				Large memory for high level languages, Watchdog Timer
83CE528	X		X				Reduced EMI version of 8XC528
83CE598/80CE598		X				X	CAN bus, Watchdog Timer, Capture/ Compare Counter/Timer, Pulse Width Modulation, reduced EMI
87CE558/83CE558/80CE558	X			X			Watchdog Timer, Pulse Width Modulation, Low EMI
89C51RD+/82C51RD+/83C51RD+		X					1K RAM, Watchdog Timer
XA Family							
P51XAG30/G37/G33		X (2)					2 enhanced UARTs
P51XAS3		X(2)	X				

* Extended Feature Set includes Low V_{CC} Detect, Low Active Reset, Oscillator Fail Detect, Reduced EMI and 4 Analog Comparators.

LOW VOLTAGE OPERATION

Philips offers 80C51 derivative microcontrollers that have the lowest operating voltages available today. In addition, you can select from devices that operate from 5.5V down to 2.7V or 6V down to 1.8V, with such features as idle and power-down modes and a fully static core. This enables the tailoring of performance for lowest power consumption.

Static and 2.7V/1.8V Operation

Philips offers two voltage range choices for low voltage operation, 2.7V to 5.5V and 1.8V to 6V.

The 8XCLXXX family of devices addresses your need for very low power and very low operating voltage microcontrollers. Our 8XCLXXX family of microcontrollers is designed to operate down to 1.8V and have an operating frequency range from DC to 16MHz. The 8XCL580 is designed to operate down to 2V.

Designed with a static core, the expanding family of 2.7V to 5.5V devices include the mask and OTP-EPROM versions, 87/80C51/52/54/58, ROMless 80C31 and 80C32, and FX series. Philips is planning a broad family of OTPs and will be adding mask devices which have 2.7V to 5.5V operation, all designed with a static core. The fully static core enables the oscillator to be stopped for minimal power consumption and then be restarted easily without complicated restart procedures or loss of data.

LOW VOLTAGE (BROAD RANGE) OTP ROM, AND ROMLESS MICROCONTROLLERS

Device	ROM	OTP	RAM	Operating Voltage Range	Frequency	Fully Static	Timers	Special Features
80C31	—	—	128	2.7–5.5V	DC–16Mhz	Yes	2	
80CL31	—	—	128	1.8V–6V	DC–16MHz	Yes	2	80C31 Pin-Compatible, UART
87C51/80C51	4K	4K	128	2.7–5.5V	DC–16MHz	Yes	2	
80CL51	4K	—	128	1.8V–6V	DC–16MHz	Yes	2	80C31 Pin-Compatible, UART
80CL410/83CL410	4K	—	128	1.8V–6V	DC–12MHz	Yes	2	80C31 Pin-Compatible, I^2C
80CL580/83CL580	6K	—	256	2.5V–6V	DC–12MHz	Yes	3+ Watch-dog	4x8-bit A/D, PWM, I^2C, UART
80C32	—	—	256	2.7V–5.5V	DC–16MHz	Yes	3	
87C52/80C52	8K	8K	256	2.7V–5.5V	DC–16MHz	Yes	3	
87C51FA/83C51FA/80C51FA	8K	8K	256	2.7V–5.5V	DC–16MHz	Yes	4	PCA
87C51RA+/83C51RA+/80C51RA+	8K	8K	512	2.7V–5.5V	DC–16MHz	Yes	4	PCA, Hardware Watchdog
87C51RB+/83C51RB+	16K	16K	512	2.7V–5.5V	DC–16MHz	Yes	4	PCA, Hardware Watchdog
87C51RC+/83C51RC+	32K	32K	512	2.7V–5.5V	DC–16MHz	Yes	4	PCA, Hardware Watchdog
87C51RD+/83C51RD+	64K	64K	1024	2.7V–5.5V	DC–16MHz	Yes	4	PCA, Hardware Watchdog
83CL267	12K	—	256	4.5V–5.5V	4MHz–8MHz		3	OSD, 9 PWM Outputs, 4 Channel 4-bit A/D, 9 LED Drivers
83CL268	12K	—	256	4.5V–5.5V	4MHz–8MHz		3	OSD, 9 PWM, Outputs, 4 Channel 4-bit A/D, 9 LED Drivers
87C54/80C54	16K	16K	256	2.7V–5.5V	DC–16MHz	Yes	3	
83CL781	16K	—	256	1.8V–6V	DC–12MHz	Yes	3	UART, I^2C
83CL782	16K	—	256	1.8V–6V	DC–12MHz	Yes	3	Optimized for 12MHz@3.1V, UART, I^2C
87C51FB/83C51FB	16K	16K	256	2.7V–5.5V	DC–16MHz	Yes	4	PCA
83CL167	16K	—	256	4.5–5.5V	4MHz–8MHz		3	OSD, 9 PWM Outputs, 4 Software A/D Inputs, 8 LED Drivers
83CL168	16K	—	256	4.5–5.5V	4MHz–8MHz		3	OSD, 9PWM Ouputs, 4 Software A/D Inputs, 8 LED Drivers
87C51FC/83C51FC	32K	32K	256	2.7V–5.5V	DC–16MHz	Yes	4	PCA
XAFamily								
P51XAG30/G33	32K	—	512	2.7–5.5V	0–30MHz	Yes	3+ Watch-dog	High performance, 16-bit, 80C51 compatible
P51XAS3		32K	1024	2.7V–5.5V	0–30MHz	Yes	4	16Mbyte Address Range

ANALOG FEATURES

Philips offers a variety of 80C51 derivative microcontrollers with on-board, 8-bit and 10-bit A/D converters. Also offered are 80C51 derivatives with 8-bit DAC, Pulse Width Modulated (PWM) output, and those with precision comparators.

Analog-to-Digital Converters

Philips offers 8-bit and 10-bit analog-to-digital (A/D) and digital-to-analog converters (D/A). The A/Ds have multiplexed inputs which enable the selection of up to eight analog sources, and a sample-and-hold front end, making it easier for high frequency measurement. Each input has a high input impedance to reduce loading, resulting in a more accurate measurement.

Pulse Width Modulation (PWM)

The Pulse Width Modulated output allows wave forms to be generated with a very precise frequency and duty cycle. The PWM output offers two features that provide a wide degree of design flexibility: variable frequency and adjustable duty cycle. For the 8-bit PWM, the duty cycle is adjustable from 0% to 100% in 255 steps. These PWM outputs are useful in the design of low cost digital-to-analog converters and, with a comparator, an analog-to-digital converter can be designed.

Comparators

Philips integrated precision comparators provide low-input offset and high flexibility. To reduce the number of external components, Philips has integrated four comparators on the 8XC575 and 8XC576 and one on the 8XC754 which can be used for many functions, such as low cost A/D conversion and cross-over detection.

ANALOG FEATURES

	Analog-to-Digital		Pulse Width Modulation		
Device	Bits	Multiplexed Inputs	# PWM Timers	Frequency Range @12MHz	Number of Comparators
83C749/87C749	8	5	1 8-bit	90Hz–23.5KHz	
83C752/87C752	8	5	1 8-bit	90Hz–23.5KHz	
83C754/87C754			1 8-bit (PCA)	15Hz–11.7KHz	1
80C550/83C550/87C550	8	8	1 8-bit	90Hz–23.5KHz	
80CL580/83CL580	8	4	1 8-bit	90Hz–23.5KHz	
83C145			5 8-bit (PCA)	15Hz–11.7KHz	
80C562/83C562	8	8	2 8-bit	90Hz–23.5KHz	
80C552/83C552/87C552	10	8	2 8-bit	90Hz–23.5KHz	
80C592/83C592/87C592	10	8	2 8-bit	90Hz–23.5KHz	
80CE598/83CE598/87CE598	10	8	2 8-bit	90Hz–23.5KHz	
80CE558/83CE558/89CE558	10	8	2 8-bit	90Hz–23.5KHz	
80C575/83C575/87C575			5 8-bit (PCA)	15Hz–11.7KHz	4
80C576/83C576/87C576	10	6	5 8-bit (PCA)	15Hz–11.7KHz	4
83C845			5 8-bit (PCA)	15Hz–11.7KHz	
80C51FA/83C51FA/87C51FA			5 8-bit (PCA)	15Hz–11.7KHz	
80C51RA+/83C51RA+ /87C51RA+/89C51RA+			5 8-bit (PCA)	15Hz–11.7KHz	
83C51FB/87C51FB			5 8-bit (PCA)	15Hz–11.7KHz	
83C51RB+/87C51RB+/89C51RB+			5 8-bit (PCA)	15Hz–11.7KHz	
83C055/87C055			1 14-bit/8 6-bit (MTV)	60Hz–47KHz	
83CL267/83CL268	4	4	1 14-bit/4 6-bit/ 4 7-bit	90Hz–20KHz@8MHz	
83CL167/83CL168	4	4	1 14-bit/4 6-bit/ 4 7-bit	90Hz–20KHz@8MHz	
83C51FC/87C51FC			5 8-bit (PCA)	15Hz–11.7KHz	
83C51RC+/87C51RC+/89C51RC+			5 8-bit (PCA)	15Hz–11.7KHz	
83C51RD+/87C51RD+/89C51RD+			5 8-bit (PCA)	15Hz–11.7KHz	
P51XAS3	10	8	4 8-bit (PCA)		

PROTECTION CAPABILITIES AND REDUCED EMI/RFI

The Philips family of microcontrollers that contains extremely low levels of EMI and RFI can result in easier FCC certification, lower board design costs, simpler shielding solutions, easier power and ground layout, and reduced cross-talk. In addition, Philips offers protection circuits, such as Watchdog timers, oscillator-failure detection, and low-voltage detection, giving you shorter design cycles, lower system costs, and no external component requirements.

Protection Circuits for High Reliability

Reliability is always a key design concern. This concern is amplified when a circuit is used in life-critical applications such as medical instrumentation, aboard aircraft or on vehicles where circuit lock-up could be disastrous. And, in industrial applications, the failure of one component could shut down an entire production line. Philips offers a wide range of protection circuits that addresses these concerns with a sophisticated set of built-in, hardware-based protection circuits that enhance your system's fault tolerance and improve its reliability: a Watchdog timer, low voltage detection, and oscillator failure detection.

Watchdog Timer

The Philips Watchdog timer is designed to be fail-safe. The Watchdog timer actually resets the microcontroller within a programmable time if the microcontroller enters any potentially fatal processor state. If the user program fails to reload the Watchdog timer within a specified time, the Watchdog circuitry automatically generates a system reset. Protection circuits, such as the Philips Watchdog timer, greatly increase a designer's confidence that the microcontroller will reliably recover from any uncontrolled situations.

Because Philips Watchdog timers are hardware configured, they cannot be corrupted by software-based problems or events that often can occur in microcontroller applications. For added flexibility, the Watchdog timer offers an adjustable time-out period, from 2ms to 512ms at 12MHz oscillator depending on the demands of your applications. (See also Counters/Timers section for more information on the special features of the Philips Watchdog timer.)

Failure Detection Circuits

Other protection features integrated into the Philips products with Watchdog timers are Oscillator Failure Detection (OFD) and Low Voltage Detection. Like the Watchdog timer, these features generate a reset if the oscillator frequency slows below a predefined frequency range or if voltage levels to the part are reduced.

Our integrated protection features mean you don't have to spend extra design time adding components to protect the microcontroller and system. The result is a highly reliable design featuring reduced part count, quicker time-to-market, and reduced cost.

Reduced EMI

Recognizing the challenges in meeting FCC certification, Philips has developed a family of products that has significantly reduced EMI signatures. By changing the internal circuitry on our reduced EMI microcontrollers, radiated noise is reduced by more than 20dB, especially important for frequencies greater than 100MHz where board-level noise reduction is difficult and expensive to achieve.

With our reduced EMI/RFI devices, your design is simpler, your costs are lower, and your power, ground layout, and shielding are simplified. And by using our reduced EMI/RFI products, you're already a step closer to FCC approval.

One example of our family of reduced EMI/RFI devices is the 8XC575, on which many techniques are used to reduce EMI/RFI to a level that is 100 times lower than emissions from a standard 80C51 microcontroller. And everything we've learned from the 8XC575 has been incorporated into the growing list of reduced EMI/RFI products.

ADDITIONAL FEATURES

To make your designs easier, Philips has several specialty microcontroller products, each offered in a range of packages.

On-Screen Display (OSD)

Philips OSD, featured on our 8XC055, 83C145 and 83C845 and 83C366/566/766, allows you to put text over video easily, and with a surprising degree of color and shading flexibility.

Philips OSD offers a user-defined character set, including font types and sizes, starting position, character matrix, foreground and background colors and shadowing, as well as character blinking ratio and vertical jitter canceling. With Philips OSD you can put as many as 128 characters on screen at a time. Both the number of rows and the number of characters per row are user-defined.

Philips OSD microcontroller capability offers all the design flexibility you need to custom tailor your system. In addition, Philips combines that flexibility with hard-ware that makes OSD simple to include in many different designs.

Universal Peripheral Interface (UPI)

For products that require interface to the ISA bus in personal computer and other host CPU buses, the 8XC451, 87C453, and 8XC576 are ideal because they have the UPI built in. On-board UPI means it's easier for you to design serial I/O controllers, servo controllers, and keyboard scanners.

Smart Card

The Philips Smart Card microcontroller provides features for a range of applications and devices, such as credit cards, electronic keys, medical records, and identification.

Typical access control applications that use Smart Cards are satellite TV networks, mobile phones, computers, pay TV, auto key cards, and electronic keys. Other Smart Card applications are medical records, patient passports, and cash cards. In production markets, Smart Card is used in flow control, tool handling, maintenance, and process control.

Philips' Smart Card microcontroller, the 83C852, features 6K bytes of ROM, 256 bytes of RAM, and 2K bytes of EEPROM. The 83C852 has a cryptographic calculation unit (CCU) to enhance data security, and it comes in a credit card style package or can be purchased in die form.

Also available are the P83C855, P83C858 and P83C864 designed specifically for smart card applications.

PROTECTION CAPABILITIES AND REDUCED EMI/RFI

Device	Watchdog	Oscillator Failure Detection	Low Voltage Detection	Reduced EMI	Comments/Special Features
80C51/80C31/87C51				X	Operation at 2.7V to 5.5V @ 16MHz
80C550/83C550/87C550	X				8 Channel 8-bit Analog-to-Digital Converter
80CL580/83CL580	X			X	4 Channel 8-bit Analog-to-Digital Converter, Pulse Width Modulation Output, Low Power/Volt (2.5V to 6V)
80C52/80C32/87C52				X	Operation at 2.7V to 5.5V @ 16MHz
83C145				X	OSD, 9 PWM Ouputs, 3 Software Analog-to-Digital Converter Inputs
83C51FA/80C51FA/87C51FA	PCA			X	Industry Standard
80C575/83C575/87C575	X	X	X	X	High Reliability, Low V_{CC}/Osc Fail Detect, Analog Comparators, Programmable Counter Array
80C576/83C576/87C576	X	X	X	X	Same as 8XC575 plus UPI and 10-bit Analog-to-Digital Converter
80C562/83C562	X				8 Channel 8-bit Analog-to-Digital Converter, 2 Pulse Width Modulation Outputs, Capture/Compare Timer
80C552/83C552/87C552	X				8 Channel 10-bit Analog-to-Digital Converter, 2 Pulse Width Modulation Outputs, Capture/Compare Timer
83C845				X	OSD, 9 PWM Ouputs, 3 Software Analog-to-Digital Converter Inputs
83C055/87C055				X	OSD, 9 PWM Ouputs, 3 Software Analog-to-Digital Converter Inputs
80C54/87C54				X	Operation at 2.7V to 5.5V @ 16MHz
83C51FB/87C51FB	PCA			X	Industry Standard
83C524/87C524	X				512 RAM
80C58/87C58				X	Operation at 2.7V to 5.5V @ 16MHz
83C51FC/87C51FC	PCA			X	Industry Standard
80C528/83C528/87C528	X				Large memory for high-level languages
80CE598/83CE598/87CE598	X			X	CAN Bus, 8x10-bit Analog-to-Digital Converter, Low Electro-Magnetic Interference, 2 Pulse Width Modulation Outputs, Enhanced UART
80CE558/83CE558/89CE558	X			X	Low EMI, 8 Channel 10-bit Analog-to-Digital Converter, 2 Pulse Width Modulation Outputs, Capture/Compare
89C51RA+/87C51RA+/ 83C51RA+/80C51RA+	X			X	Double RAM/512 bytes & WD
89C51RB+/87C51RB+/83C51RB+	X			X	Double RAM/512 bytes & WD
89C51RC+/87C51RC+/83C51RC+	X			X	Double RAM/512 bytes & WD
89C51RD+/87C51RD+/83C51RD+	X			X	1KRAM &WD

80C51 microcontroller family features guide

Memory from 1K to 8K

Prefix	Part Number ROM/ROMless/ OTP/Flash	Memory			New and Improved (Note 6)	Counter				I/O Pins	Serial Interfaces	Comments/ Special Features
		ROM	EPROM	RAM		#	PWM	PCA	WD			
P	83C750	1K		64		1	N	N	N	19	–	Lowest cost, 1 (16-bit) Timer, SSOP
P	87C750		1K	64		1	N	N	N	19	–	Lowest cost, 1 (16-bit) Timer, SSOP
P	83C748	2K		64		2	N	N	N	19	–	’751 w/o I^2C, 1 (16-bit) Timer, SSOP
P	87C748		2K	64		2	N	N	N	19	–	’751 w/o I^2C, 1 (16-bit) Timer, SSOP
S	83C751	2K		64		1	N	N	N	19	I^2C (bit)	1 (16-bit) Timer, SSOP
S	87C751		2K	64		1	N	N	N	19	I^2C (bit)	1 (16-bit) Timer, SSOP
P	83C749	2K		64		2	Y	N	N	21	–	’752 w/o I^2C, 1 (16-bit) Timer, SSOP
P	87C749		2K	64		2	Y	N	N	21	–	’752 w/o I^2C, 1 (16-bit) Timer, SSOP
S	83C752	2K		64		1	Y	N	N	21	I^2C (bit)	1 (16-bit) Timer, SSOP
S	87C752		2K	64		1	Y	N	N	21	I^2C (bit)	1 (16-bit) Timer, SSOP
P	80C51/80C31	4K		128	Y	2	N	N	N	32	UART	CMOS
P	87C51		4K	128	Y	2	N	N	N	32	UART	CMOS
P	80CL51/80CL31	4K		128		2	N	N	N	32	UART	Low voltage (1.8V–6V), Low power
P	83C434	4K		128								LCD driver
P	83CL410/80CL410	4K		128		2	N	N	N	32	I^2C	Low voltage (1.8V–6V), Low power
SC	83C451/80C451	4K		128		2	N	N	N	56	UART	Extended I/O, Processor bus interface
SC	87C451		4K	128		2	N	N	N	56	UART	Extended I/O, Processor bus interface
P	83C550/80C550	4K		128		2	N	N	Y	32	UART	8 channel 8-bit A/D w/Hdw WD
P	87C550		4K	128		2	N	N	Y	32	UART	8 channel 8-bit A/D w/Hdw WD
P	83C851/80C851	4K		128		2	N	N	N	32	UART	256B EEPROM, 80C51 pin-compatible
P	83C754	4K		256		3	Y	Y	N	11	UART	8-bit DAC, 3-input mux comparator, Ref V Out
P	87C754		4K	256		3	Y	Y	N	11	UART	(see above)
P	83C852	6K		256		2	N	N	N	16	–	Smartcard controller with 2K EEPROM (Data, Code) Cryptographic Calc Unit
P	83CL580/80CL580	6K		256		3	Y	Y	Y	40	UART, I^2C	4 channel 8-bit A/D, w/Hdw WD, low voltage (2.5V–6V), low power
P	80C52/80C32	8K		256	Y	3	N	N	N	32	UART	80C51 pin-compatible
P	87C52		8K	256	Y	3	N	N	N	32	UART	80C51 pin-compatible
P	83C51RA+/80C51RA+	8K		512	Y	4	Y	Y	Y	32	UART	w/Hdw WD, 2.7–5.5V versions
P	89C51RA+/87C51RA+		8K	512	Y	4	Y	Y	Y	32		(see above) (FLASH–5V only)
P	83C652/80C652	8K		256		2	N	N	N	32	UART, I^2C	80C51 pin-compatible
S	87C652		8K	256		2	N	N	N	32	UART, I^2C	80C51 pin-compatible
P	83C453/80C453	8K		256		2	N	N	N	56	UART	Extended I/O, processor bus interface
P	87C453		8K	256		2	N	N	N	56	UART	Extended I/O, processor bus interface
P	83C51FA/80C51FA	8K		256	Y	4	Y	Y	Y	32	UART	Enhanced UART, 3 timers + PCA
P	87C51FA		8K	256	Y	4	Y	Y	Y	32	UART	Enhanced UART, 3 timers + PCA
P	83C575/80C575	8K		256		4	Y	Y	Y	32	UART	w/Hdw WD, low voltage detect, osc fail detect, analog comparators, PCA
P	87C575		8K	256		4	Y	Y	Y	32	UART	(see above)
P	83C576	8K		256		4	Y	Y	Y	32	UART	Same as 8xC575 plus UPI and 10-bit A/D
P	87C576	8K		256		4	Y	Y	Y	32	UART	(see above)
P	83C845	8K		256		2	Y	N	N	28	–	On-screen display, 9 PWM outputs, 3 software A/D inputs
P	83C880	8K		512								DDC interface for monitors, auto sync detection and sync processor
PCx	83C562/80C562	8K		256		3	Y	N	Y	48	UART	8 channel 8-bit A/D, 2 PWM outputs, Capture/Compare timer, w/Hdw WD
PCx	83C552/80C552	8K		256		3	Y	N	Y	48	UART, I^2C	8 channel 10-bit A/D, 2 PWM outputs, Capture/Compare timer, w/Hdw WD
S	87C552		8K	256		3	Y	N	Y	48	UART, I^2C	(see above)
P	83C834	8K		256								LCD driver
P	83CL883	8K	8K	256		3	N	N	Y	19	UART	1.8–3.6V operation, low voltage detection
P	83CL884	8K	8K	256		3	N	N	Y	18	UART	1.8–3.6V operation, low voltage detection

NOTES:
Part number prefixes are noted in the first column.
All combinations of part type, speed, temperature and package may not be available.

Memory from 1K to 8K (continued)

Part Number ROM/ROMless/ OTP/Flash		A/D		External Interrupt	Program Security ?	Clock Freq. (MHz)	Temperature Range (°C)			Package		
		Bits	Channels				0 to +70	–40 to +85	–55 to +125	PDIP	PLCC	PQFP/SSOP
83C750	S			2	N	3.5 to 40	X	X		N24	A28	DB24 (0–70F)
87C750	S			2	Y	3.5 to 40	X	X		N24	A28	DB24 (0–70F)
83C748	S			2	N	3.5 to 16	X	X		N24	A28	DB24 (0–70F)
87C748	S			2	Y	3.5 to 16	X	X		N24	A28	DB24 (0–70F)
83C751	S			2	N	3.5 to 16	X	X		N24	A28	DB24 (0–70F)
87C751	S			2	Y	3.5 to 16	X	X		N24	A28	DB24 (0–70F)
83C749	S	8	5	2	N	3.5 to 16	X	X		N28	A28	DB28 (0–70F)
87C749	S	8	5	2	Y	3.5 to 16	X	X		N28	A28	DB28 (0–70F)
83C752	S	8	5	2	N	3.5 to 16	X	X	X	N28	A28	DB28 (0–70F)
87C752	S	8	5	2	Y	3.5 to 16	X	X	X	N28	A28	DB28 (0–70F)
80C51/80C31	S			2	Y	0 to 33	X	X		N40	A44	B44 (5)
87C51	S			2	Y	0 to 33	X	X		N40	A44	B44 (5)
80CL51/80CL31	Z			10	N	0 to 16 (1)		X		N40 (2)		B44
83C434	T					12MHz		X		NB42		B44
83CL410/80CL410	Z			10	N	0 to 12 (1)		X		N40 (2)		B44
83C451/80C451	S			2	N	3.5 to 16	X	X		N64 (4)	A68	
87C451	S			2	Y	3.5 to 16	X	X		N64 (4)	A68	
83C550/80C550	S	8	8	2	Y	3.5 to 16	X	X		N40	A44	
87C550	S	8	8	2	Y	3.5 to 16	X	X		N40	A44	
83C851/80C851	H			2	Y	1.2 to 16	X	X		N40	A44	B44
83C754	S			2	Y	3.5 to 16	X					
87C754	S			2	Y	3.5 to 16	X					DB28
83C852	H			1	Y	1 to 12	X			SO28 or die		
83CL580/ 80CL580	Z	8	4	9	N	0 to 12 (1)		X		(3)		B64
80C52/80C32	S			2	Y	0 to 33	X	X		N40	A44	B44 (5)
87C52	S			2	Y	0 to 33	X	X	X	N40	A44	B44 (5)
83C51RA+/80C51RA+	S			2	Y	0 to 33	X	X		N40	A44	B44
89C51RA+/87C51RA+	S				Y	0 to 33	X	X		N40	A44	B44
83C652/80C652	H			2	Y	3.5 to 24	X	X	–40 to +125	N40	A44	B44
87C652	S			2	Y	3.5 to 16	X	X	X	N40	A44	
83C453/80C453	S			2	N	3.5 to 16	X	X			A68	
87C453	S			2	Y	3.5 to 16	X	X			A68	
83C51FA/80C51FA	S			2	Y	0 to 33	X	X		N40	A44	B44
87C51FA	S			2	Y	0 to 33	X	X		N40	A44	B44
83C575/80C575	S			2	Y	4 to 16	X	X	X	N40	A44	B44
87C575	S			2	Y	4 to 16	X	X	X	N40	A44	B44
83C576	S	10	6	2	Y	6 to 16	X	X	X	N40	A44	B44
87C576	S	10	6	2	Y	6 to 16	X	X	X	N40	A44	B44
83C845	T			2	N	3.5 to 20	X			NB42		
83C880				2								
83C562/80C562	H	8	8	2	N	3.5 to 16	X	X	–40 to +125		A68	B80
83C552/80C552	H	10	8	2	N	3.5 to 30	X	X	–40 to +125		A68	B80
87C552	S	10	8	2	Y	3.5 to 16	X				A68	
83C834	T					16		X		NB42		B44
83CL883	Z			8		3.58	–25 to +70					SO28
83CL884	Z			8		3.58	–25 to +70					SO28

NOTES:
Production Centers are indicated in the second column:
H = Hamburg; S = Sunnyvale, T = Taiwan, Z = Zurich
All combinatiuons of part type, speed, temperature and package may not be available.

1. Oscillator options start from 32kHz.
2. Also available in VSO40 package.
3. Also available in VSO56 package.
4. Not recommended for new design.
5. Package available up to 16MHz only.
6. New and Improved devices operate from 2.7V–5.5V @ 16MHz. Static Core, 33MHz operation, Dual Data Pointers, and more.

80C51 microcontroller family features guide

Memory from 12K to 64K

Prefix	Part Number ROM/ROMless/ OTP/Flash	Memory			New and Improved (Note 6)	Counter				I/O Pins	Serial Interfaces	Comments/ Special Features
		ROM	EPROM/ FLASH	RAM		#	PWM	PCA	WD			
P	83C145	12K		256		2	Y	N	N	28	–	On-Screen Display, 9 PWM outputs, 3 software A/D inputs
P	83CL887/87CL887	12K	12K	256		3	N	N	Y	18	UART	1.8V–3.6V operation, low voltage detection
P	83C055	16K		256		2	Y	N	N	28	–	On-Screen Display, 9 PWM outputs 3 software A/D inputs
P	87C055		16K	256		2	Y	N	N	28	–	(see above)
P	83C180	16K	16K	512								DDC interface for monitors, auto sync detection and sync processor
P	80C54	16K		256	Y	3	N	N	N	32	UART	Standard; 80C51 compatible
P	87C54		16K	256	Y	3	N	N	N	32	UART	Standard; 87C51 compatible
P	89C138		16K	256		3	N	N	N	32	UART	Reduced EMI, Hdw. Watchdog timer
P	83C654/80C654	16K		256		2	N	N	N	32	UART, I^{2}C	80C51 pin-compatible
P	87C654		16K	256		2	N	N	N	32	UART, I^{2}C	80C51 pin-compatible
P	83CL781	16K		256		3	N	N	N	32	UART, I^{2}C	Low voltage (1.8V–6V), low power
P	83CL782	16K		256		3	N	N	N	32	UART, I^{2}C	83CL781 optimized 12MHz@3.1V
P	83C51FB	16K		256	Y	4	Y	Y	Y	32	UART	Enhanced UART, 3 timers + PCA
P	87C51FB		16K	256	Y	4	Y	Y	Y	32	UART	Enhanced UART, 3 timers + PCA
P	83C524	16K		512		3	N	N	Y	32	UART, I^{2}C-bit	512 RAM
P	87C524		16K	512		3	N	N	Y	32	UART, I^{2}C-bit	512 RAM
P	83C51RB+	16K		512	Y	4	Y	Y	Y	32	UART	Extended RAM (512 bytes), 2.7V–5.5V versions
P	89C51RB+/87C51RB+		16K	512		4	Y	Y	Y	32	UART	(see above) (FLASH–5V only)
P	83CL886/87CL886	16K	16K	512		3		N	Y	18	UART	1.8V–3.6V operation, low voltage detection
P	83C592/80C592	16K		512	Y	3	N	N	Y	48	UART, CAN	CAN bus controller, 8×10-bit A/D, 2 PWM outputs, Capture/Compare timer
P	83C280	24K		512								DDC interface for monitors, auto sync detection and sync processor
P	83C380	32K		512			Y					DDC interface for monitors, auto sync detection and sync processor
P	80C58	32K		256	Y	3	N	N	N	32	UART	Standard; 80C51 compatible
P	87C58		32K	256	Y	3	N	N	N	32	UART	Standard; 87C51 compatible
P	83C51FC	32K		256	Y	4	Y	Y	Y	32	UART	Enhanced UART, 3 timers + PCA
P	87C51FC		32K	256	Y	4	Y	Y	Y	32	UART	Enhanced UART, 3 timers + PCA
P	83C528/80C528	32K		512		3	N	N	Y	32	UART, I^{2}C-bit	Large memory for high level languages
P	87C528		32K	512		3	N	N	Y	32	UART, I^{2}C-bit	Large memory for high level languages
P	83CE528	32K		512		3	N	N	Y	32	UART, I^{2}C-bit	8XC528 with Reduced EMI
P	83C51RC+	32K		512	Y	4	Y	Y	Y	32	UART	Extended RAM (512 bytes), 2.7V–5.5V versions
P	89C51RC+/87C51RC+		32K	512	Y	4	Y	Y	Y	32	UART	(see above) (FLASH–5V only)
P	89C238		32K	256		3	N	N	Y	32	UART	Reduced EMI, Hdw. Watchdog timer
P	83CE598/80CE598	32K		512		3	Y	N	Y	48	UART, CAN	CAN bus controller, 8×10-bit A/D, 2 PWM outputs, WD, T2, Reduced EMI
P	83CE558/80CE558/ 89CE558	32K	32K	1024		3	Y	N	Y	48	UART, I^{2}C	Low EMI, 8 Channel 10-bit A/D, 2 PWM outputs, Capture/Compare Timer
P	89C738	64K		512		3	N	N	N	32	UART	Open Collector outputs
P	83C51RD+	64K		1024	Y	4	Y	Y	Y	32	UART	Extended RAM (1024 bytes), 2.7V–5.5V versions
P	89C51RD+/87C51RD+		64K	1024	Y	4	Y	Y	Y	32	UART	(see above) (FLASH–5V only)
P	87CL881		64K	2048		3	Y	N	Y	32	UART	1.8V–3.6V operaiton, low voltage detection
XA Family												
P	51XAG30/G37/G33	32K	32K	512		3	N	N	Y	32	2 UARTs	2.7V–5.5V operation, 16-bit XA core, compatible with 80C51
P	51XAS3		32K	1024		4	Y	Y	Y	50	2 UARTs, I^{2}C	16M byte address range, 2.7V–5.5V operation

NOTES:
Part number prefixes are noted in the first column.
All combinations of part type, speed, temperature and package may not be available.

80C51 microcontroller family features guide

Memory from 12K to 64K (continued)

Part Number ROM/ROMless/ OTP/Flash		A/D		External Interrupt	Program Security?	Clock Freq. (MHz)	Temperature Range (°C)			Package		
		Bits	Channels				0 to +70	–40 to +85	–55 to +125	PDIP	PLCC	PQFP/ Other
83C145	T			2	N	3.5 to 20	X			NB42		
83CL887/ 87CL887	T			8	N	3.58	–25 to +70					SO28
83C055	T			2	N	3.5 to 20	X			NB42		
87C055	T			2	N	3.5 to 20	X			NB42		
83C180	T					16		–25 to +85		NB42		
80C54	S			2	Y	0 to 33	X	X		N40	A44	B44
87C54	S			2	Y	0 to 33	X	X		N40	A44	B44
89C138	T			3	Y	3.5 to 40	X			NB42	A44	B44
83C654/80C654	H			2	Y	3.5 to 24	X	X	–40 to +125	R42, N40	A44	B44
87C654	S			2	Y	3.5 to 20	X	X		N40	A44	B44
83CL781	Z			10	N	0 to 12 (1)		X		N40		B44
83CL782	Z			10	N	0 to 12 (1)			–25 to +55	N40		B44
83C51FB	S			2	Y	0 to 33	X	X		N40	A44	B44
87C51FB	S			2	Y	0 to 33	X	X		N40	A44	B44
83C524	H			2	Y	3.5 to 24	X	X		N40	A44	B44
87C524	S			2	Y	3.5 to 16	X	X		N40	A44	B44
83C51RB+	S			2	Y	0 to 33	X	X		N40	A44	B44
89C51RB+/87C51RB+	S			2	Y	0 to 33	X	X		N40	A44	B44
83CL886/87CL886	Z			8								
83C592/80C592	H	10	8	6	Y	1.2 to 16		X	–40 to +125		A68	
83C280	T					16		–25 to +85		NB42		
83C380	T					16		–25 to +85		NB42		
80C58	S			2	Y	0 to 33	X	X		N40	A44	B44
87C58	S			2	Y	0 to 33	X	X		N40	A44	B44
83C51FC	S			2	Y	0 to 33	X	X		N40	A44	B44
87C51FC	S			2	Y	0 to 33	X	X		N40	A44	B44
83C528/80C528	H			2	Y	3.5 to 16	X	X	–40 to +125	N40, R42	A44	B44
87C528	S			2	Y	3.5 to 16	X	X		N40	A44	B44
83CE528	H			2	Y	3.5 to 16		X			A44	B44
83C51RC+	S			2	Y	0 to 33	X	X		N40	A44	B44
89C51RC+/87C51RC+	S			2	Y	0 to 33	X			N40	A44	B44
89C238	T			2	Y	3.5 to 40		X		NB42	A44	B44
83CE598/80CE598	H	10	8	6	Y	1.2 to 16	X	X	–40 to +125			B80
83CE558/80CE558 89CE558	T	10	8	2	Y	1.2 to 16	X	X	–40 to +125 Except 89CE558			B80
89C738	T			2	N	3.5 to 16	X			N40	A44	B44
83C51RD+	S			2	Y	0 to 33	X	X		N40	A44	B44
89C51RD+/87C51RD+	S			2	Y	0 to 33	X	X		N40	A44	B44
87CL881	T			8		1 to 10	–25 to +70					BD44
XA Family												
51XAG30/G37/G33	S				Y	30	X	X			A44	BD44
51XAS3	S	10	8		Y	30	X	X			A68	B80

NOTES:
Production Centers are indicated in the second column:
H = Hamburg; S = Sunnyvale, T = Taiwan, Z = Zurich
All combinatiuons of part type, speed, temperature and package may not be available.

1. Oscillator options start from 32kHz.
2. Also available in VSO40 package.
3. Also available in VSO56 package.
4. Not recommended for new design.
5. Package available up to 16MHz only.
6. New and Improved devices operate from 2.7V–5.5V @ 16MHz. Static Core, 33MHz operation, Dual Data Pointers, and more.

8051 microcontroller cross-reference guide

	INTEL	SIEMENS	OKI	MATRA/HARRIS	PHILIPS SEMICONDUCTORS
CMOS	80C31BH	SAB 80C31	MSM80C31	80C31	P80C31
	80C31BH-1			80C31-1	P80C31
	80C31BH-2		MSM80C31	80C31	P80C31
	80C51BH	SAB 80C51	MSM80C51	80C51	P80C51
	80C51BH-1			80C51-1	P80C51
	80C51BH-2		MSM80C51	80C51	P80C51
	87C51				P87C51
	87C51-1				P87C51
	87C51-2				P87C51
	80C32	SAB80C32			P80C32
	80C32-1			80C32-25	P80C32
	80C52	SAB80C52			P80C52
	80C52-1			80C52-25	P80C52
	80C54				80C54
	83C54				83C54
	87C54				87C54
	80C58				80C58
	83C58				83C58
	87C58				87C58
CMOS	83C51FA				P83C51FA
	87C51FA				P87C51FA
	83C51FB				P83C51FB
	87C51FB				P87C51FB
	83C51FC				P83C51FC
	87C51FC				P87C51FC

NOTES:

1. 80XXAHL = 80XX with low power standby pin; H = HMOS.

Low power / low voltage microcontroller family

80C51 LOW POWER FAMILY

Type	Available	ROM	RAM	I/O	I^2C	UART	Features	Package
80CL51	Yes	4k	128	32	No	Yes	Low Voltage 80C51	40-Pin Dual In-Line 40-Pin Very Small Outline 44-Pin Quad Flat Pack
83CL410	Yes	4k	128	32	Yes	No	80CL51 with I^2C-bus	40-Pin Dual In-Line 40-Pin Very Small Outline 44-Pin Quad Flat Pack
80CL410	Yes	–	128	32	Yes	No	80CL51 with I^2C-bus	40-Pin Dual In-Line 40-Pin Very Small Outline 44-Pin Quad Flat Pack
83CL580	Yes	6k	256	40	Yes	Yes	ADC, PWM, Watchdog, T2	50–Pin Very Small Outline 64-Pin Quad Flat Pack
80CL580	Yes	–	256	40	Yes	Yes	ADC, PWM, Watchdog, T2	50–Pin Very Small Outline 64-Pin Quad Flat Pack
83CL781	Yes	16k	256	32	Yes	Yes	Low voltage 83C654, T2	40-Pin Dual In-Line 44-Pin Quad Flat Pack
83CL782	Yes	16k	256	32	Yes	Yes	Fast 83CL781: 12MHz/3V	44-Pin Quad Flat Pack
85CL000	Yes	–	256	32	Yes	Yes	For SW development	Piggyback
85CL580	Yes	–	256	40	Yes	Yes	For SW development	Piggyback
85CL782	Yes	–	256	32	Yes	Yes	For SW development	Piggyback

LOW VOLTAGE DEVICES

Type	Available	ROM/ EPROM	RAM	I/O	I^2C	UART	Features	Package
8XC51FA	Yes	8k	256	32	No	Yes	PCA, Enhanced UART, 2.7V to 5.5V @16MHz	40-Pin Dual In-Line 44-Pin PLCC 44-Pin Quad Flat Pack
8XC51FB	Yes	16k	256	32	No	Yes	PCA, Enhanced UART, 2.7V TO 5.5V @16MHz	40-Pin Dual In-Line 44-Pin PLCC 44-Pin Quad Flat Pack
8XC51FC	Yes	32k	256	32	No	Yes	PCA, Enhanced UART, 2.7V TO 5.5V @16MHz	40-Pin Dual In-Line 44-Pin PLCC 44-Pin Quad Flat Pack
8XC51RA+	Yes	8k	512	32	No	Yes	PCA, Enhanced UART, Watchdog	40-Pin Dual In-Line 44-Pin PLCC 44-Pin Quad Flat Pack
8XC51	Yes	4k	128	32	No	Yes	2.7 to 5.5V@16MHz, Enhanced UART	40-Pin Dual In-Line 44-Pin PLCC 44-Pin Quad Flat Pack
8XC52	Yes	8k	256	32	No	Yes	2.7 to 5.5V@16MHz, Enhanced UART	40-Pin Dual In-Line 44-Pin PLCC 44-Pin Quad Flat Pack
8XC54	Yes	16k	256	32	No	Yes	2.7 to 5.5V@16MHz, Enhanced UART	40-Pin Dual In-Line 44-Pin PLCC 44-Pin Quad Flat Pack
8XC58	Yes	32k	256	32	No	Yes	2.7 to 5.5V@16MHz, Enhanced UART	40-Pin Dual In-Line 44-Pin PLCC 44-Pin Quad Flat Pack
8XC51RB+	Yes	16k	512	32	No	Yes	2.5 to 5.5V@16MHz, Enhanced UART, Watchdog	40-Pin Dual In-Line 44-Pin PLCC 44-Pin Quad Flat Pack
8XC51RC+	Yes	32k	512	32	No	Yes	2.5 to 5.5V@16MHz, Enhanced UART, Watchdog	40-Pin Dual In-Line 44-Pin PLCC 44-Pin Quad Flat Pack

LOW VOLTAGE DEVICES (continued)

Type	Available	ROM/E PROM	RAM	I/O	I^{2}C	UART	Features	Package
8X51RD+	YES	64K	1K	32	No	Yes	2.5 to 5.5V, Enhanced UART, Watchdog	
P83CL883T/ P87CL883T	Q4'97	8K	256	19/18	Yes	Yes	MSK modem, in-circuit programming, fast I^{2}C, for CT0 applications, DTMF, 2.5–3.6V	28-pin SO
P83CL884T/ P87CL884T	Q4'97	8K	256	19/18	Yes	Yes	MSK modem, in-circuit programming, fast I^{2}C, EEPROM, for CT0 applications, DTMF, 2.7–3.6V	28-pin SO, compatible with '883
P83CL886/ P87CL886	Q4'97	16K	512	19/18	Yes	Yes	MSK modem, in-circuit programming, fast I^{2}C, DTMF, for CT0 applications, DTMF, 2.7–3.6V	28-pin SO, compatible with '883
P83CL887/ P87CL887	Q4'97	12K	512	19/18	Yes	Yes	MSK modem, in-circuit programming, fast I^{2}C, DTMF, for CT0 applications, DTMF, 2.7–3.6V	28-pin SO, compatible with '883
P87CL881	Q4'97	63K	2K	32	Yes	Yes	In-circuit programming, fast I^{2}C, UART, for Pager applications, DTMF, 2.7–3.6V	44-pin Quad Flat Pack

CMOS and NMOS 8-bit microcontroller family

8400 FAMILY CMOS

TYPE	ROM	RAM	SPEED (MHz)	PACKAGE	FUNCTIONS	REMARKS	PROBE SDS	EMULATOR
84C81A	8k	256	16	DIL28/SO28	20 I/O lines 8-bit timer Byte I^2C			OM5501
84C12A	1k	64	16	DIL20/SO20	13 I/O lines 8-bit timer			OM5501
84C00B 84C00T	0 0	256 256	10 10	piggyback VSO-56	20 I/O lines 8-bit timer Byte I^2C	Piggyback ROMless		
84C122A 84C122B 84C422A 84C422B 84C822A 84C822B 84C822C	1k 4K 8K	32 32 32	10	A: SO20 B: SO24 C: SO28	Controller for remote control A: 12 I/O B: 16 I/O C: 20 I/O		OM4830	
84C440 84C441 84C443 84C444 84C640 84C641 84C643 84C644 84C840 84C841 84C843 84C844	4k 4k 4k 4k 6k 6k 6k 6k 8k 8k 8k 8k	128 128 128 128 128 128 128 128 192 192 192 192	10 10 10 10 10 10 10 10 10 10 10 10	DIP42 shrunk	RC: 29 I/O lines LC: 28 I/O lines 8-bit timer 1 14-bit PWM 5 6-bit PWM 3-bit ADC OSD 2L-16	I^2C, RC I^2C, LC RC LC I^2C, RC I^2C, LC RC LC I^2C, RC I^2C, LC RC LC	OM1074	For emulation of LC versions, use OM1074 + adapter_3 + 2 adapter_5 Baud for LCDS OM4831
84C646 84C846	6k 8k	192 192	10 10	DIP42 shrunk	30 I/O lines DOS clock = PLL 8 bit timer 1-14 bit PWM 4-6 bit PWM 4-7 bit PWM 3-4 bit ADC DOS: 64 disp. RAM 62 char. fonts Char. blinking Shadow modes 8 foreground colors/char. 8 background colors/word DOS: clock: 8 . . 20MHz	I^2C, RC I^2C, RC	OM4829 + OM4832	OM4833 for LCD584

CMOS and NMOS 8-bit microcontroller family

8400 FAMILY NMOS

TYPE	ROM	RAM	SPEED (MHz)	PACKAGE	FUNCTIONS	REMARKS	EMULATOR TOOLS	REMARKS
8411	1k	64	6	DIL28/SO28	20 I/O lines			OM1025
8421	2k	64	6	DIL28/SO28	8-bit timer			(LCDS) +
8441	4k	128	6	DIL28/SO28	Byte I^{2}C			OM1026
8461	6k	128	6	DIL28/SO28				
8422	2k	64	6	DIL20	13 I/O lines			
8442	4k	128	6	DIL20	8-bit timer Bit I^{2}C			
8401B	0	128	6	28-pin		Piggyback for 84X1		

3300 FAMILY CMOS

TYPE	ROM	RAM	SPEED (MHz)	PACKAGE	FUNCTIONS	REMARKS	emulator
3349A	4k	224	1–16	DIL28/SO28	20 I/O lines 8-bit timer DTMF generator		OM5501/2
3350A	8k	128	1–16	QFP44 LQFP32	34 I/O lines 8-bit timer DTMF generator 256 bytes EEPROM		OM5501/2
3351A/C	2k	64	1–16	DIL28/SO28 LQFP32	20 I/O lines 8-bit timer DTMF generator 128 bytes EEPROM		OM5501/2
3352A/C	4k	128	1–16	DIL28/SO28 LQFP32	20 I/O lines 8-bit timer DTMF generator 128 byte EEPROM		OM5501/2
3353A/C	6k	128	1–16	DIL28/SO28 LQFP32	20 I/O lines 8-bit timer DTMF generator Ringer out 128 bytes EEPROM		OM5501/2
3354A	8k	256	1–16	QFP44	36 I/O lines 2x 8-bit timer DTMF generator Ringer out 256 bytes EEPROM		OM5501/2
3355A	8	128	1–16	DIL28 SO28 LQFP32	20 I/O lines 128 bytes EEPROM DTMF, 2x 8-bit counters		OM5501/2
3356A	8	128	1–16	DIL28 SO28 LQFP32	20 I/O lines 128 bytes EEPROM DTMF, 2x 8-bit counters		OM5501/2
3357A	6	128	1–16	DIL28 SO28 LQFP32	20 I/O lines 128 bytes EEPROM DTMF, 2x 8-bit counters		OM5501/2

3300 FAMILY CMOS (continued)

TYPE	ROM	RAM	SPEED (MHz)	PACKAGE	FUNCTIONS	REMARKS	emulator
3359A	2	64	1–16	DIL28 SO28 LQFP32	20 I/O lines 128 bytes EEPROM DTMF, 2x 8-bit counters		OM5501/2
3745A	4.5k (OTP)	256	1–16	SO28 DIL28 LQFP32	16 I/O lines, 8-bit timer, RTC, V_{DD} 1.8V–6V 2 Programmable counters with 2 inputs		OM5501/2
3755A/3756A	8k(OTP)	128	1–16	DIL28/SO28	20 I/O lines 2x 8-bit timer DTMF generator Melody output 128 bytes EEPROM		OM5501/2
3354B					Piggyback for 3354A		
All 33xx +37xx							OM1025 + OM5024

NOTE: Further information on these products can be found in Databook IC03 or on our internet page http://www.semiconductors.philips.com

CMOS 16-bit microcontroller family

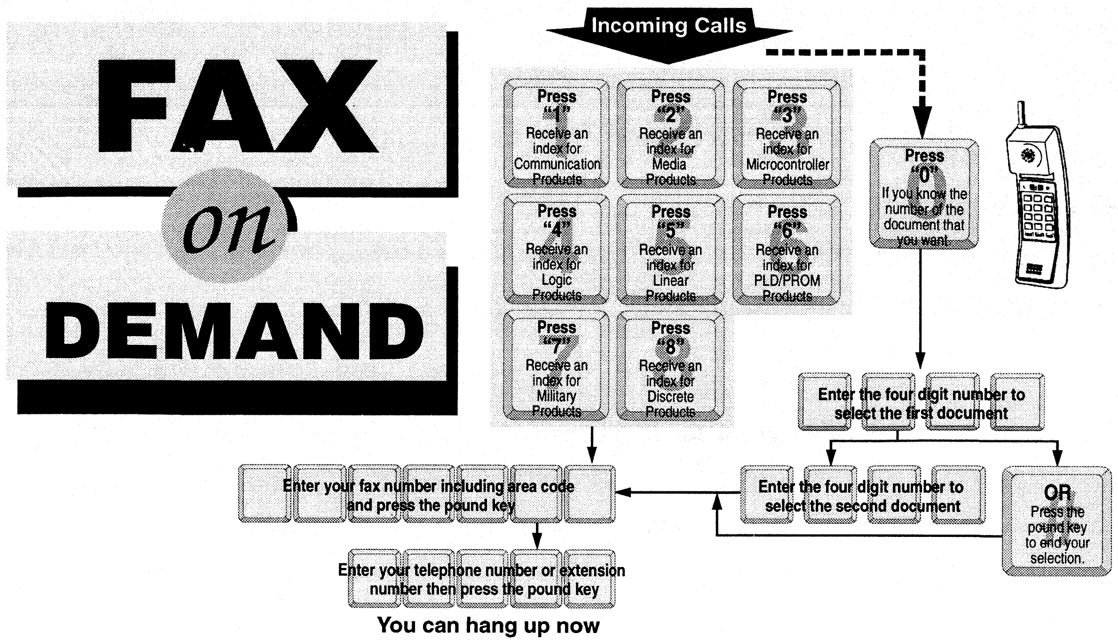
16-BIT CONTROLLERS (XA ARCHITECTURE)

TYPE	(EP)ROM	RAM	SPEED (MHz)	FUNCTIONS	REMARKS	DEVELOPMENT TOOLS
XA-G1	8k	512	30	3 timers, watchdog, 2 UARTs	–40 to +125°C	Nohau Ashling Future Designs MacCraigor Systems
XA-G2	16k	512	30	3 timers, watchdog, 2 UARTs	–40 to +125°C	Nohau Ashling Future Designs MacCraigor Systems
XA-G3	32k	512	30	3 timers, watchdog, 2 UARTs	–40 to +125°C	Nohau Ashling Future Designs MacCraigor Systems
XA–S3	32K	1K	30	A/D converter, 3 Timers, PCA, watchdog, 2 UARTS, I^{2}C		Nohau Future Designs

16-BIT CONTROLLERS (68000 ARCHITECTURE)

TYPE	(EP)ROM	RAM	SPEED (MHz)	FUNCTIONS	REMARKS	PHILIPS TOOLS	THIRD-PARTY TOOLS
68070	–	–	17.5	2 DMA channels, MMU, UART, 16-bit timer, I^{2}C, 68000 bus interface, 16Mb address range		OM4160 Microcore 1 OM4160/2 Microcore 2 OM4161 (SBE68070) OM4767/2 XRAY68070SBE high level symbolic debugger OM4222 68070DS development system OM4226 XRAY68070DS high level symbolic debugger	TRACE32-ICE68070 (Lauterbach)
93C101	34k	512	15	Derivative with low power modes	Not for new design		
90CE201	16MB external ROM	16MB external RAM	24	UART, fast I^{2}C, 3 timers (16 bit), Watchdog timer. 68000 software compatible, EMC, QFP64	–25 to +85°C	OM4162 Microcore 4	TRACE32 – (Lauterbach)
P90CL301	16MB external ROM 256 internal	512	27	2xUART, 12C, 2xtimer (16–bit), watchdog timer (21–bit), low power modes, 2xPWM (8–bit), 4xinput ADC (8–bit), LQPF80		OM5040 Microcore 5	TRACE32 (Lauterbach)

FAX-on-DEMAND System


FAX
on
DEMAND
Incoming Calls
Press "1" Receive an index for Communication Products
Press "2" Receive an index for Media Products
Press "3" Receive an index for Microcontroller Products
Press "4" Receive an index for Logic Products
Press "5" Receive an index for Linear Products
Press "6" Receive an index for PLD/PROM Products
Press "7" Receive an index for Military Products
Press "8" Receive an index for Discrete Products
Press "0" If you know the number of the document that you want
Enter the four digit number to select the first document
Enter the four digit number to select the second document
OR Press the pound key to end your selection.
Enter your fax number including area code and press the pound key
Enter your telephone number or extension number then press the pound key
You can hang up now

What is it?

The FAX-on-DEMAND system is a computer facsimile system that allows customers to receive selected documents by fax automatically.

How does it work?

To order a document, you simply enter the document number. This number can be obtained by asking for an index of available documents to be faxed to you the first time you call the system.

Our system has a selection of the latest product data sheets from Philips with varying page counts. As you know, it takes approximately one minute to FAX one page. This isn't bad if the number of pages is less than 10. But if the document is 37 pages long, be ready for a long transmission!

Philips Semiconductors also maintains product information on the World-Wide Web. Our home page can be located at:
http://www.semiconductors.philips.com

Who do I contact if I have a question about FAX-on-DEMAND?

Contact your local Philips sales office.

FAX-on-DEMAND phone numbers:

United Kingdom, Ireland, Benelux & Scandinavia	+44-181-730-5020
Italy (only)	+39-167-295502
North America	1-800-282-2000
Asia/Pacific (Australia, China/HK, India, Indonesia, Japan, Korea, Malaysia, New Zealand, Philippines, Singapore, Taiwan & Thailand)	+852 2811 9990

Microcontroller internet and bulletin board access

INTERNET ACCESS

Philips Semiconductors World Wide Web:

http://www.semiconductors.philips.com

Microcontroller Support Files:

Using a web browser:	www.philipsmcu.com
Using FTP:	ftp.philipsmcu.com

Philips Microcontroller Discussion Forum:

Send forum messages to:	forum@philipsmcu.com
Forum messages on the web:	webforum.philipsmcu.com/
Email forum Subscriptions*:	forum–request@philipsmcu.com

Philips Microcontroller Newsletter:

Newsletter Subscriptions*:	news–request@philipsmcu.com

80C51 Applications Support Email Address:

80C51_help@sv.sc.philips.com

XA Applications Support Email Address:

XA_help@sv.sc.philips.com

* These are email-oriented internet services. To subscribe, send an email to the internet address listed above and include ‘subscribe’ in the subject category.

BULLETIN BOARD

To better serve our customers, Philips maintains a microcontroller bulletin board. This computer bulletin board system features microcontroller newsletters, application and demonstration programs for download, and the ability to send messages to microcontroller application engineers.

The telephone number is:

+31 40 2721102
MAX 14.400 baud
Standards V32/V42/V42.bis/HST
(The Netherlands)

Files from the former North American Bulletin Board are available on the world wide web (see previous page).

Sunnyvale ROMcode Bulletin Board

We also have a ROM code bulletin board through which you can submit ROM codes. This is a closed bulletin board for security reasons. To get an ID, contact your local sales office. The system can be accessed with a 2400, 1200, or 300 baud modem, and is available 24 hours a day.

The telephone number is:

(408) 991-3459

All code for application notes in this databook are available on the Philips web site.

Ordering Information

MICROCONTROLLER PRODUCTS

Example: P 8 X C X X X E B P N

0 = ROMLESS
5 = Bond-Out (emulation)
3 = ROM
7 = OTP – EPROM
9 = FEEPROM (FLASH)

Exceptions:
P80C31, P80C32 = ROMless
P80C87, P80C52 = ROM

Device Number

Speed
C = 12MHz
E = 3.5MHz to 16MHz
F = 1.2MHz to 16MHz
G = 20MHz
H = 32kHz to 12MHz
I = 24MHz
P = 40MHz
S = 0 to 16 MHz
T = 0 to 24 MHz
U = 0 to 33 MHz

Philips North America Package Code
A = Plastic Leaded Chip Carrier (PLCC)
B = Quad Flat Pack (QFP)
FA = Hermetic Cerdip (window)
KA = CerQuad (window)
N = Plastic Dual In-Line

Philips Package Code
A = Plastic Leaded Chip Carrier (PLCC)
B = Quad Flat Pack (QFP)
F = Hermetic Cerdip (window)
L = Cerquad (window)
P = Plastic Dual In-Line
Q = Ceramic Quad Flat Pack (window)

Temperature
B = 0°C to +70°C
F = –40°C to +85°C
H = –40°C to +125°C

Example: SC 8 X C X X X B C C N 40

Memory Options:
0 = ROMLESS
3 = ROM
7 = OTP – EPROM

Device Number

Pin Count

Package Code
A = Plastic Leaded Chip Carrier (PLCC)
B = Quad Flat Pack (QFP)
F = Ceramic Dual In-Line
FA = Hermetic Cerdip (window)
KA = CerQuad (window)
L = Chip Carrier, Leaded
N = Plastic Dual In-Line

Speed
B = 0.5 to 12MHz
C = 12MHz
E = 3.5MHz to 16MHz
F = 1.2MHz to 16MHz
G = 20MHz
H = 32kHz to 12MHz
I = 24MHz
P = 40MHz

Temperature
C = Commercial 0°C to +70°C
A = Industrial –40°C to +85°C

Revision (optional)

Ordering Information

MICROCONTROLLER PRODUCTS (Continued)

Example: S 8 X C XXXX –1 N 24

S = Standard
P = Static Design

Memory Options:
0 = ROMLESS
3 = ROM
7 = OTP – EPROM

Device Number

Pin Count (optional)

Package Code
A = Plastic Leaded Chip Carrier (PLCC)
B = Quad Flat Pack (QFP)
F = Ceramic Dual In-Line
FA = Hermetic Cerdip (window)
KA = CerQuad (window)
L = Chip Carrier, Leaded
N = Plastic Dual In-LIne

Speed / Temperature Range
1 = 12MHz, 0°C to +70°C
2 = 12MHz, –40°C to +85°C
3 = 0.5 to 12MHz, 0°C to 70°C
4 = 16MHz, 0°C to 70°C
5 = 16MHz, –40°C to +85°C
6 = 12 or 16MHz, –55°C to +125°C
7 = 20MHz, 0°C to +70°C
8 = 20MHz, –40°C to +85°C
A = 24MHz, 0°C to +70°C
B = 24MHz, –40°C to +85°C
I = 33MHz, 0°C to 70°C
J = 33MHz, –40°C to +85°C

Note: – for FX
+ for RX

MICROCONTROLLER PRODUCTS – XA

Example: P51XA G3 7 K B A

Philips 80C51 eXtended Architecture

Derivative Name

Package Code
A = Plastic Leaded Chip Carrier (PLCC)
B = Quad Flat Pack (QFP)
BD = Thin Quad Flat Pack (TQFP)
FA = Hermetic Cerdip (window)
KA = CerQuad (window)
N = Plastic Dual In-Line

Temperature
B = 0°C to +70°C
F = –40°C to +85°C
H = –40°C to +125°C

Speed
E = 16MHz
G = 20MHz
J = 25MHz
K = 30MHz

Memory Opation
0 = ROMless
3 = ROM
5 = Bond-Out (emulation)
7 = EPROM/OTP
9 = FEEPROM (FLASH)

Section 2
80C51 Technical Description

CONTENTS

80C51 family architecture 43
80C51 family hardware description 58
80C51 family programmer's guide and instruction set 82
80C51 family EPROM products 137

80C51 ARCHITECTURE

MEMORY ORGANIZATION

All 80C51 devices have separate address spaces for program and data memory, as shown in Figures 1 and 2. The logical separation of program and data memory allows the data memory to be accessed by 8-bit addresses, which can be quickly stored and manipulated by an 8-bit CPU. Nevertheless, 16-bit data memory addresses can also be generated through the DPTR register.

Program memory (ROM, EPROM) can only be read, not written to. There can be up to 64k bytes of program memory. In the 80C51, the lowest 4k bytes of program are on-chip. In the ROMless versions, all program memory is external. The read strobe for external program memory is the $\overline{\text{PSEN}}$ (program store enable).

Data Memory (RAM) occupies a separate address space from Program Memory. In the 80C51, the lowest 128 bytes of data memory are on-chip. Up to 64k bytes of external RAM can be addressed in the external Data Memory space. In the ROMless version, the lowest 128 bytes are on-chip. The CPU generates read and write signals, $\overline{\text{RD}}$ and $\overline{\text{WR}}$, as needed during external Data Memory accesses.

External Program Memory and external Data Memory may be combined if desired by applying the RD and PSEN signals to the inputs of an AND gate and using the output of the gate as the read strobe to the external Program/Data memory.

Program Memory

Figure 3 shows a map of the lower part of the Program Memory. After reset, the CPU begins execution from location 0000H. As shown in Figure 3, each interrupt is assigned a fixed location in Program Memory. The interrupt causes the CPU to jump to that location, where it commences execution of the service routine. External Interrupt 0, for example, is assigned to location 0003H. If External Interrupt 0 is going to be used, its service routine must begin at location 0003H. If the interrupt is not going to be used, its service location is available as general purpose Program Memory.

The interrupt service locations are spaced at 8-byte intervals: 0003H for External Interrupt 0, 000BH for Timer 0, 0013H for External Interrupt 1, 001BH for Timer 1, etc. If an interrupt service routine is short enough (as is often the case in control applications), it can reside entirely within that 8-byte interval. Longer service routines can use a jump instruction to skip over subsequent interrupt locations, if other interrupts are in use.

The lowest 4k bytes of Program Memory can either be in the on-chip ROM or in an external ROM. This selection is made by strapping the $\overline{\text{EA}}$ (External Access) pin to either V_{CC}, or V_{SS}. In the 80C51, if the $\overline{\text{EA}}$ pin is strapped to V_{CC}, then the program fetches to addresses 0000H through 0FFFH are directed to the internal ROM. Program fetches to addresses 1000H through FFFFH are directed to external ROM.

If the $\overline{\text{EA}}$ pin is strapped to V_{SS}, then all program fetches are directed to external ROM. The ROMless parts (8031, 80C31, etc.) must have this pin externally strapped to V_{SS} to enable them to execute from external Program Memory.

The read strobe to external ROM, $\overline{\text{PSEN}}$, is used for all external program fetches. $\overline{\text{PSEN}}$ is not activated for internal program fetches.

The hardware configuration for external program execution is shown in Figure 4. Note that 16 I/O lines (Ports 0 and 2) are dedicated to bus functions during external Program Memory fetches. Port 0 (P0 in Figure 4) serves as a multiplexed address/data bus. It emits the low byte of the Program Counter (PCL) as an address, and then goes into a float state awaiting the arrival of the code byte from the Program Memory. During the time that the low byte of the Program Counter is valid on Port 0, the signal ALE (Address Latch Enable) clocks this byte into an address latch. Meanwhile, Port 2 (P2 in Figure 4) emits the high byte of the Program Counter (PCH). Then $\overline{\text{PSEN}}$ strobes the EPROM and the code byte is read into the microcontroller.

Program Memory addresses are always 16 bits wide, even though the actual amount of Program Memory used may be less than 64k bytes. External program execution sacrifices two of the 8-bit ports, P0 and P2, to the function of addressing the Program Memory.

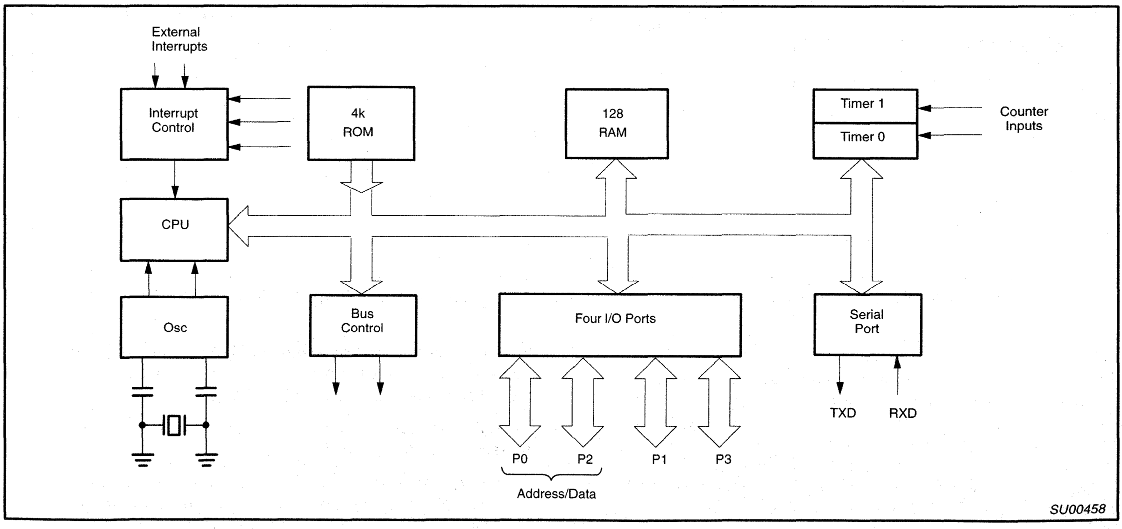


Figure 1. 80C51 Block Diagram

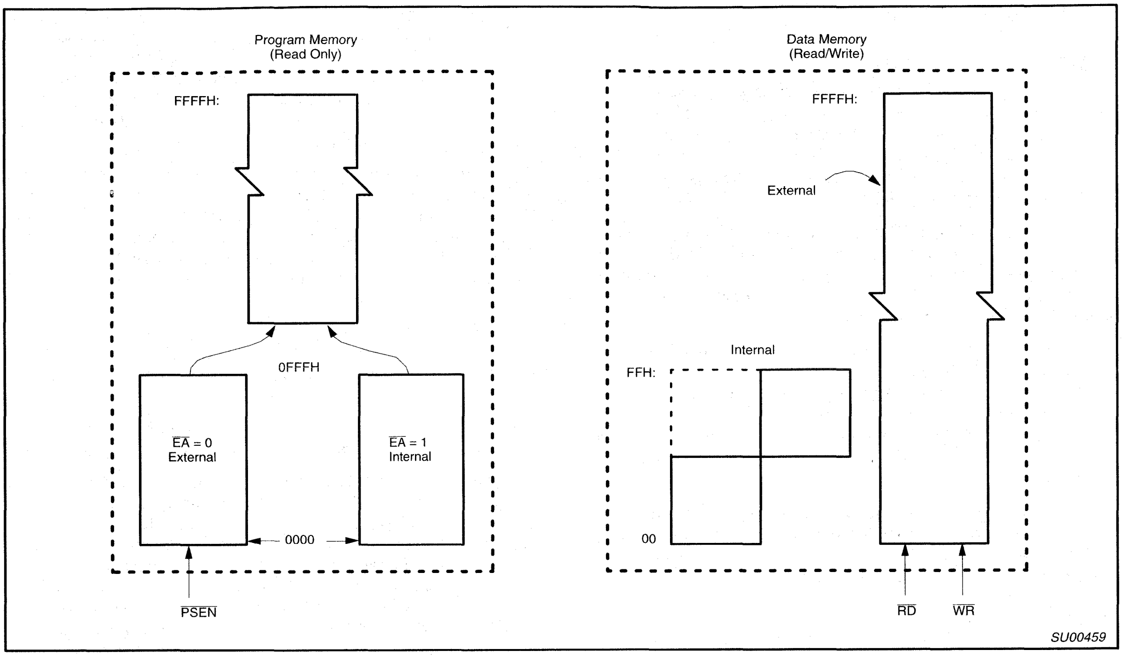


Figure 2. 80C51 Memory Structure

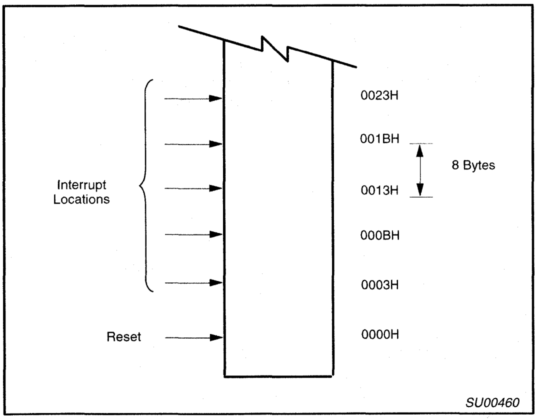


Figure 3. 80C51 Program Memory

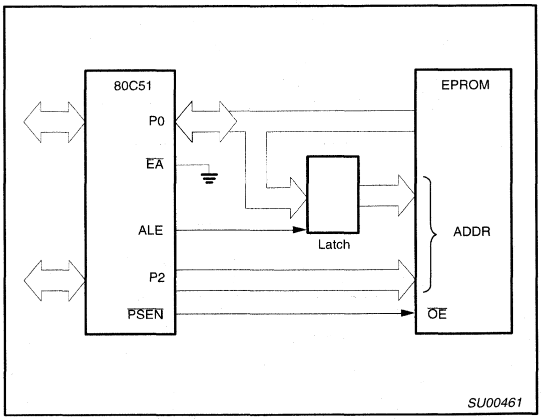


Figure 4. Executing from External Program Memory

Data Memory

The right half of Figure 2 shows the internal and external Data Memory spaces available to the 80C51 user. Figure 5 shows a hardware configuration for accessing up to 2k bytes of external RAM. The CPU in this case is executing from internal ROM. Port 0 serves as a multiplexed address/data bus to the RAM, and 3 lines of Port 2 are being used to page the RAM. The CPU generates $\overline{RD}$ and $\overline{WR}$ signals as needed during external RAM accesses. There can be up to 64k bytes of external Data Memory. External Data Memory addresses can be either 1 or 2 bytes wide. One-byte addresses are often used in conjunction with one or more other I/O lines to page the RAM, as shown in Figure 5.

Two-byte addresses can also be used, in which case the high address byte is emitted at Port 2.

Internal Data Memory is mapped in Figure 6. The memory space is shown divided into three blocks, which are generally referred to as the Lower 128, the Upper 128, and SFR space.

Internal Data Memory addresses are always one byte wide, which implies an address space of only 256 bytes. However, the addressing modes for internal RAM can in fact accommodate 384 bytes, using a simple trick. Direct addresses higher than 7FH access one memory space, and indirect addresses higher than 7FH access a different memory space. Thus Figure 6 shows the Upper 128 and SFR space occupying the same block of addresses, 80H through FFH, although they are physically separate entities.

The Lower 128 bytes of RAM are present in all 80C51 devices as mapped in Figure 7. The lowest 32 bytes are grouped into 4 banks of 8 registers. Program instructions call out these registers as R0 through R7. Two bits in the Program Status Word (PSW) select which register bank is in use. This allows more efficient use of code space, since register instructions are shorter than instructions that use direct addressing.

The next 16 bytes above the register banks form a block of bit-addressable memory space. The 80C51 instruction set includes a wide selection of single-bit instructions, and the 128 bits in this area can be directly addressed by these instructions. The bit addresses in this area are 00H through 7FH.

All of the bytes in the Lower 128 can be accessed by either direct or indirect addressing. The Upper 128 (Figure 8) can only be accessed by indirect addressing.

Figure 9 gives a brief look at the Special Function Register (SFR) space. SFRs include the Port latches, timers, peripheral controls, etc. These registers can only be accessed by direct addressing. Sixteen addresses in SFR space are both byte- and bit-addressable. The bit-addressable SFRs are those whose address ends in 0H or 8H.

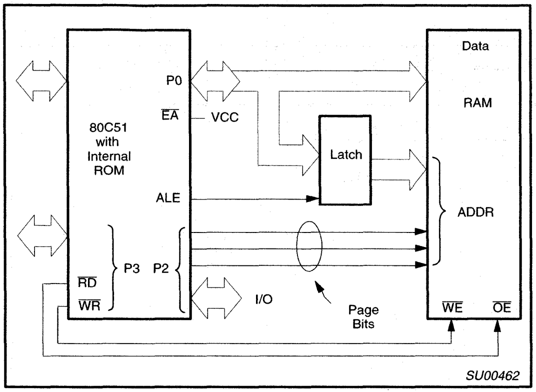


Figure 5. Accessing External Data Memory If the Program Memory Is Internal, the Other Bits of P2 Are Available as I/O

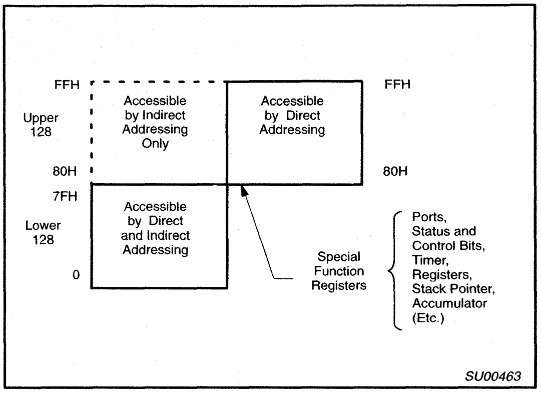


Figure 6. Internal Data Memory

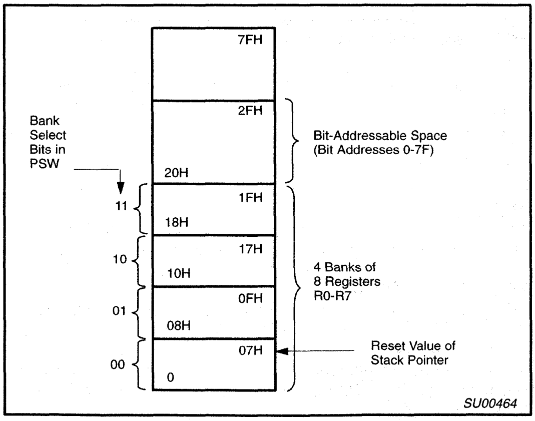


Figure 7. Lower 128 Bytes of Internal RAM

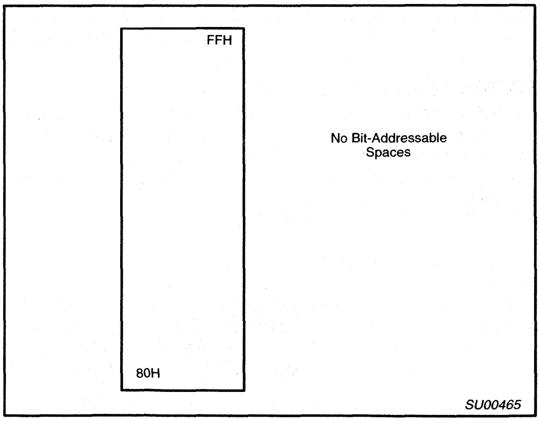


Figure 8. Upper 128 Bytes of Internal RAM

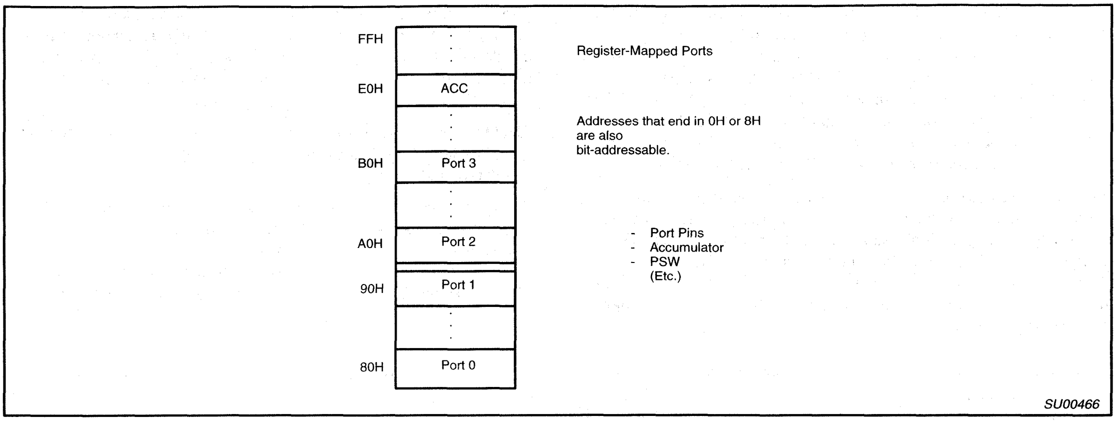


Figure 9. SFR Space

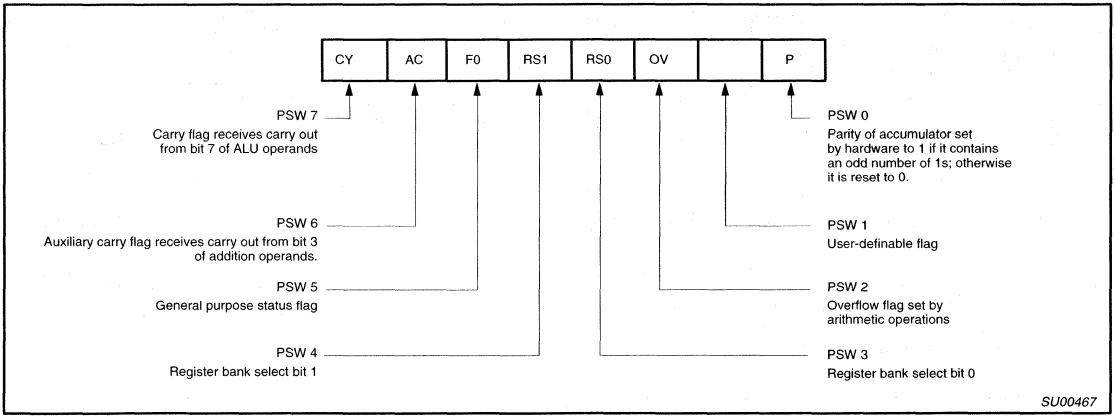

CY
AC
F0
RS1
RS0
OV
P
PSW 7
Carry flag receives carry out from bit 7 of ALU operands
PSW 6
Auxiliary carry flag receives carry out from bit 3 of addition operands.
PSW 5
General purpose status flag
PSW 4
Register bank select bit 1
PSW 0
Parity of accumulator set by hardware to 1 if it contains an odd number of 1s; otherwise it is reset to 0.
PSW 1
User-definable flag
PSW 2
Overflow flag set by arithmetic operations
PSW 3
Register bank select bit 0
SU00467

Figure 10. PSW (Program Status Word) Register in 80C51 Devices

80C51 FAMILY INSTRUCTION SET

The 80C51 instruction set is optimized for 8-bit control applications. It provides a variety of fast addressing modes for accessing the internal RAM to facilitate byte operations on small data structures. The instruction set provides extensive support for one-bit variables as a separate data type, allowing direct bit manipulation in control and logic systems that require Boolean processing.

Program Status Word

The Program Status Word (PSW) contains several status bits that reflect the current state of the CPU. The PSW, shown in Figure 10, resides in the SFR space. It contains the Carry bit, the Auxiliary Carry (for BCD operations), the two register bank select bits, the Overflow flag, a Parity bit, and two user-definable status flags.

The Carry bit, other than serving the function of a Carry bit in arithmetic operations, also serves as the "Accumulator" for a number of Boolean operations.

The bits RS0 and RS1 are used to select one of the four register banks shown in Figure 7. A number of instructions refer to these RAM locations as R0 through R7. The selection of which of the four is being referred to is made on the basis of the RS0 and RS1 at execution time.

The Parity bit reflects the number of 1s in the Accumulator: P = 1 if the Accumulator contains an odd number of 1s, and P = 0 if the Accumulator contains an even number of 1s. Thus the number of 1s in the Accumulator plus P is always even. Two bits in the PSW are uncommitted and may be used as general purpose status flags.

Addressing Modes

The addressing modes in the 80C51 instruction set are as follows:

Direct Addressing

In direct addressing the operand is specified by an 8-bit address field in the instruction. Only internal Data RAM and SFRs can be directly addressed.

Indirect Addressing

In indirect addressing the instruction specifies a register which contains the address of the operand. Both internal and external RAM can be indirectly addressed.

The address register for 8-bit addresses can be R0 or R1 of the selected bank, or the Stack Pointer. The address register for 16-bit addresses can only be the 16-bit "data pointer" register, DPTR.

Register Instructions

The register banks, containing registers R0 through R7, can be accessed by certain instructions which carry a 3-bit register specification within the opcode of the instruction. Instructions that access the registers this way are code efficient, since this mode eliminates an address byte. When the instruction is executed, one of the eight registers in the selected bank is accessed. One of four banks is selected at execution time by the two bank select bits in the PSW.

Register-Specific Instructions

Some instructions are specific to a certain register. For example, some instructions always operate on the Accumulator, or Data Pointer, etc., so no address byte is needed to point to it. The opcode itself does that. Instructions that refer to the Accumulator as A assemble as accumulator specific opcodes.

Immediate Constants

The value of a constant can follow the opcode in Program Memory. For example,

```
MOV   A, #100
```

loads the Accumulator with the decimal number 100. The same number could be specified in hex digits as 64H.

Indexed Addressing

Only program Memory can be accessed with indexed addressing, and it can only be read. This addressing mode is intended for reading look-up tables in Program Memory A 16-bit base register (either DPTR or the Program Counter) points to the base of the table, and the Accumulator is set up with the table entry number.

The address of the table entry in Program Memory is formed by adding the Accumulator data to the base pointer.

Another type of indexed addressing is used in the "case jump" instruction. In this case the destination address of a jump instruction is computed as the sum of the base pointer and the Accumulator data.

Arithmetic Instructions

The menu of arithmetic instructions is listed in Table 1. The table indicates the addressing modes that can be used with each instruction to access the <byte> operand. For example, the ADD A,<byte> instruction can be written as:

```
ADD     a, 7FH (direct addressing)
ADD     A, @R0 (indirect addressing)
ADD     a, R7 (register addressing)
ADD     A, #127 (immediate constant)
```

The execution times listed in Table 1 assume a 12MHz clock frequency. All of the arithmetic instructions execute in 1µs except the INC DPTR instruction, which takes 2µs, and the Multiply and Divide instructions, which take 4µs.

Note that any byte in the internal Data Memory space can be incremented without going through the Accumulator.

One of the INC instructions operates on the 16-bit Data Pointer. The Data Pointer is used to generate 16-bit addresses for external memory, so being able to increment it in one 16-bit operation is a useful feature.

The MUL AB instruction multiplies the Accumulator by the data in the B register and puts the 16-bit product into the concatenated B and Accumulator registers.

The DIV AB instruction divides the Accumulator by the data in the B register and leaves the 8-bit quotient in the Accumulator, and the 8-bit remainder in the B register.

Oddly enough, DIV AB finds less use in arithmetic "divide" routines than in radix conversions and programmable shift operations. An example of the use of DIV AB in a radix conversion will be given later. In shift operations, dividing a number by 2n shifts its n bits to the right. Using DIV AB to perform the division completes the shift in 4µs and leaves the B register holding the bits that were shifted out. The DA A instruction is for BCD arithmetic operations. In BCD arithmetic, ADD and ADDC instructions should always be followed by a DA A operation, to ensure that the result is also in BCD. Note that DA A will not convert a binary number to BCD. The DA A operation produces a meaningful result only as the second step in the addition of two BCD bytes.

Table 1. 80C51 Arithmetic Instructions

MNEMONIC	OPERATION	ADDRESSING MODES				EXECUTION
		DIR	IND	REG	IMM	TIME (µs)
ADD A,<byte>	A = A + <byte>	X	X	X	X	1
ADDC A,<byte>	A = A + <byte> + C	X	X	X	X	1
SUBB A,<byte>	A = A – <byte> – C	X	X	X	X	1
INC A	A = A + 1	Accumulator only				1
INC <byte>	<byte> = <byte> + 1	X	X	X		1
INC DPTR	DPTR = DPTR + 1	Data Pointer only				2
DEC A	A = A – 1	Accumulator only				1
DEC <byte>	<byte> = <byte> – 1	X	X	X		1
MUL AB	B:A = B x A	ACC and B only				4
DIV AB	A = Int[A/B] B = Mod[A/B]	ACC and B only				4
DA A	Decimal Adjust	Accumulator only				1

Logical Instructions

Table 2 shows the list of 80C51 logical instructions. The instructions that perform Boolean operations (AND, OR, Exclusive OR, NOT) on bytes perform the operation on a bit-by-bit basis. That is, if the Accumulator contains 00110101B and byte contains 01010011B, then:

```
ANL    A, <byte>
```

will leave the Accumulator holding 00010001B.

The addressing modes that can be used to access the <byte> operand are listed in Table 2.

The ANL A, <byte> instruction may take any of the forms:

```
ANL    A,7FH (direct addressing)
ANL    A,@R1 (indirect addressing)
ANL    A,R6 (register addressing)
ANL    A,#53H (immediate constant)
```

All of the logical instructions that are Accumulator-specific execute in 1µs (using a 12MHz clock). The others take 2µs.

Note that Boolean operations can be performed on any byte in the internal Data Memory space without going through the Accumulator. The XRL <byte>, #data instruction, for example, offers a quick and easy way to invert port bits, as in XRL P1, #OFFH.

If the operation is in response to an interrupt, not using the Accumulator saves the time and effort to push it onto the stack in the service routine.

The Rotate instructions (RL, A, RLC A, etc.) shift the Accumulator 1 bit to the left or right. For a left rotation, the MSB rolls into the LSB position. For a right rotation, the LSB rolls into the MSB position.

The SWAP A instruction interchanges the high and low nibbles within the Accumulator. This is a useful operation in BCD manipulations. For example, if the Accumulator contains a binary number which is known to be less than 100, it can be quickly converted to BCD by the following code:

```
MOVE  B,#10
DIV   AB
SWAP  A
ADD   A,B
```

Dividing the number by 10 leaves the tens digit in the low nibble of the Accumulator, and the ones digit in the B register. The SWAP and ADD instructions move the tens digit to the high nibble of the Accumulator, and the ones digit to the low nibble.

Data Transfers

Internal RAM

Table 3 shows the menu of instructions that are available for moving data around within the internal memory spaces, and the addressing modes that can be used with each one. With a 12MHz clock, all of these instructions execute in either 1 or 2µs.

The MOV <dest>, <src> instruction allows data to be transferred between any two internal RAM or SFR locations without going through the Accumulator. Remember, the Upper 128 bytes of data RAM can be accessed only by indirect addressing, and SFR space only by direct addressing.

Note that in 80C51 devices, the stack resides in on-chip RAM, and grows upwards. The PUSH instruction first increments the Stack Pointer (SP), then copies the byte into the stack. PUSH and POP use only direct addressing to identify the byte being saved or restored, but the stack itself is accessed by indirect addressing using the SP register. This means the stack can go into the Upper 128 bytes of RAM, if they are implemented, but not into SFR space.

The Upper 128 bytes of RAM are not implemented in the 80C51 nor in its ROMless or EPROM counterparts. With these devices, if the SP points to the Upper 128, PUSHed bytes are lost, and POPed bytes are indeterminate.

The Data Transfer instructions include a 16-bit MOV that can be used to initialize the Data Pointer (DPTR) for look-up tables in Program Memory, or for 16-bit external Data Memory accesses.

Table 2. 80C51 Logical Instructions

MNEMONIC	OPERATION	ADDRESSING MODES				EXECUTION
		DIR	IND	REG	IMM	TIME (µs)
ANL A,<byte>	A = A.AND. <byte>	X	X	X	X	1
ANL <byte>,A	<byte> = <byte> .AND.A	X				1
ANL <byte>,#data	<byte> = <byte> .AND.#data	X				2
ORL A,<byte>	A = A.OR.<byte>	X	X	X	X	1
ORL <byte>,A	<byte> = <byte> .OR.A	X				1
ORL <byte>,#data	<byte> = <byte> .OR.#data	X				2
XRL A,<byte>	A = A.XOR. <byte>	X	X	X	X	1
XRL <byte>,A	<byte> = <byte> .XOR.A	X				1
XRL <byte>,#data	<byte> = <byte> .XOR.#data	X				2
CRL A	A = 00H	Accumulator only				1
CPL A	A = .NOT.A	Accumulator only				1
RL A	Rotate ACC Left 1 bit	Accumulator only				1
RLC A	Rotate Left through Carry	Accumulator only				1
RR A	Rotate ACC Right 1 bit	Accumulator only				1
RRC A	Rotate Right through Carry	Accumulator only				1
SWAP A	Swap Nibbles in A	Accumulator only				1

Table 3. Data Transfer Instructions that Access Internal Data Memory Space

MNEMONIC	OPERATION	ADDRESSING MODES				EXECUTION
		DIR	IND	REG	IMM	TIME (µs)
MOV A,<src>	A = <src>	X	X	X	X	1
MOV <dest>,A	<dest> = A	X	X	X		1
MOV <dest>,<src>	<dest> = <src>	X	X	X	X	2
MOV DPTR,#data16	DPTR = 16-bit immediate constant				X	2
PUSH <src>	INC SP:MOV"@SP",<src>	X				2
POP <dest>	MOV <dest>,"@SP":DEC SP	X				2
XCH A,<byte>	ACC and <byte> exchange data	X	X	X		1
XCHD A,@Ri	ACC and @Ri exchange low nibbles		X			1

The XCH A, <byte> instruction causes the Accumulator and addressed byte to exchange data. The XCHD A, @Ri instruction is similar, but only the low nibbles are involved in the exchange.

To see how XCH and XCHD can be used to facilitate data manipulations, consider first the problem of shifting an 8-digit BCD number two digits to the right. Figure 11 shows how this can be done using direct MOVs, and for comparison how it can be done using XCH instructions. To aid in understanding how the code works, the contents of the registers that are holding the BCD number and the content of the Accumulator are shown alongside each instruction to indicate their status after the instruction has been executed.

After the routine has been executed, the Accumulator contains the two digits that were shifted out on the right. Doing the routine with direct MOVs uses 14 code bytes and 9µs of execution time (assuming a 12MHz clock). The same operation with XCHs uses only 9 bytes and executes almost twice as fast.

To right-shift by an odd number of digits, a one-digit shift must be executed.

Figure 12 shows a sample of code that will right-shift a BCD number one digit, using the XCHD instruction. Again, the contents of the registers holding the number and of the Accumulator are shown alongside each instruction.

First, pointers R1 and R0 are set up to point to the two bytes containing the last four BCD digits. Then a loop is executed which leaves the last byte, location 2EH, holding the last two digits of the shifted number. The pointers are decremented, and the loop is repeated for location 2DH. The CJNE instruction (Compare and Jump if Not Equal) is a loop control that will be described later. The loop executed from LOOP to CJNE for R1 = 2EH, 2DH, 2CH, and 2BH. At that point the digit that was originally shifted out on the right has propagated to location 2AH. Since that location should be left with 0s, the lost digit is moved to the Accumulator.

External RAM

Table 4 shows a list of the Data Transfer instructions that access external Data Memory. Only indirect addressing can be used. The choice is whether to use a one-byte address, @Ri, where Ri can be either R0 or R1 of the selected register bank, or a two-byte address, @DPTR. The disadvantage to using 16-bit addresses if only a few k bytes of external RAM are involved is that 16-bit addresses use all 8 bits of Port 2 as address bus. On the other hand, 8-bit addresses allow one to address a few bytes of RAM, as shown in Figure 5, without having to sacrifice all of Port 2. All of these instructions execute in 2 µs, with a 12MHz clock.

Note that in all external Data RAM accesses, the Accumulator is always either the destination or source of the data.

The read and write strobes to external RAM are activated only during the execution of a MOVX instruction. Normally these signals are inactive, and in fact if they're not going to be used at all, their pins are available as extra I/O lines.

```
                     2A   2B   2C   2D   2E   ACC
MOV   A,2EH          00   12   34   56   78   78
MOV   2EH,2DH        00   12   34   56   56   78
MOV   2DH,2CH        00   12   34   34   56   78
MOV   2CH,2BH        00   12   12   34   56   78
MOV   2BH,#0         00   00   12   34   56   78
```

A. Using direct MOVs: 14 bytes, 9 µs

```
                     2A   2B   2C   2D   2E   ACC
CLR   A              00   12   34   56   78   00
XCH   A,2BH          00   00   34   56   78   12
XCH   A,2CH          00   00   12   56   78   34
XCH   A,2DH          00   00   12   34   78   56
XCH   A2EH           00   00   12   34   56   78
```

B. Using XCHs: 9 bytes, 5 µs

SU00468

Figure 11. Shifting a BCD Number Two Digits to the Right

```
                                2A   2B   2C   2D   2E   ACC
         MOV    R1,#2EH         00   12   34   56   78   XX
         MOV    R0,#2DH         00   12   34   56   78   XX

       loop for R1 = 2EH:

LOOP:    MOV    A,@R1           00   12   34   56   78   78
         XCHD   A,@R0           00   12   34   58   78   76
         SWAP   A               00   12   34   58   78   67
         MOV    @R1,A           00   12   34   58   67   67
         DEC    R1              00   12   34   58   67   67
         DEC    R0              00   12   34   58   67   67
         CJNE   R1,#2AH,LOOP


       loop for R1 = 2DH:       00   12   38   45   67   45
       loop for R1 = 2CH:       00   18   23   45   67   23
       loop for R1 = 2BH:       08   01   23   45   67   01


         CLR    A               08   01   23   45   67   00
         XCH    A,2AH           00   01   23   45   67   08
```

SU00469

Figure 12. Shifting a BCD Number One Digit to the Right

Table 4. 80C51 Data Transfer Instructions that Access External Data Memory Space

ADDRESS WIDTH	MNEMONIC	OPERATION	EXECUTION TIME (μs)
8 bits	MOVX A,@Ri	Read external RAM @Ri	2
8 bits	MOVX @Ri,A	Write external RAM @ Ri	2
16 bits	MOVX A,@DPTR	Read external RAM @ DPTR	2
16 bits	MOVX @DPTR,A	Write external RAM @ DPTR	2

Table 5. 80C51 Lookup Table Read Instructions

MNEMONIC	OPERATION	EXECUTION TIME (μs)
MOVC A,@A+DPTR	Read program memory at (A + DPTR)	2
MOVC A,@A+PC	Read program memory at (A + PC)	2

Lookup Tables

Table 5 shows the two instructions that are available for reading lookup tables in Program Memory. Since these instructions access only Program Memory, the lookup tables can only be read, not updated.

If the table access is to external Program Memory, then the read strobe is $\overline{\text{PSEN}}$.

The mnemonic is MOVC for "move constant." The first MOVC instruction in Table 5 can accommodate a table of up to 256 entries numbered 0 through 255. The number of the desired entry is loaded into the Accumulator, and the Data Pointer is set up to point to the beginning of the table. Then:

```
        MOVC    A,@A+DPTR
```

copies the desired table entry into the Accumulator.

The other MOVC instruction works the same way, except the Program Counter (PC) is used as the table base, and the table is accessed through a subroutine. First the number of the desired entry is loaded into the Accumulator, and the subroutine is called:

```
        MOV     A,ENTRY NUMBER
        CALL    TABLE
```

The subroutine "TABLE" would look like this:

```
TABLE:  MOVC    A,@A+PC
        RET
```

The table itself immediately follows the RET (return) instruction in Program Memory. This type of table can have up to 255 entries, numbered 1 through 255. Number 0 cannot be used, because at the time the MOVC instruction is executed, the PC contains the address of the RET instruction. An entry numbered 0 would be the RET opcode itself.

Boolean Instructions

80C51 devices contain a complete Boolean (single-bit) processor. The internal RAM contains 128 addressable bits, and the SFR space can support up to 128 addressable bits as well. All of the port lines are bit-addressable, and each one can be treated as a separate single-bit port. The instructions that access these bits are not just conditional branches, but a complete menu of move, set, clear, complement, OR, and AND instructions. These kinds of bit operations are not easily obtained in other architectures with any amount of byte-oriented software.

The instruction set for the Boolean processor is shown in Table 6. All bit accesses are by direct addressing.

Bit addresses 00H through 7FH are in the Lower 128, and bit addresses 80H through FFH are in SFR space.

Note how easily an internal flag can be moved to a port pin:

```
        MOV     C,FLAG
        MOV     P1.0,C
```

In this example, FLAG is the name of any addressable bit in the Lower 128 or SFR space. An I/O line (the LSB of Port 1, in this case) is set or cleared depending on whether the flag bit is 1 or 0.

The Carry bit in the PSW is used as the single-bit Accumulator of the Boolean processor. Bit instructions that refer to the Carry bit as C assemble as Carry-specific instructions (CLR C, etc.). The Carry bit also has a direct address, since it resides in the PSW register, which is bit-addressable.

Note that the Boolean instruction set includes ANL and ORL operations, but not the XRL (Exclusive OR) operation. An XRL operation is simple to implement in software. Suppose, for example, it is required to form the Exclusive OR of two bits:

```
        C = bit1 .XRL. bit2
```

The software to do that could be as follows:

```
        MOV     C,bit1
        JNB     bit2,OVER
        CPL     C
OVER:   (continue)
```

First, bit1 is moved to the Carry. If bit2 = 0, then C now contains the correct result. That is, bit1 .XRL. bit2 = bit1 if bit2 = 0. On the other hand, if bit2 = 1, C now contains the complement of the correct result. It need only be inverted (CPL C) to complete the operation.

This code uses the JNB instruction, one of a series of bit-test instructions which execute a jump if the addressed bit is set (JC, JB, JBC) or if the addressed bit is not set (JNC, JNB). In the above case, bit2 is being tested, and if bit2 = 0, the CPL C instruction is jumped over.

JBC executes the jump if the addressed bit is set, and also clears the bit. Thus a flag can be tested and cleared in one operation. All the PSW bits are directly addressable, so the Parity bit, or the general purpose flags, for example, are also available to the bit-test instructions.

Relative Offset

The destination address for these jumps is specified to the assembler by a label or by an actual address in Program memory. However, the destination address assembles to a relative offset byte. This is a signed (two's complement) offset byte which is added to the PC in two's complement arithmetic if the jump is executed. The range of the jump is therefore –128 to +127 Program Memory bytes relative to the first byte following the instruction.

Table 6. 80C51 Boolean Instructions

MNEMONIC		OPERATION	EXECUTION TIME (μs)
ANL	C,bit	C = C.AND.bit	2
ANL	C,/bit	C = C.AND..NOT.bit	2
ORL	C,bit	C = C.OR.bit	2
ORL	C,/bit	C = C.OR..NOT.bit	2
MOV	C,bit	C = bit	1
MOV	bit,C	bit = C	2
CLR	C	C = 0	1
CLR	bit	bit = 0	1
SETB	C	C = 1	1
SETB	bit	bit = 1	1
CPL	C	C = .NOT.C	1
CPL	bit	bit = .NOT.bit	1
JC	rel	Jump if C = 1	2
JNC	rel	Jump if C = 0	2
JB	bit,rel	Jump if bit = 1	2
JNB	bit,rel	Jump if bit = 0	2
JBC	bit,rel	Jump if bit = 1; CLR bit	2

Table 7. Unconditional Jumps in 80C51 Devices

MNEMONIC		OPERATION	EXECUTION TIME (μs)
JMP	addr	Jump to addr	2
JMP	@A+DPTR	Jump to A + DPTR	2
CALL	addr	Call subroutine at addr	2
RET		Return from subroutine	2
RETI		Return from interrupt	2
NOP		No operation	1

Jump Instructions

Table 7 shows the list of unconditional jumps with execution time for a 12MHz clock.

The table lists a single "JMP addr" instruction, but in fact there are three SJMP, LJMP, and AJMP, which differ in the format of the destination address. JMP is a generic mnemonic which can be used if the programmer does not care which way the jump is encoded.

The SJMP instruction encodes the destination address as a relative offset, as described above. The instruction is 2 bytes long, consisting of the opcode and the relative offset byte. The jump distance is limited to a range of –128 to +127 bytes relative to the instruction following the SJMP.

The LJMP instruction encodes the destination address as a 16-bit constant. The instruction is 3 bytes long, consisting of the opcode and two address bytes. The destination address can be anywhere in the 64k Program Memory space.

The AJMP instruction encodes the destination address as an 11-bit constant. The instruction is 2 bytes long, consisting of the opcode, which itself contains 3 of the 11 address bits, followed by another byte containing the low 8 bits of the destination address. When the instruction is executed, these 11 bits are simply substituted for the low 11 bits in the PC. The high 5 bits stay the same. Hence the destination has to be within the same 2k block as the instruction following the AJMP.

In all cases the programmer specifies the destination address to the assembler in the same way: as a label or as a 16-bit constant. The assembler will put the destination address into the correct format for the given instruction. If the format required by the instruction will not support the distance to the specified destination address, a "Destination out of range" message is written into the List file.

The JMP @A+DPTR instruction supports case jumps. The destination address is computed at execution time as the sum of the 16-bit DPTR register and the Accumulator. Typically, DPTR is set up with the address of a jump table. In a 5-way branch, for example, an integer 0 through 4 is loaded into the Accumulator. The code to be executed might be as follows:

```
        MOV     DPTR,#JUMP TABLE
        MOV     A,INDEX_NUMBER
        RL      A
        JMP     @A+DPTR
```

The RL A instruction converts the index number (0 through 4) to an even number on the range 0 through 8, because each entry in the jump table is 2 bytes long:

```
JUMP TABLE:
        AJMP    CASE 0
        AJMP    CASE 1
        AJMP    CASE 2
        AJMP    CASE 3
        AJMP    CASE 4
```

Table 7 shows a single "CALL addr" instruction, but there are two of them, LCALL and ACALL, which differ in the format in which the subroutine address is given to the CPU. CALL is a generic mnemonic which can be used if the programmer does not care which way the address is encoded.

The LCALL instruction uses the 16-bit address format, and the subroutine can be anywhere in the 64k Program Memory space. The ACALL instruction uses the 11-bit format, and the subroutine must be in the same 2k block as the instruction following the ACALL.

In any case, the programmer specifies the subroutine address to the assembler in the same way: as a label or as a 16-bit constant. The assembler will put the address into the correct format for the given instructions.

Subroutines should end with a RET instruction, which returns execution to the instruction following the CALL.

RETI is used to return from an interrupt service routine. The only difference between RET and RETI is that RETI tells the interrupt control system that the interrupt in progress is done. If there is no interrupt in progress at the time RETI is executed, then the RETI is functionally identical to RET.

Table 8 shows the list of conditional jumps available to the 80C51 user. All of these jumps specify the destination address by the relative offset method, and so are limited to a jump distance of –128 to +127 bytes from the instruction following the conditional jump instruction. Important to note, however, the user specifies to the assembler the actual destination address the same way as the other jumps: as a label or a 16-bit constant.

There is no Zero bit in the PSW. The JZ and JNZ instructions test the Accumulator data for that condition.

The DJNZ instruction (Decrement and Jump if Not Zero) is for loop control. To execute a loop N times, load a counter byte with N and terminate the loop with a DJNZ to the beginning of the loop, as shown below for N = 10.

```
      MOV          COUNTER,#10
LOOP: (begin loop)
      •
      •
      •
      (end loop)
      DJNZ         COUNTER,LOOP
      (continue)
```

The CJNE instruction (Compare and Jump if Not Equal) can also be used for loop control as in Figure 12. Two bytes are specified in the operand field of the instruction. The jump is executed only if the two bytes are not equal. In the example of Figure 12, the two bytes were data in R1 and the constant 2AH. The initial data in R1 was 2EH. Every time the loop was executed, R1 was decremented, and the looping was to continue until the R1 data reached 2AH.

Another application of this instruction is in "greater than, less than" comparisons. The two bytes in the operand field are taken as unsigned integers. If the first is less than the second, then the Carry bit is set (1). If the first is greater than or equal to the second, then the Carry bit is cleared.

CPU Timing

All 80C51 microcontrollers have an on-chip oscillator which can be used if desired as the clock source for the CPU. To use the on-chip oscillator, connect a crystal or ceramic resonator between the XTAL1 and XTAL2 pins of the microcontroller, and capacitors to ground as shown in Figure 13.

Examples of how to drive the clock with an external oscillator are shown in Figure 14. Note that in the NMOS devices (8051, etc.) the signal at the XTAL2 pin actually drives the internal clock generator. In the CMOS devices (80C51, etc.), the signal at the XTAL1 pin drives the internal clock generator. The internal clock generator defines the sequence of states that make up the 80C51 machine cycle.

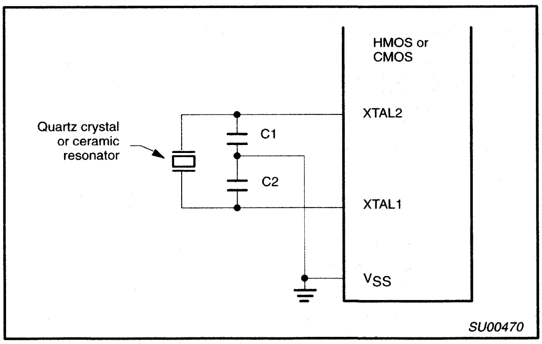


Figure 13. Using the On-Chip Oscillator

Table 8. Conditional Jumps in 80C51 Devices

MNEMONIC	OPERATION	ADDRESSING MODES				EXECUTION
		DIR	IND	REG	IMM	TIME (μs)
JZ rel	Jump if A = 0	Accumulator only				2
JNZ rel	Jump if A ≠ 0	Accumulator only				2
DJNZ <byte>,rel	Decrement and jump if not zero	X		X		2
CJNE A,<byte>,rel	Jump if A ≠ <byte>	X			X	2
CJNE <byte>,#data,rel	Jump if <byte> ≠ #data		X	X		2

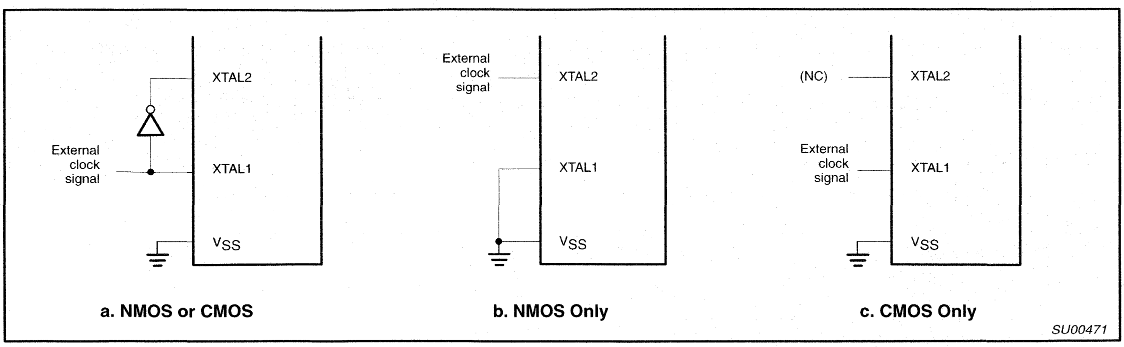


Figure 14. Using an External Clock

Machine Cycles

A machine cycle consists of a sequence of 6 states, numbered S1 through S6. Each state time lasts for two oscillator periods. Thus a machine cycle takes 12 oscillator periods or 1µs if the oscillator frequency is 12MHz.

Each state is divided into a Phase 1 half and a Phase 2 half. Figure 15 shows that fetch/execute sequences in states and phases for various kinds of instructions. Normally two program fetches are generated during each machine cycle, even if the instruction being executed doesn't require it. If the instruction being executed doesn't need more code bytes, the CPU simply ignores the extra fetch, and the Program Counter is not incremented.

Execution of a one-cycle instruction (Figures 15a and 15b) begins during State 1 of the machine cycle, when the opcode is latched into the Instruction Register. A second fetch occurs during S4 of the same machine cycle. Execution is complete at the end of State 6 of this machine cycle.

The MOVX instructions take two machine cycles to execute. No program fetch is generated during the second cycle of a MOVX instruction. This is the only time program fetches are skipped. The fetch/execute sequence for MOVX instructions is shown in Figure 15d.

The fetch/execute sequences are the same whether the Program Memory is internal or external to the chip. Execution times do not depend on whether the Program Memory is internal or external.

Figure 16 shows the signals and timing involved in program fetches when the Program Memory is external. If Program Memory is external, then the Program Memory read strobe $\overline{\text{PSEN}}$ is normally activated twice per machine cycle, as shown in Figure 16a. If an access to external Data Memory occurs, as shown in Figure 16b, two $\overline{\text{PSEN}}$s are skipped, because the address and data bus are being used for the Data Memory access.

Note that a Data Memory bus cycle takes twice as much time as a Program Memory bus cycle. Figure 16 shows the relative timing of the addresses being emitted at Ports 0 and 2, and of ALE and $\overline{\text{PSEN}}$. ALE is used to latch the low address byte from P0 into the address latch.

When the CPU is executing from internal Program Memory, PSEN is not activated, and program addresses are not emitted. However, ALE continues to be activated twice per machine cycle and so it is available as a clock output signal. Note, however, that one ALE is skipped during the execution of the MOVX instruction.

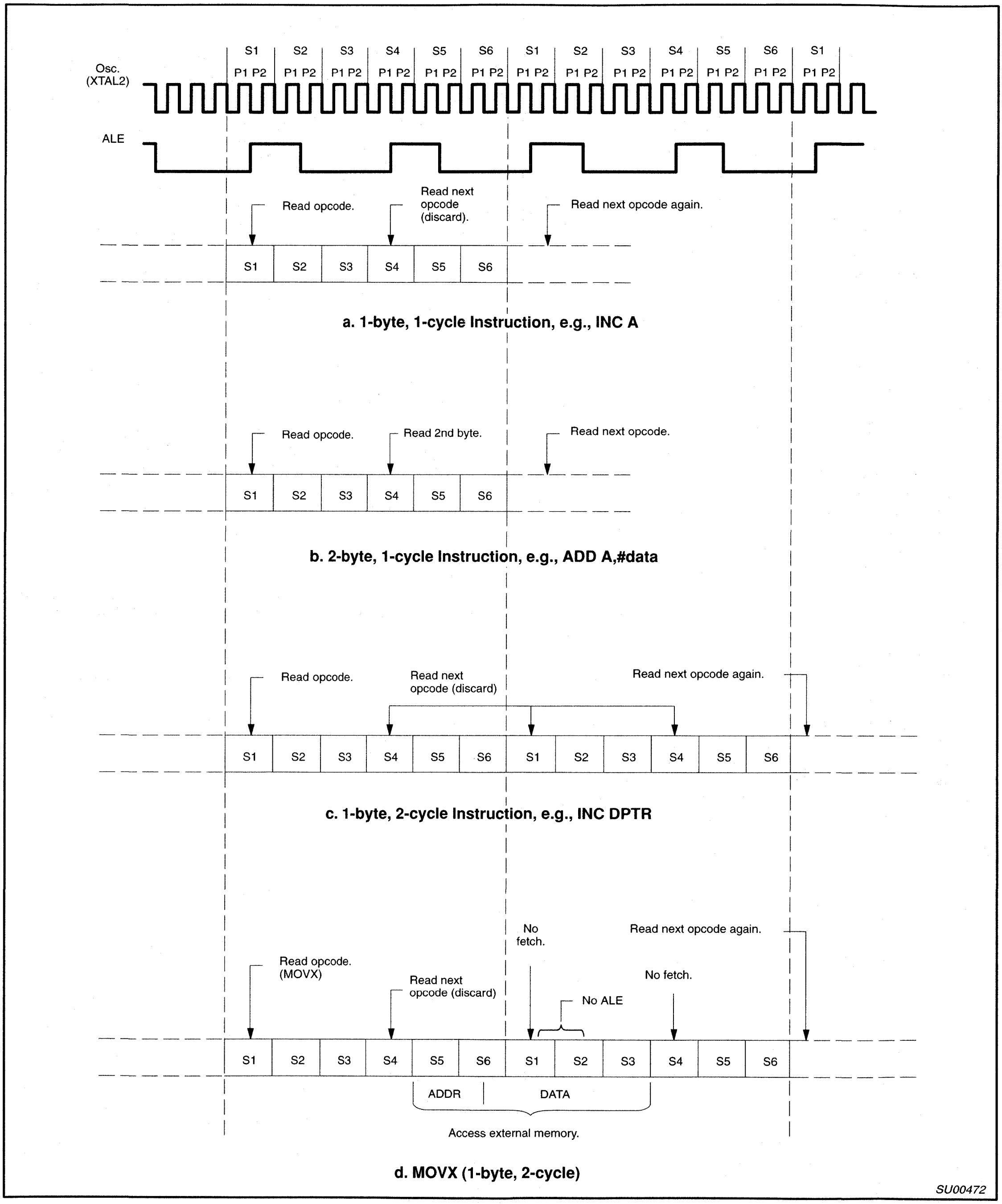

S1 S2 S3 S4 S5 S6 S1 S2 S3 S4 S5 S6 S1
P1 P2
Osc. (XTAL2)
ALE
Read opcode.
Read next opcode (discard).
Read next opcode again.
a. 1-byte, 1-cycle Instruction, e.g., INC A
Read 2nd byte.
Read next opcode.
b. 2-byte, 1-cycle Instruction, e.g., ADD A,#data
Read next opcode (discard)
c. 1-byte, 2-cycle Instruction, e.g., INC DPTR
Read opcode. (MOVX)
No fetch.
No ALE
ADDR
DATA
Access external memory.
d. MOVX (1-byte, 2-cycle)
SU00472

Figure 15. State Sequence in 80C51 Family Devices

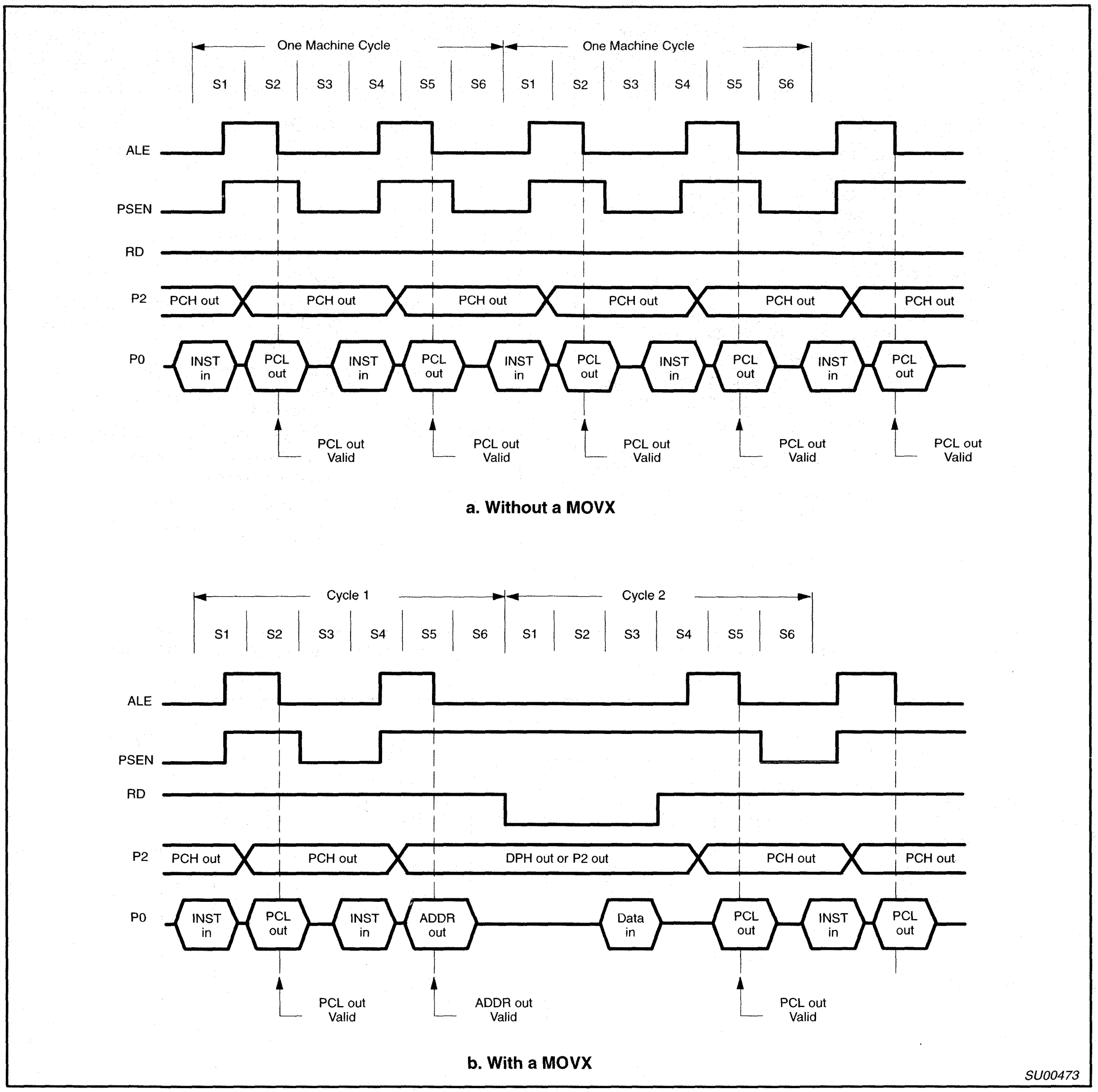


Figure 16. Bus Cycles in 80C51 Family Devices Executing from External Program Memory

(MSB) (LSB)

$\overline{EA}$	X	X	ES	ET1	EX1	ET0	EX0

Symbol	Position	Function
$\overline{EA}$	IE.7	Disables all interrupts. If $\overline{EA}$ = 0, no interrupt will be acknowledged. If $\overline{EA}$ = 1, each interrupt source is individually enabled or disabled by setting or clearing its enable bit.
	IE.6	Reserved.
	IE.5	Reserved.
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, External Interrupt 1 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 Exeternal Interrupt 0. If EX0 = 0, External Interrupt 0 is disabled.

SU00474

Figure 17. Interrupt Enable (IE) Register

(MSB) (LSB)

X	X	X	PS	PT1	PX1	PT0	PX0

Symbol	Position	Function
	IP.7	Reserved.
	IP.6	Reserved.
	IP.5	Reserved.
PS	IP.4	Defines the Serial Port interrupt priority level. PS = 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	Enables or disables the Timer 0 Interrupt priority level. PT) = 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.

SU00475

Figure 18. Interrupt Priority (IP) Register

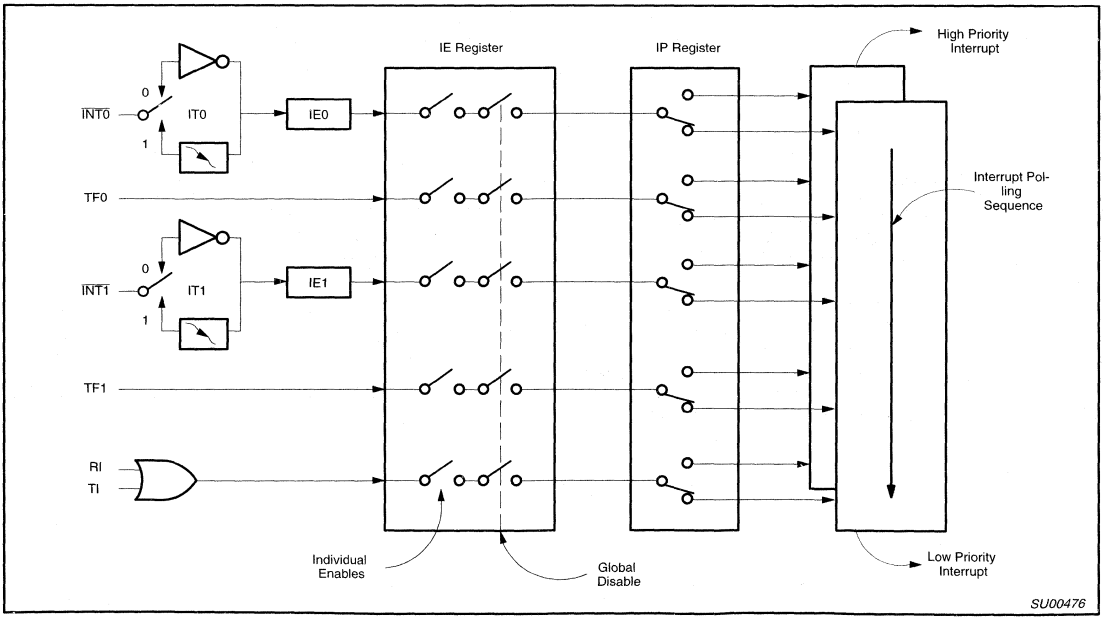


Figure 19. Interrupt Control System

Interrupt Structure

The 80C51 and its ROMless and EPROM versions have 5 interrupt sources: 2 external interrupts, 2 timer interrupts, and the serial port interrupt.

What follows is an overview of the interrupt structure for the device. More detailed information for specific members of the 80C51 derivative family is provided in later chapters of this user's guide.

Interrupt Enables

Each interrupt source can be individually enabled or disabled by setting or clearing a bit in the SFR named IE (Interrupt Enable). This register also contains a global disable bit, which can be cleared to disable all interrupts at once. Figure 17 shows the IE register.

Interrupt Priorities

Each interrupt source can also be individually programmed to one of two priority levels by setting or clearing a bit in the SFR named IP (Interrupt Priority). Figure 18 shows the IP register. A low-priority interrupt can be interrupted by a high-priority interrupt, but not by another low-priority interrupt. A high-priority interrupt can't be interrupted by any other interrupt source.

If two interrupt requests of different priority levels are received simultaneously, the request of higher priority is serviced. If interrupt requests of the same priority level are received simultaneously, an internal polling sequence determines which request is serviced. Thus within each priority level there is a second priority structure determined by the polling sequence. Figure 19 shows how the IE and IP registers and the polling sequence work to determine which if any interrupt will be serviced.

In operation, all the interrupt flags are latched into the interrupt control system during State 5 of every machine cycle. The samples are polled during the following machine cycle. If the flag for an enabled interrupt is found to be set (1), the interrupt system generates an LCALL to the appropriate location in Program Memory, unless some other condition blocks the interrupt. Several conditions can block an interrupt, among them that an interrupt of equal or higher priority level is already in progress.

The hardware-generated LCALL causes the contents of the Program Counter to be pushed into the stack, and reloads the PC with the beginning address of the service routine. As previously noted (Figure 3), the service routine for each interrupt begins at a fixed location.

Only the Program Counter is automatically pushed onto the stack, not the PSW or any other register. Having only the PC automatically saved allows the programmer to decide how much time should be spent saving other registers. This enhances the interrupt response time, albeit at the expense of increasing the programmer's burden of responsibility. As a result, many interrupt functions that are typical in control applications toggling a port pin for example, or reloading a timer, or unloading a serial buffer can often be completed in less time than it takes other architectures to complete.

Simulating a Third Priority Level in Software

Some applications require more than two priority levels that are provided by on-chip hardware in 80C51 devices. In these cases, relatively simple software can be written to produce the same effect as a third priority level. First, interrupts that are to have higher priority than 1 are assigned to priority 1 in the Interrupt Priority (IP) register. The service routines for priority 1 interrupts that are supposed to be interruptable by priority 2 interrupts are written to include the following code:

```
        PUSH  IE
        MOV   IE,#MASK
        CALL  LABEL
        ************************
        (execute service routine)
        *************************
        POP   IE
        RET
LABEL:        RETI
```

As soon as any priority interrupt is acknowledged, the Interrupt Enable (IE) register is redefined so as to disable all but priority 2 interrupts. Then a CALL to LABEL executes the RETI instruction, which clears the priority 1 interrupt-in-progress flip-flop. At this point any priority 1 interrupt that is enabled can be serviced, but only priority 2 interrupts are enabled.

POPing IE restores the original enable byte. Then a normal RET (rather than another RETI) is used to terminate the service routine. The additional software adds 10µs (at 12MHz) to priority 1 interrupts.

HARDWARE DESCRIPTION

This chapter provides a detailed description of the 80C51 microcontroller (see Figure 1). Included in this description are:

- The port drivers and how they function both as ports and, for Ports 0 and 2, in bus operations
- The Timers/Counters
- The Serial Interface
- The Interrupt System
- Reset
- The Reduced Power Modes in CMOS devices
- The EPROM version of the 80C51

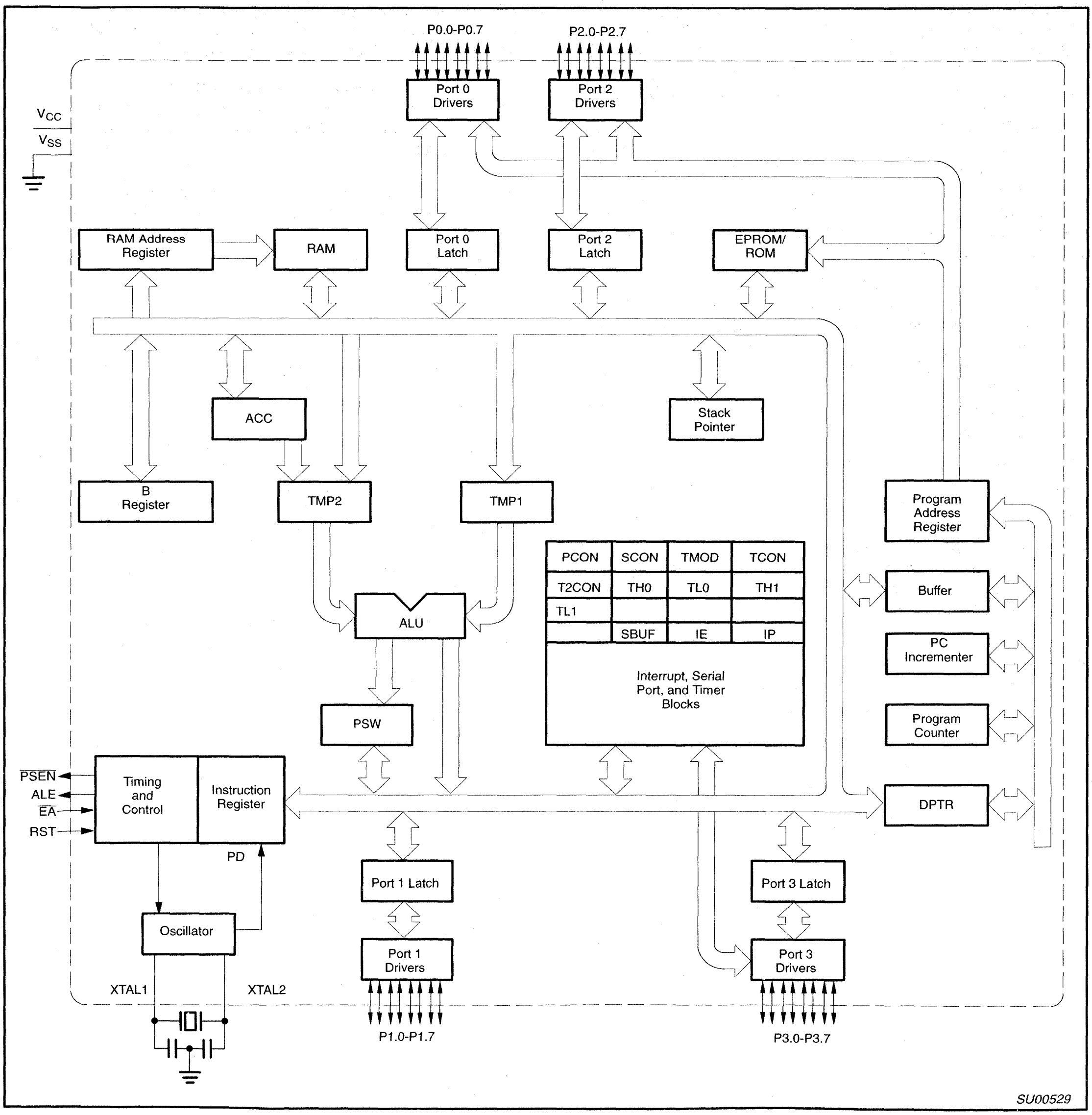


Figure 1. 80C51 Architecture

Special Function Registers

A Map of the on-chip memory area called the Special Function Register (SFR) space is shown in Figure 2.

Note that in the SFRs not all of the addresses are occupied. Unoccupied addresses are not implemented on the chip. Read accesses to these addresses will in general return random data, and write accesses will have no effect.

User software should not write 1s to these unimplemented locations, since they may be used in other 80C51 Family derivative products to invoke new features. The functions of the SFRs are described in the text that follows.

Accumulator

ACC is the Accumulator register. The mnemonics for Accumulator-Specific instructions, however, refer to the Accumulator simply as A.

B Register

The B register is used during multiply and divide operations. For other instructions it can be treated as another scratch pad register.

Program Status Word

The PSW register contains program status information as detailed in Figure 3.

Stack Pointer

The Stack Pointer register is 8 bits wide. It is incremented before data is stored during PUSH and CALL executions. While the stack may reside anywhere in on-chip RAM, the Stack Pointer is initialized to 07H after a reset. This causes the stack to begin at locations 08H.

Data Pointer

The Data Pointer (DPTR) consists of a high byte (DPH) and a low byte (DPL). Its intended function is to hold a 16-bit address. It may be manipulated as a 16-bit register or as two independent 8-bit registers.

Ports 0 to 3

P0, P1, P2, and P3 are the SFR latches of Ports 0, 1, 2, and 3, respectively. Writing a one to a bit of a port SFR (P0, P1, P2, or P3) causes the corresponding port output pin to switch high. Writing a zero causes the port output pin to switch low. When used as an input, the external state of a port pin will be held in the port SFR (i.e., if the external state of a pin is low, the corresponding port SFR bit will contain a 0; if it is high, the bit will contain a 1).

Serial Data Buffer

The Serial Buffer is actually two separate registers, a transmit buffer and a receive buffer. When data is moved to SBUF, it goes to the transmit buffer and is held for serial transmission. (Moving a byte to SBUF is what initiates the transmission.) When data is moved from SBUF, it comes from the receive buffer.

Timer Registers Basic to 80C51

Register pairs (TH0, TL0), and (TH1, TL1) are the 16-bit Counting registers for Timer/Counters 0 and 1, respectively.

Control Register for the 80C51

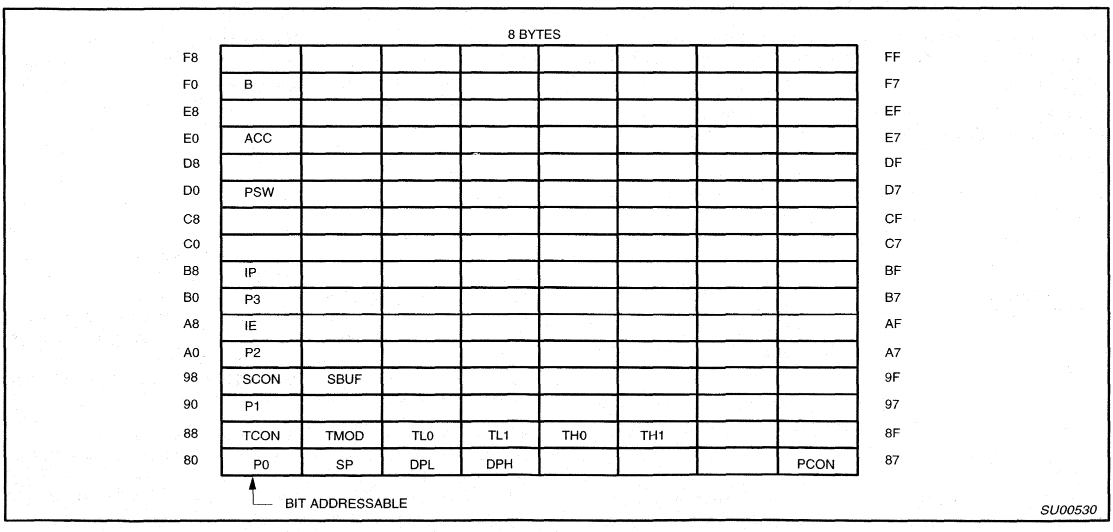
Special Function Registers IP, IE, TMOD, TCON, SCON, and PCON contain control and status bits for the interrupt system, the Timer/Counters, and the serial port. They are described in later sections.

Port Structures and Operation

All four ports in the 80C51 are bidirectional. Each consists of a latch (Special Function Registers P0 through P3), an output driver, and an input buffer.

The output drivers of Ports 0 and 2, and the input buffers of Port 0, are used in accesses to external memory. In this application, Port 0 outputs the low byte of the external memory address, time-multiplexed with the byte being written or read.

Port 2 outputs the high byte of the external memory address when the address is 16 bits wide. Otherwise, the Port 2 pins continue to emit the P2 SFR content.



	8 BYTES								
F8									FF
F0	B								F7
E8									EF
E0	ACC								E7
D8									DF
D0	PSW								D7
C8									CF
C0									C7
B8	IP								BF
B0	P3								B7
A8	IE								AF
A0	P2								A7
98	SCON	SBUF							9F
90	P1								97
88	TCON	TMOD	TL0	TL1	TH0	TH1			8F
80	P0	SP	DPL	DPH				PCON	87

BIT ADDRESSABLE
SU00530

Figure 2. 80C51 SFR Memory Map

MSB LSB

CY	AC	F0	RS1	RS0	OV	—	P

BIT	SYMBOL	FUNCTION
PSW.7	CY	Carry flag.
PSW.6	AC	Auxilliary Carry flag. (For BCD operations.)
PSW.5	F0	Flag 0. (Available to the user for general purposes.)
PSW.4	RS1	Register bank select control bit 1. Set/cleared by software to determine working register bank. (See Note.)
PSW.3	RS0	Register bank select control bit 0. Set/cleared by software todetermine working register bank. (See Note.)
PSW.2	OV	Overflow flag.
PSW.1	—	User-definable flag.
PSW.0	P	Parity flag. Set/cleared by hardware each instruction cycle to indicate an odd/even number of "one" bits in the Accumulator, i.e., even parity.

NOTE: The contents of (RS1, RS0) enable the working register banks as follows:

(0,0)—	Bank 0	(00H–07H)
(0,1)—	Bank 1	(08H–0fH)
(1,0)—	Bank 2	(10H–17H)
(1,1)—	Bank 3	(18H–17H)

SU00531A

Figure 3. Program Status Word (PSW) Register

All the Port 3 pins are multifunctional. They are not only port pins, but also serve the functions of various special features as listed below:

Port Pin	Alternate Function
P3.0	RxD (serial input port)
P3.1	TxD (serial output port)
P3.2	$\overline{\text{INT0}}$ (external interrupt)
P3.3	$\overline{\text{INT1}}$ (external interrupt)
P3.4	T0 (Timer/Counter 0 external input)
P3.5	T1 (Timer/Counter 1 external input)
P3.6	$\overline{\text{WR}}$ (external Data Memory write strobe)
P3.7	$\overline{\text{RD}}$ (external Data Memory read strobe)

The alternate functions can only be activated if the corresponding bit latch in the port SFR contains a 1. Otherwise the port pin remains at 0.

I/O Configurations

Figure 4 shows a functional diagram of a typical bit latch and I/O buffer in each of the four ports. The bit latch (one bit in the port's SFR) is represented as a Type D flip-flop, which will clock in a value from the internal bus in response to a "write to latch" signal from the CPU. The level of the port pin itself is placed on the internal bus in response to a "read pin" signal from the CPU. Some instructions that read a port activate the "read latch" signal, and others activate the "read pin" signal.

As shown in Figure 4, the output drivers of Port 0 and 2 are switchable to an internal ADDR and ADDR/DATA bus by an internal CONTROL signal for use in external memory accesses. During external memory accesses, the P2 SFR remains unchanged, but the P0 SFR gets 1s written to it.

Also shown in Figure 4 is that if a P3 bit latch contains a 1, then the output level is controlled by the signal labeled "alternate output function." The actual P3.X pin level is always available to the pin's alternate input function, if any.

Ports 1, 2, and 3 have internal pullups, and Port 0 has open drain outputs. Each I/O line can be independently used as an input or an output. (Port 0 and 2 may not be used as general purpose I/O when being used as the ADDR/DATA BUS for external memory during normal operation.) To be used as an input, the port bit latch must contain a 1, which turns off the output driver FET. Then, for Ports 1, 2, and 3, the pin is pulled high by a weak internal pullup, and can be pulled low by an external source.

Port 0 differs in that its internal pullups are not active during normal port operation. The pullup FET in the P0 output driver (see Figure 4) is used only when the port is emitting 1s during external memory accesses. Otherwise the pullup FET is off. Consequently P0 lines that are being used as output port lines are open drain. Writing a 1 to the bit latch leaves both output FETs off, so the pin floats. In that condition it can be used as a high-impedance input.

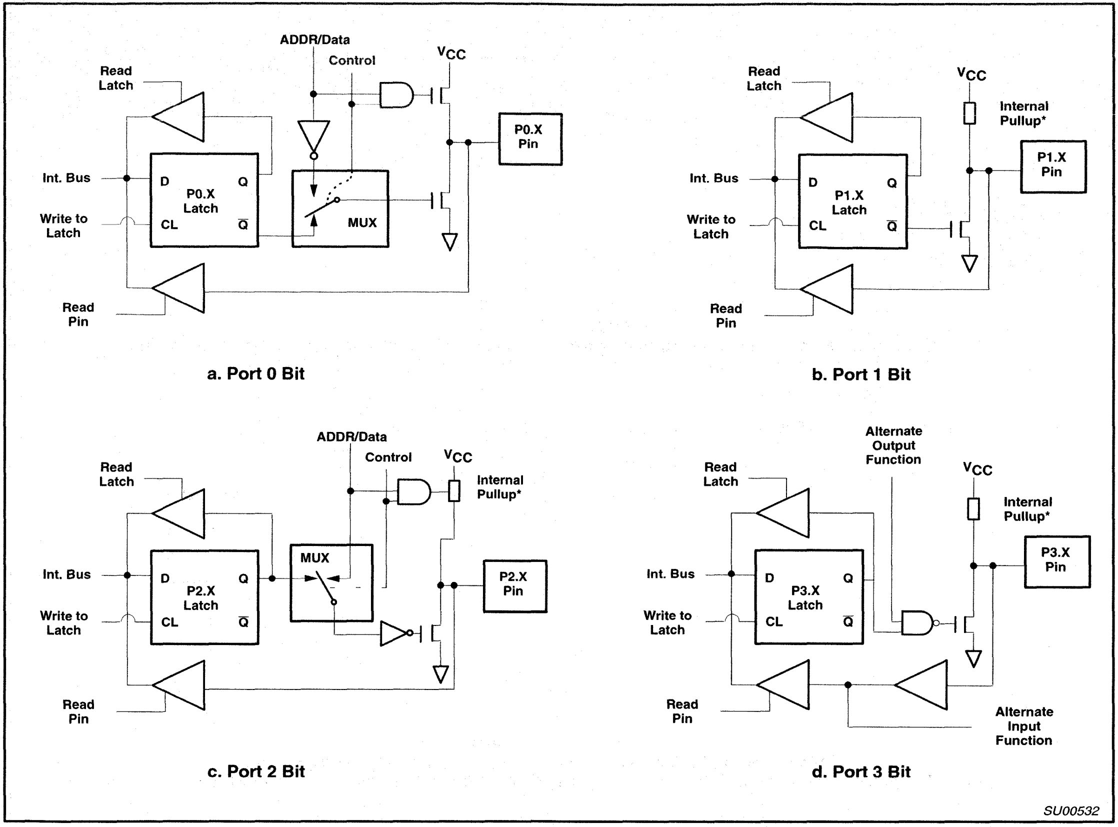
Because Ports 1, 2, and 3 have fixed internal pullups, they are sometimes called "quasi- bidirectional" ports. When configured as inputs they pull high and will source current (I_{IL}, in the data sheets) when externally pulled low. Port 0, on the other hand, is considered "true" bidirectional, because when configured as an input it floats.

All the port latches in the 80C51 have 1s written to them by the reset function. If a 0 is subsequently written to a port latch, it can be reconfigured as an input by writing a 1 to it.

Writing to a Port

In the execution of an instruction that changes the value in a port latch, the new value arrives at the latch during S6P2 of the final cycle of the instruction. However, port latches are in fact sampled by their output buffers only during Phase 1 of an clock period. (During Phase 2 the output buffer holds the value it saw during the previous Phase 1). Consequently, the new value in the port latch won't actually appear at the output pin until the next Phase 1, which will be at S1P1 of the next machine cycle.

If the change requires a 0-to-1 transition in Port 1, 2, or 3, an additional pullup is turned on during S1P1 and S1P2 of the cycle in which the transition occurs. This is done to increase the transition speed. The extra pullup can source about 100 times the current that the normal pullup can. It should be noted that the internal pullups are field-effect transistors, not linear resistors. The pullup arrangements are shown in Figure 5.



*See Figure 5 for details of the internal pullup.

Figure 4. 80C51 Port Bit Latches and I/O Buffers

In the NMOS 8051 part, the fixed part of the pullup is a depletion mode transistor with the gate wired to the source. This transistor will allow the pin to source about 0.25mA when shorted to ground. In parallel with the fixed pullup is an enhancement mode transistor, which is activated during S1 whenever the port bit does a 0-to-1 transition. During this interval, if the port pin is shorted to ground, this extra transistor will allow the pin to source an additional 30mA.

In the CMOS 80C51, the pullup consists of three pFETs. It should be noted that an n-channel FET (nFET) is turned on when a logical 1 is applied to its gate, and is turned off when a logical 0 is applied to its gate. A p-channel FET (pFET) is the opposite: it is on when its gate sees a 0, and off when its gate sees a 1.

pFET1 in Figure 5 is the transistor that is turned on for 2 oscillator periods after a 0-to-1 transition in the port latch. While it's on, it turns on pFET3 (a weak pullup), through the inverter. This inverter and pFET3 form a latch which holds the 1.

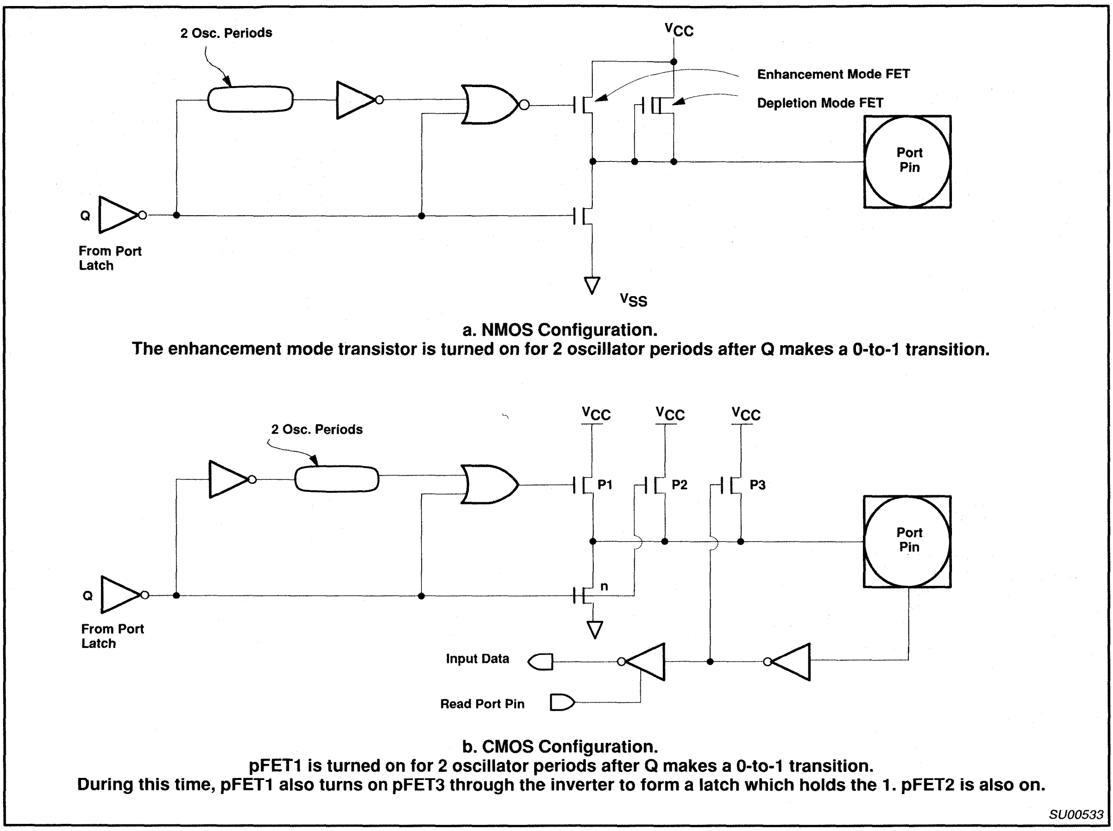
Note that if the pin is emitting a 1, a negative glitch on the pin from some external source can turn off pFET3, causing the pin to go into a float state. pFET2 is a very weak pullup which is on whenever the nFET is off, in traditional CMOS style. It's only about 1/10 the strength of pFET1. Its function is to restore a 1 to the pin in the event the pin had a 1 and lost it to a glitch.

Port Loading and Interfacing

The output buffers of Ports 1, 2, and 3 can each drive 4 LS TTL inputs. These ports on NMOS versions can be driven in a normal manner by a TTL or NMOS circuit. Both NMOS and CMOS pins can be driven by open-collector and open-drain outputs, but note that 0-to-1 transitions will not be fast.

In the NMOS device, if the pin is driven by an open-collector output, a 0-to-1 transition will have to be driven by the relatively weak depletion mode FET in Figure 5a. In the CMOS device, an input 0 turns off pullup pFET3, leaving only the very weak pullup pFET2 to drive the transition.

Port 0 output buffers can each drive 8 LS TTL inputs. They do, however, require external pullups to drive NMOS inputs, except when being used as the ADDRESS/DATA bus for external memory.



a. NMOS Configuration.
The enhancement mode transistor is turned on for 2 oscillator periods after Q makes a 0-to-1 transition.

b. CMOS Configuration.
pFET1 is turned on for 2 oscillator periods after Q makes a 0-to-1 transition.
During this time, pFET1 also turns on pFET3 through the inverter to form a latch which holds the 1. pFET2 is also on.

Figure 5. Ports 1 and 3 NMOS and CMOS Internal Pullup Configurations
(Port 2 is similar except that it holds the strong pullup on while emitting 1s that are address bits. See *Accessing External Memory.*)

Read-Modify-Write Feature

Some instructions that read a port read the latch and others read the pin. Which ones do which? The instructions that read the latch rather than the pin are the ones that read a value, possibly change it, and then rewrite it to the latch. These are called "read-modify-write" instructions. The instructions listed below are read-modify-write instructions. When the destination operand is a port, or a port bit, these instructions read the latch rather than the pin:

ANL	(logical AND, e.g., ANL P1,A)
ORL	(logical OR, e.g., ORL P2,A)
XRL	(logical EX-OR, e.g., XRL P3,A)
JBC	(jump if bit = 1 and clear bit, e.g., JBC P1.1,LABEL)
CPL	(complement bit, e.g., CPL P3.0)
INC	(increment, e.g., INC P2)
DEC	(decrement, e.g., DEC P2)
DJNZ	(decrement and jump if not zero, e.g., DJNZ P3,LABEL)
MOV PX.Y,C	(move carry bit to bit Y of Port X)
CLR PX.Y	(clear bit Y of Port X)
SET PX.Y	(set bit Y of Port X)

It is not obvious that the last three instructions in this list are read-modify-write instructions, but they are. They read the port byte, all 8 bits, modify the addressed bit, then write the new byte back to the latch.

The reason that read-modify-write instructions are directed to the latch rather than the pin is to avoid a possible misinterpretation of the voltage level at the pin. For example, a port bit might be used to drive the base of a transistor. When a 1 is written to the bit, the transistor is turned on. If the CPU then reads the same port bit at the pin rather than the latch, it will read the base voltage of the transistor and interpret it as a 0. Reading the latch rather than the pin will return the correct value of 1.

Accessing External Memory

Accesses to external memory are of two types: accesses to external Program Memory and accesses to external Data Memory. Accesses to external Program Memory use signal $\overline{PSEN}$ (program store enable) as the read strobe. Accesses to external Data Memory use $\overline{RD}$ or $\overline{WR}$ (alternate functions of P3.7 and P3.6) to strobe the memory. Fetches from external Program Memory always use a 16-bit address. Accesses to external Data Memory can use either a 16-bit address (MOVX @ DPTR) or an 8-bit address (MOVX @Ri).

Whenever a 16-bit address is used, the high byte of the address comes out on Port 2, where it is held for the duration of the read or write cycle. Note that the Port 2 drivers use the strong pullups during the entire time that they are emitting address bits that are 1s. This is during the execution of a MOVX @DPTR instruction. During this time the Port 2 latch (the Special Function Register) does not have to contain 1s, and the contents of the Port 2 SFR are not modified. If the external memory cycle is not immediately followed by another external memory cycle, the undisturbed contents of the Port 2 SFR will reappear in the next cycle.

If an 8-bit address is being used (MOVX @Ri), the contents of the Port 2 SFR remain at the Port 2 pins throughout the external memory cycle. This will facilitate paging.

In any case, the low byte of the address is time-multiplexed with the data byte on Port 0. The ADDR/DATA signals drive both FETs in the Port 0 output buffers. Thus, in this application the Port 0 pins are not open-drain outputs, and do not require external pullups. ALE (Address Latch Enable) should be used to capture the address byte into an external latch. The address byte is valid at the negative transition of ALE. Then, in a write cycle, the data byte to be written appears on Port 0 just before $\overline{WR}$ is activated, and remains there until after $\overline{WR}$ is deactivated. In a read cycle, the incoming byte is accepted at Port 0 just before the read strobe is deactivated.

During any access to external memory, the CPU writes 0FFH to the Port 0 latch (the Special Function Register), thus obliterating whatever information the Port 0 SFR may have been holding.

External Program Memory is accessed under two conditions: Whenever signal $\overline{EA}$ is active; or whenever the program counter (PC) contains a number that is larger than 0FFFH (in the 80C51).

This require that the ROMless versions have $\overline{EA}$ wired low to enable the lower 4k program bytes to be fetched from external memory.

When the CPU is executing out of external Program Memory, all 8 bits of Port 2 are dedicated to an output function and may not be used for general purpose I/O. During external program fetches they output the high byte of the PC. During this time the Port 2 drivers use the strong pullups to emit PC bits that are 1s.

Timer/Counters

The 80C51 has two 16-bit Timer/Counter registers: Timer 0 and Timer 1. Both can be configured to operate either as timers or event counters (see Figure 6).

In the "Timer" function, the register is incremented every machine cycle. Thus, one can think of it as counting machine cycles. Since a machine cycle consists of 12 oscillator periods, the count rate is 1/12 of the oscillator frequency.

In the "Counter" function, the register is incremented in response to a 1-to-0 transition at its corresponding external input pin, T0 or T1. In this function, the external input is sampled during S5P2 of every machine cycle.

When the samples show a high in one cycle and a low in the next cycle, the count is incremented. The new count value appears in the register during S3P1 of the cycle following the one in which the transition was detected. Since it takes 2 machine cycles (24 oscillator periods) to recognize a 1-to-0 transition, the maximum count rate is 1/24 of the oscillator frequency. There are no restrictions on the duty cycle of the external input signal, but to ensure that a given level is sampled at least once before it changes, it should be held for at least one full cycle. In addition to the "Timer" or "Counter" selection, Timer 0 and Timer 1 have four operating modes from which to select.

Timer 0 and Timer 1

The "Timer" or "Counter" function is selected by control bits C/T in the Special Function Register TMOD. These two Timer/Counters have four operating modes, which are selected by bit-pairs (M1, M0) in TMOD. Modes 0, 1, and 2 are the same for both Timers/Counters. Mode 3 is different. The four operating modes are described in the following text.

Mode 0

Putting either Timer into Mode 0 makes it look like an 8048 Timer, which is an 8-bit Counter with a divide-by-32 prescaler. Figure 7 shows the Mode 0 operation as it applies to Timer 1.

In this mode, the Timer register is configured as a 13-bit register. As the count rolls over from all 1s to all 0s, it sets the Timer interrupt flag TF1. The counted input is enabled to the Timer when TR1 = 1 and either GATE = 0 or $\overline{INT1}$ = 1. (Setting GATE = 1 allows the Timer to be controlled by external input $\overline{INT1}$, to facilitate pulse width measurements). TR1 is a control bit in the Special Function Register TCON (Figure 8). GATE is in TMOD.

The 13-bit register consists of all 8 bits of TH1 and the lower 5 bits of TL1. The upper 3 bits of TL1 are indeterminate and should be ignored. Setting the run flag (TR1) does not clear the registers.

Mode 0 operation is the same for the Timer 0 as for Timer 1. Substitute TR0, TF0, and $\overline{INT0}$ for the corresponding Timer 1 signals in Figure 7. There are two different GATE bits, one for Timer 1 (TMOD.7) and one for Timer 0 (TMOD.3).

Mode 1

Mode 1 is the same as Mode 0, except that the Timer register is being run with all 16 bits.

Mode 2

Mode 2 configures the Timer register as an 8-bit Counter (TL1) with automatic reload, as shown in Figure 9. Overflow from TL1 not only sets TF1, but also reloads TL1 with the contents of TH1, which is preset by software. The reload leaves TH1 unchanged.

Mode 2 operation is the same for Timer/Counter 0.

Mode 3

Timer 1 in Mode 3 simply holds its count. The effect is the same as setting TR1 = 0.

Timer 0 in Mode 3 establishes TL0 and TH0 as two separate counters. The logic for Mode 3 on Timer 0 is shown in Figure 10. TL0 uses the Timer 0 control bits: C/T, GATE, TR0, INT0, and TF0. TH0 is locked into a timer function (counting machine cycles) and takes over the use of TR1 and TF1 from Timer 1. Thus, TH0 now controls the "Timer 1" interrupt.

Mode 3 is provided for applications requiring an extra 8-bit timer on the counter. With Timer 0 in Mode 3, an 80C51 can look like it has three Timer/Counters. When Timer 0 is in Mode 3, Timer 1 can be turned on and off by switching it out of and into its own Mode 3, or can still be used by the serial port as a baud rate generator, or in fact, in any application not requiring an interrupt.

MSB LSB

GATE	C/$\overline{T}$	M1	M0	GATE	C/$\overline{T}$	M1	M0
TIMER 1				**TIMER 0**			

GATE Gating control when set. Timer/Counter “x” is enabled only while “$\overline{\text{INTx}}$” pin is high and “TRx” control pin is set. when cleared Timer “x” is enabled whenever “TRx” control bit is set.

C/$\overline{T}$ Timer or Counter Selector cleared for Timer operation (input from in=ternal system clock.) Set for Counter operation (input from “Tx” input pin).

M1	**M0**	**OPERATING**
0	0	8048 Timer “TLx” serves as 5-bit prescaler.
0	1	16-bit Timer/Counter “THx” and “TLx” are cascaded; there is no prescaler.
1	0	8-bit auto-reload Timer/Counter “THx” holds a value which is to be reloaded into “TLx” each time it overflows.
1	1	(Timer 0) TL0 is an 8-bit Timer/Counter controlled by the standard Timer 0 control bits. TH0 is an 8-bit timer only controlled by Timer 1 control bits.
1	1	(Timer 1) Timer/Counter 1 stopped.

SU00534

Figure 6. Timer/Counter Mode Control (TMOD) Register

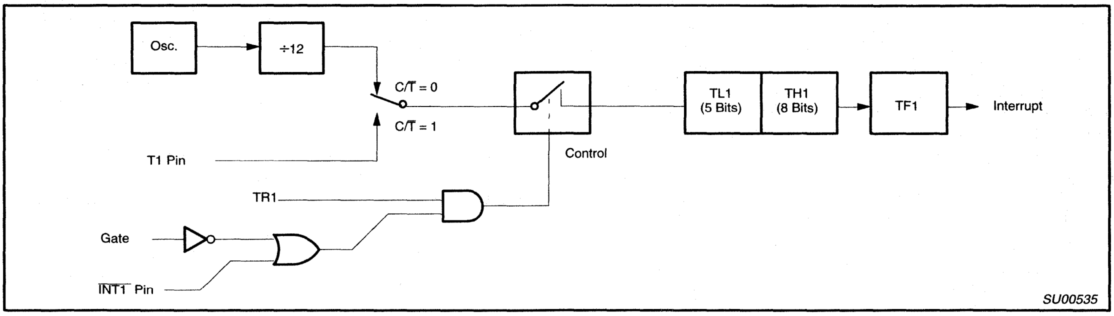


Figure 7. Timer/Counter Mode 0: 13-Bit Counter

MSB LSB

TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

BIT	**SYMBOL**	**FUNCTION**
TCON.7	TF1	Timer 1 overflow flag. Set by hardware on Timer/Counter overflow. Cleared by hardware when processor vectors to interrupt routine, or clearing the bit in software.
TCON.6	TR1	Timer 1 Run control bit. Set/cleared by software to turn Timer/Counter on/off.
TCON.5	TF0	Timer 0 overflow flag. Set by hardware on Timer/Counter overflow. Cleared by hardware when processor vectors to interrupt routine, or by clearing the bit in software.
TCON.4	TR0	Timer 0 Run control bit. Set/cleared by software to turn Timer/Counter on/off.
TCON.3	IE1	Interrupt 1 Edge flag. Set by hardware when external interrupt edge detected. Cleared when interrupt processed.
TCON.2	IT1	Interrupt 1 type control bit. Set/cleared by software to specify falling edge/low level triggered external interrupts.
TCON.1	IE0	Interrupt 0 Edge flag. Set by hardware when external interrupt edge detected. Cleared when interrupt processed.
TCON.0	IT0	Interrupt 0 Type control bit. Set/cleared by software to specify falling edge/low level triggered external interrupts.

SU00536

Figure 8. Timer/Counter Control (TCON) Register

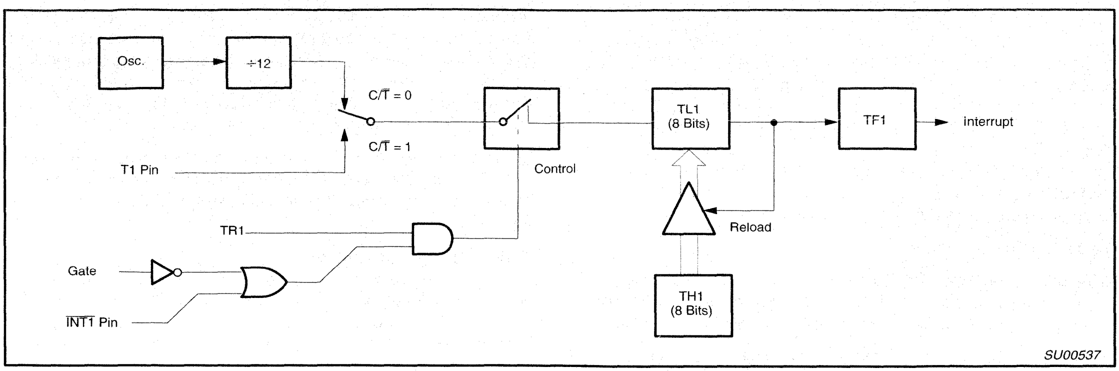


Figure 9. Timer/Counter Mode 2: 8-Bit Auto-Load

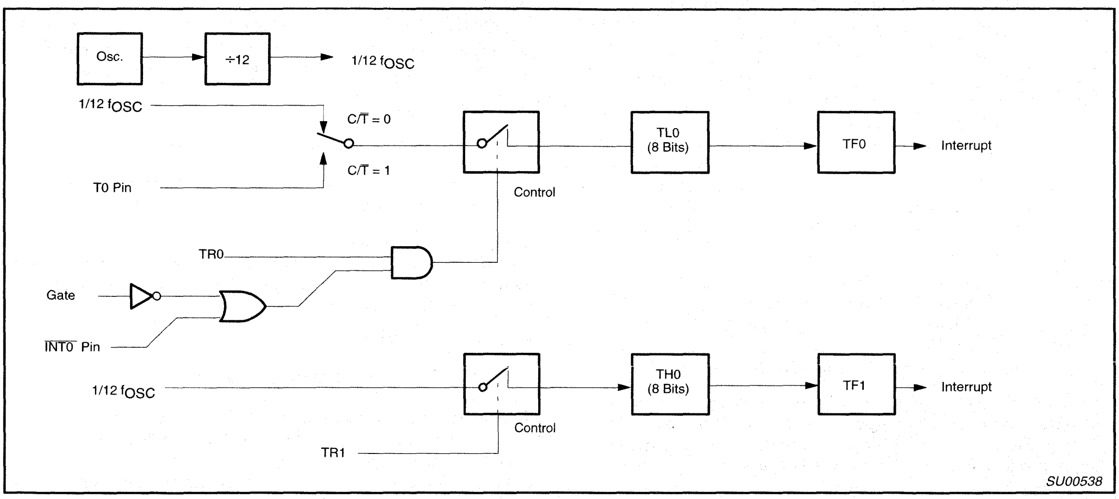


Figure 10. Timer/Counter 0 Mode 3: Two 8-Bit Counters

Standard Serial Interface

The serial port is full duplex, meaning it can transmit and receive simultaneously. It is also receive-buffered, meaning it can commence reception of a second byte before a previously received byte has been read from the register. (However, if the first byte still hasn't been read by the time reception of the second byte is complete, one of the bytes will be lost.) The serial port receive and transmit registers are both accessed at Special Function Register SBUF. Writing to SBUF loads the transmit register, and reading SBUF accesses a physically separate receive register.

The serial port can operate in 4 modes:

Mode 0: Serial data enters and exits through RxD. TxD outputs the shift clock. 8 bits are transmitted/received (LSB first). The baud rate is fixed at 1/12 the oscillator frequency.

Mode 1: 10 bits are transmitted (through TxD) or received (through RxD): a start bit (0), 8 data bits (LSB first), and a stop bit (1). On receive, the stop bit goes into RB8 in Special Function Register SCON. The baud rate is variable.

Mode 2: 11 bits are transmitted (through TxD) or received (through RxD): start bit (0), 8 data bits (LSB first), a programmable 9th data bit, and a stop bit (1). On Transmit, the 9th data bit (TB8 in SCON) can be assigned the value of 0 or 1. Or, for example, the parity bit (P, in the PSW) could be moved into TB8. On receive, the 9th data bit goes into RB8 in Special Function Register SCON, while the stop bit is ignored. The baud rate is programmable to either 1/32 or 1/64 the oscillator frequency.

Mode 3: 11 bits are transmitted (through TxD) or received (through RxD): a start bit (0), 8 data bits (LSB first), a programmable 9th data bit, and a stop bit (1). In fact, Mode 3 is the same as Mode 2 in all respects except baud rate. The baud rate in Mode 3 is variable.

In all four modes, transmission is initiated by any instruction that uses SBUF as a destination register. Reception is initiated in Mode 0 by the condition RI = 0 and REN = 1. Reception is initiated in the other modes by the incoming start bit if REN = 1.

Multiprocessor Communications

Modes 2 and 3 have a special provision for multiprocessor communications. In these modes, 9 data bits are received. The 9th one goes into RB8. Then comes a stop bit. The port can be programmed such that when the stop bit is received, the serial port interrupt will be activated only if RB8 = 1. This feature is enabled by setting bit SM2 in SCON. A way to use this feature in multiprocessor systems is as follows:

When the master processor wants to transmit a block of data to one of several slaves, it first sends out an address byte which identifies the target slave. An address byte differs from a data byte in that the 9th bit is 1 in an address byte and 0 in a data byte. With SM2 = 1, no slave will be interrupted by a data byte. An address byte, however, will interrupt all slaves, so that each slave can examine the received byte and see if it is being addressed. The addressed slave will clear its SM2 bit and prepare to receive the data bytes that will be coming. The slaves that weren't being addressed leave their SM2s set and go on about their business, ignoring the coming data bytes.

SM2 has no effect in Mode 0, and in Mode 1 can be used to check the validity of the stop bit. In a Mode 1 reception, if SM2 = 1, the receive interrupt will not be activated unless a valid stop bit is received.

Serial Port Control Register

The serial port control and status register is the Special Function Register SCON, shown in Figure 11. This register contains not only the mode selection bits, but also the 9th data bit for transmit and receive (TB8 and RB8), and the serial port interrupt bits (TI and RI).

Baud Rates

The baud rate in Mode 0 is fixed: Mode 0 Baud Rate = Oscillator Frequency / 12. The baud rate in Mode 2 depends on the value of bit SMOD in Special Function Register PCON. If SMOD = 0 (which is the value on reset), the baud rate is 1/64 the oscillator frequency. If SMOD = 1, the baud rate is 1/32 the oscillator frequency.

Mode 2 Baud Rate =

$$\frac{2^{SMOD}}{64} \times \text{(Oscillator Frequency)}$$

In the 80C51, the baud rates in Modes 1 and 3 are determined by the Timer 1 overflow rate.

Using Timer 1 to Generate Baud Rates

When Timer 1 is used as the baud rate generator, the baud rates in Modes 1 and 3 are determined by the Timer 1 overflow rate and the value of SMOD as follows:

Mode 1, 3 Baud Rate =

$$\frac{2^{SMOD}}{32} \times \text{(Timer 1 Overflow Rate)}$$

The Timer 1 interrupt should be disabled in this application. The Timer itself can be configured for either "timer" or "counter" operation, and in any of its 3 running modes. In the most typical applications, it is configured for "timer" operation, in the auto-reload mode (high nibble of TMOD = 0010B). In that case the baud rate is given by the formula:

Mode 1, 3 Baud Rate =

$$\frac{2^{SMOD}}{32} \times \frac{\text{Oscillator Frequency}}{12 \times [256 - (TH1)]}$$

One can achieve very low baud rates with Timer 1 by leaving the Timer 1 interrupt enabled, and configuring the Timer to run as a 16-bit timer (high nibble of TMOD = 0001B), and using the Timer 1 interrupt to do a 16-bit software reload. Figure 12 lists various commonly used baud rates and how they can be obtained from Timer 1.

MSB							LSB
SM0	SM1	SM2	REN	TB8	RB8	TI	RI

Where SM0, SM1 specify the serial port mode, as follows:

SM0	SM1	Mode	Description	Baud Rate
0	0	0	shift register	$f_{OSC}/12$
0	1	1	8-bit UART	variable
1	0	2	9-bit UART	$f_{OSC}/64$ or $f_{OSC}/32$
1	1	3	9-bit UART	variable

SM2 Enables the multiprocessor communication feature in Modes 2 and 3. In Mode 2 or 3, if SM2 is set to 1, then RI will not be activated if the received 9th data bit (RB8) is 0. In Mode 1, if SM2=1 then RI will not be activated if a valid stop bit was not received. In Mode 0, SM2 should be 0.

REN Enables serial reception. Set by software to enable reception. Clear by software to disable reception.

TB8 The 9th data bit that will be transmitted in Modes 2 and 3. Set or clear by software as desired.

RB8 In Modes 2 and 3, is the 9th data bit that was received. In Mode 1, it SM2=0, RB8 is the stop bit that was received. In Mode 0, RB8 is not used.

TI Transmit interrupt flag. Set by hardware at the end of the 8th bit time in Mode 0, or at the beginning of the stop bit in the other modes, in any serial transmission. Must be cleared by software.

RI Receive interrupt flag. Set by hardware at the end of the 8th bit time in Mode 0, or halfway through the stop bit time in the other modes, in any serial reception (except see SM2). Must be cleared by software.

SU00120

Figure 11. Serial Port Control (SCON) Register

Baud Rate	f_{OSC}	SMOD	Timer 1		
			C/$\overline{T}$	Mode	Reload Value
Mode 0 Max: 1.67MHz	20MHz	X	X	X	X
Mode 2 Max: 625k	20MHz	1	X	X	X
Mode 1, 3 Max: 104.2k	20MHz	1	0	2	FFH
19.2k	11.059MHz	1	0	2	FDH
9.6k	11.059MHz	0	0	2	FDH
4.8k	11.059MHz	0	0	2	FAH
2.4k	11.059MHz	0	0	2	F4H
1.2k	11.059MHz	0	0	2	E8H
137.5	11.986MHz	0	0	2	1DH
110	6MHz	0	0	2	72H
110	12MHz	0	0	1	FEEBH

Figure 12. Timer 1 Generated Commonly Used Baud Rates

More About Mode 0

Serial data enters and exits through RxD. TxD outputs the shift clock. 8 bits are transmitted/received: 8 data bits (LSB first). The baud rate is fixed a 1/12 the oscillator frequency.

Figure 13 shows a simplified functional diagram of the serial port in Mode 0, and associated timing.

Transmission is initiated by any instruction that uses SBUF as a destination register. The "write to SBUF" signal at S6P2 also loads a 1 into the 9th position of the transmit shift register and tells the TX Control block to commence a transmission. The internal timing is such that one full machine cycle will elapse between "write to SBUF" and activation of SEND.

SEND enables the output of the shift register to the alternate output function line of P3.0 and also enable SHIFT CLOCK to the alternate output function line of P3.1. SHIFT CLOCK is low during S3, S4, and S5 of every machine cycle, and high during S6, S1, and S2. At S6P2 of every machine cycle in which SEND is active, the contents of the transmit shift are shifted to the right one position.

As data bits shift out to the right, zeros come in from the left. When the MSB of the data byte is at the output position of the shift register, then the 1 that was initially loaded into the 9th position, is just to the left of the MSB, and all positions to the left of that contain zeros. This condition flags the TX Control block to do one last shift and then deactivate SEND and set T1. Both of these actions occur at S1P1 of the 10th machine cycle after "write to SBUF."

Reception is initiated by the condition REN = 1 and R1 = 0. At S6P2 of the next machine cycle, the RX Control unit writes the bits 11111110 to the receive shift register, and in the next clock phase activates RECEIVE.

RECEIVE enable SHIFT CLOCK to the alternate output function line of P3.1. SHIFT CLOCK makes transitions at S3P1 and S6P1 of every machine cycle. At S6P2 of every machine cycle in which RECEIVE is active, the contents of the receive shift register are shifted to the left one position. The value that comes in from the right is the value that was sampled at the P3.0 pin at S5P2 of the same machine cycle.

As data bits come in from the right, 1s shift out to the left. When the 0 that was initially loaded into the rightmost position arrives at the leftmost position in the shift register, it flags the RX Control block to do one last shift and load SBUF. At S1P1 of the 10th machine cycle after the write to SCON that cleared RI, RECEIVE is cleared as RI is set.

More About Mode 1

Ten bits are transmitted (through TxD), or received (through RxD): a start bit (0), 8 data bits (LSB first), and a stop bit (1). On receive, the stop bit goes into RB8 in SCON. In the 80C51 the baud rate is determined by the Timer 1 overflow rate.

Figure 14 shows a simplified functional diagram of the serial port in Mode 1, and associated timings for transmit receive.

Transmission is initiated by any instruction that uses SBUF as a destination register. The "write to SBUF" signal also loads a 1 into the 9th bit position of the transmit shift register and flags the TX Control unit that a transmission is requested. Transmission actually commences at S1P1 of the machine cycle following the next rollover in the divide-by-16 counter. (Thus, the bit times are synchronized to the divide-by-16 counter, not to the "write to SBUF" signal.)

The transmission begins with activation of SEND which puts the start bit at TxD. One bit time later, DATA is activated, which enables the output bit of the transmit shift register to TxD. The first shift pulse occurs one bit time after that.

As data bits shift out to the right, zeros are clocked in from the left. When the MSB of the data byte is at the output position of the shift register, then the 1 that was initially loaded into the 9th position is just to the left of the MSB, and all positions to the left of that contain zeros. This condition flags the TX Control unit to do one last shift and then deactivfate SEND and set TI. This occurs at the 10th divide-by-16 rollover after "write to SBUF."

Reception is initiated by a detected 1-to-0 transition at RxD. For this purpose RxD is sampled at a rate of 16 times whatever baud rate has been established. When a transition is detected, the divide-by-16 counter is immediately reset, and 1FFH is written into the input shift register. Resetting the divide-by-16 counter aligns its rollovers with the boundaries of the incoming bit times.

The 16 states of the counter divide each bit time into 16ths. At the 7th, 8th, and 9th counter states of each bit time, the bit detector samples the value of RxD. The value accepted is the value that was seen in at least 2 of the 3 samples. This is done for noise rejection. If the value accepted during the first bit time is not 0, the receive circuits are reset and the unit goes back to looking for another 1-to-0 transition. This is to provide rejection of false start bits. If the start bit proves valid, it is shifted into the input shift register, and reception of the rest of the frame will proceed.

As data bits come in from the right, 1s shift out to the left. When the start bit arrives at the leftmost position in the shift register (which in mode 1 is a 9-bit register), it flags the RX Control block to do one last shift, load SBUF and RB8, and set RI. The signal to load SBUF and RB8, and to set RI, will be generated if, and only if, the following conditions are met at the time the final shift pulse is generated.:

1. R1 = 0, and
2. Either SM2 = 0, or the received stop bit = 1.

If either of these two conditions is not met, the received frame is irretrievably lost. If both conditions are met, the stop bit goes into RB8, the 8 data bits go into SBUF, and RI is activated. At this time, whether the above conditions are met or not, the unit goes back to looking for a 1-to-0 transition in RxD.

More About Modes 2 and 3

Eleven bits are transmitted (through TxD), or received (through RxD): a start bit (0), 8 data bits (LSB first), a programmable 9th data bit, and a stop bit (1). On transmit, the 9th data bit (TB8) can be assigned the value of 0 or 1. On receive, the 9the data bit goes into RB8 in SCON. The baud rate is programmable to either 1/32 or 1/64 the oscillator frequency in Mode 2. Mode 3 may have a variable baud rate generated from Timer 1.

Figures 15 and 16 show a functional diagram of the serial port in Modes 2 and 3. The receive portion is exactly the same as in Mode 1. The transmit portion differs from Mode 1 only in the 9th bit of the transmit shift register.

Transmission is initiated by any instruction that uses SBUF as a destination register. The "write to SBUF" signal also loads TB8 into the 9th bit position of the transmit shift register and flags the TX Control unit that a transmission is requested. Transmission commences at S1P1 of the machine cycle following the next rollover in the divide-by-16 counter. (Thus, the bit times are synchronized to the divide-by-16 counter, not to the "write to SBUF" signal.)

The transmission begins with activation of SEND, which puts the start bit at TxD. One bit time later, DATA is activated, which enables the output bit of the transmit shift register to TxD. The first shift pulse occurs one bit time after that. The first shift clocks a 1 (the stop bit) into the 9th bit position of the shift register. Thereafter, only zeros are clocked in. Thus, as data bits shift out to the right, zeros are clocked in from the left. When TB8 is at the output position of the shift register, then the stop bit is just to the left of TB8, and all positions to the left of that contain zeros. This condition flags the TX Control unit to do one last shift and then deactivate SEND and set TI. This occurs at the 11th divide-by-16 rollover after "write to SUBF."

Reception is initiated by a detected 1-to-0 transition at RxD. For this purpose RxD is sampled at a rate of 16 times whatever baud rate has been established. When a transition is detected, the divide-by-16 counter is immediately reset, and 1FFH is written to the input shift register.

At the 7th, 8th, and 9th counter states of each bit time, the bit detector samples the value of R-D. The value accepted is the value that was seen in at least 2 of the 3 samples. If the value accepted during the first bit time is not 0, the receive circuits are reset and the unit goes back to looking for another 1-to-0 transition. If the start bit proves valid, it is shifted into the input shift register, and reception of the rest of the frame will proceed.

As data bits come in from the right, 1s shift out to the left. When the start bit arrives at the leftmost position in the shift register (which in Modes 2 and 3 is a 9-bit register), it flags the RX Control block to do one last shift, load SBUF and RB8, and set RI.

The signal to load SBUF and RB8, and to set RI, will be generated if, and only if, the following conditions are met at the time the final shift pulse is generated.

1. RI = 0, and
2. Either SM2 = 0, or the received 9th data bit = 1.

If either of these conditions is not met, the received frame is irretrievably lost, and RI is not set. If both conditions are met, the received 9th data bit goes into RB8, and the first 8 data bits go into SBUF. One bit time later, whether the above conditions were met or not, the unit goes back to looking for a 1-to-0 transition at the RxD input.

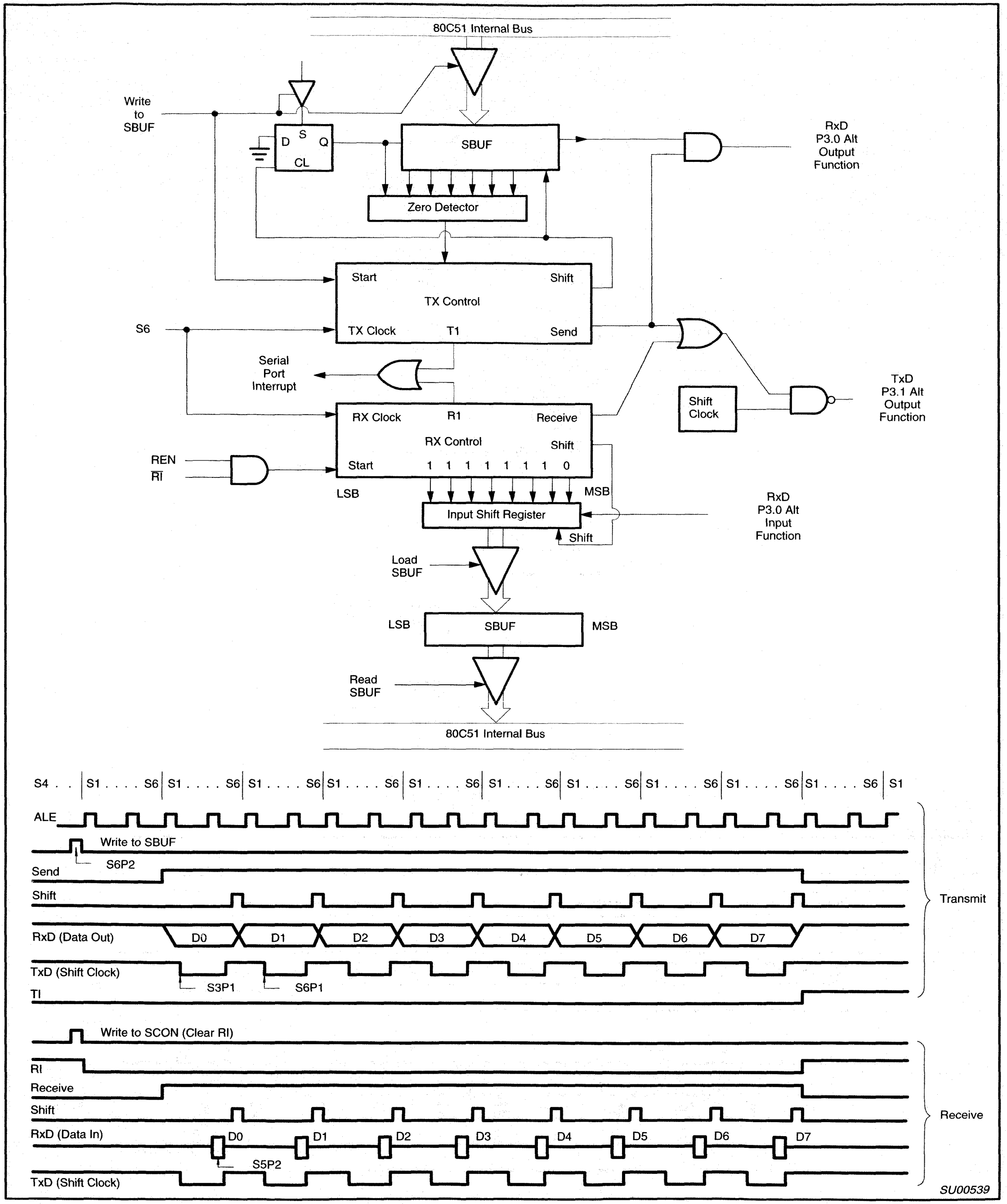


Figure 13. Serial Port Mode 0

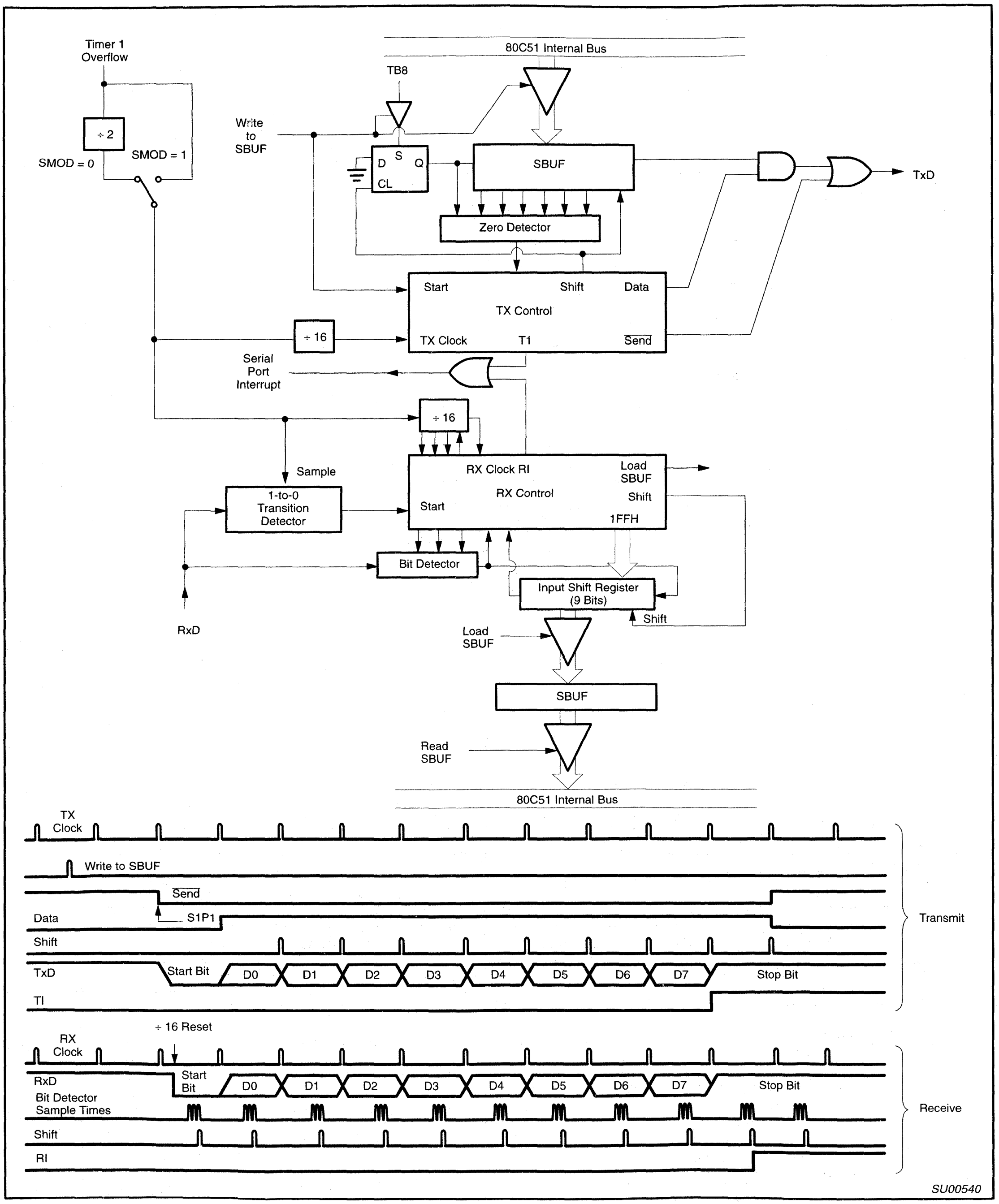


Figure 14. Serial Port Mode 1

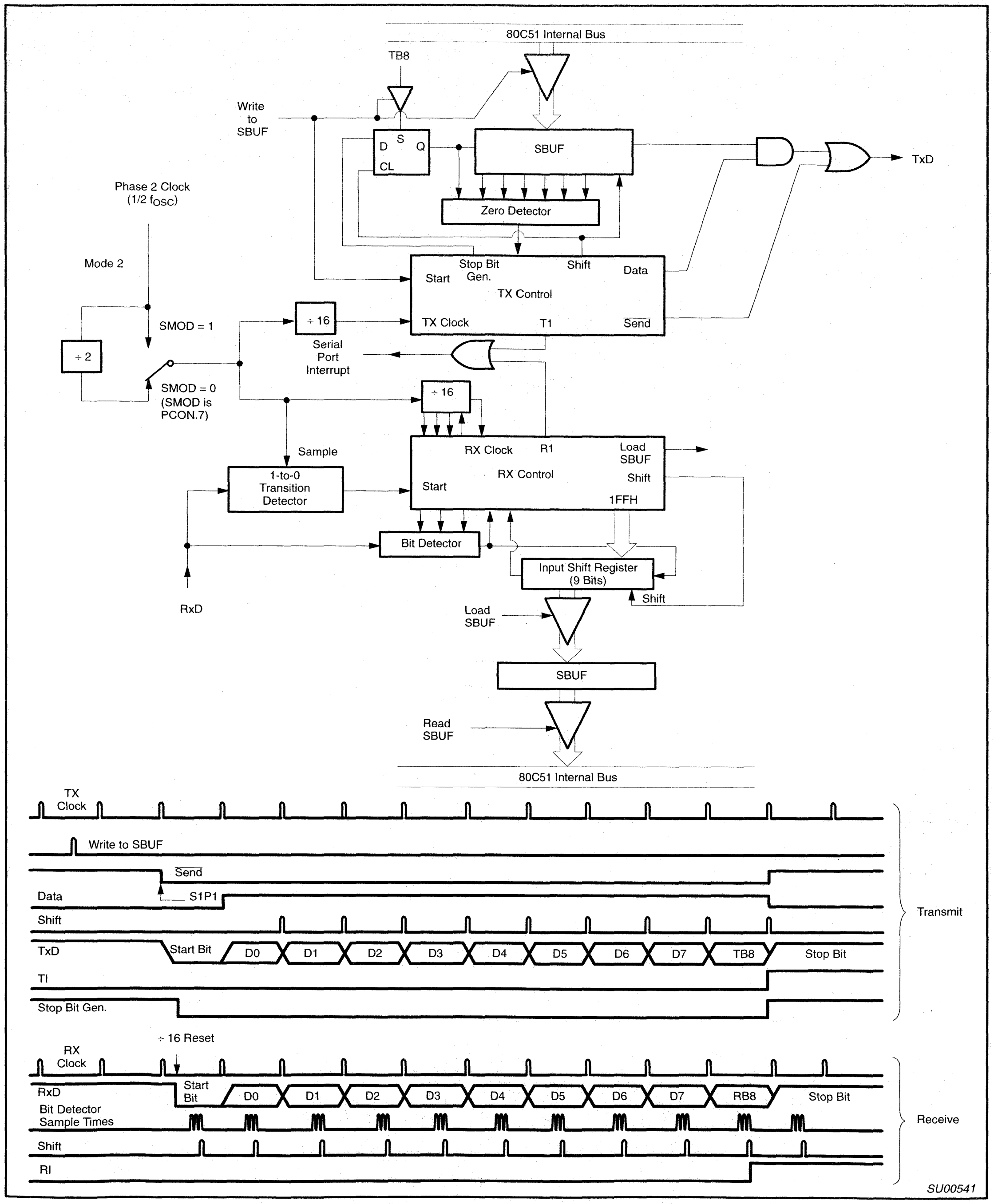


Figure 15. Serial Port Mode 2

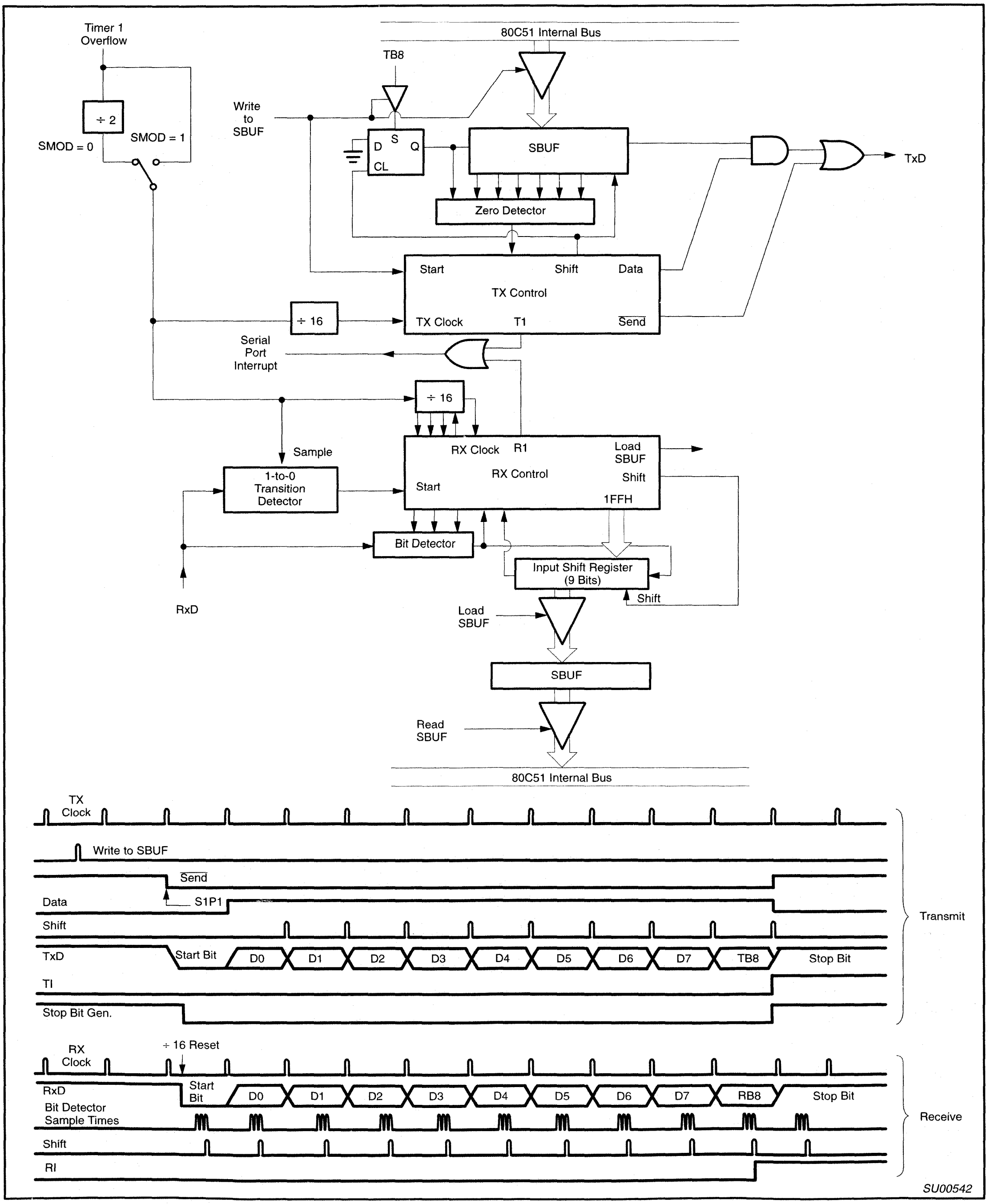


Figure 16. Serial Port Mode 3

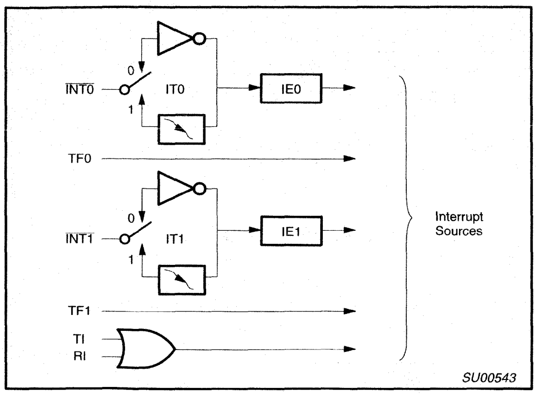


Figure 17. 80C51 Interrupt Sources

Interrupts

The 80C51 provides 5 interrupt sources. These are shown in Figure 17. The External Interrupts $\overline{\text{INT0}}$ and $\overline{\text{INT1}}$ can each be either level-activated or transition-activated, depending on bits IT0 and IT1 in Register TCON. The flags that actually generate these interrupts are bits IE0 and IE1 in TCON. When an external interrupt is generated, the flag that generated it is cleared by the hardware when the service routine is vectored to only if the interrupt was transition-activated. If the interrupt was level-activated, then the external requesting source is what controls the request flag, rather than the on-chip hardware.

The Timer 0 and Timer 1 Interrupts are generated by TF0 and TF1, which are set by a rollover in their respective Timer/Counter registers (except see Timer 0 in Mode 3). When a timer interrupt is generated, the flag that generated it is cleared by the on-chip hardware when the service routine is vectored to.

The Serial Port Interrupt is generated by the logical OR of RI and TI. Neither of these flags is cleared by hardware when the service routine is vectored to. In fact, the service routine will normally have to determine whether it was RI or TI that generated the interrupt, and the bit will have to be cleared in software.

All of the bits that generate interrupts can be set or cleared by software, with the same result as though it had been set or cleared by hardware. That is, interrupts can be generated or pending interrupts can be canceled in software.

Each of these interrupt sources can be individually enabled or disabled by setting or clearing a bit in Special Function Register IE (Figure 18). IE also contains a global disable bit, EA, which disables all interrupts at once.

Priority Level Structure

Each interrupt source can also be individually programmed to one of two priority levels by setting or clearing a bit in Special Function Register IP (Figure 19). A low-priority interrupt can itself be interrupted by a high-priority interrupt, but not by another low-priority interrupt. A high-priority interrupt can't be interrupted by any other interrupt source.

If two request of different priority levels are received simultaneously, the request of higher priority level is serviced. If requests of the same priority level are received simultaneously, an internal polling sequence determines which request is serviced. Thus within each priority level there is a second priority structure determined by the polling sequence as follows:

Source	**Priority Within Level**
1. IE0	(highest)
2. TF0	
3. IE1	
4. TF1	
5. RI+TI	(lowest)

Note that the "priority within level" structure is only used to resolve simultaneous requests of the same priority level.

The IP register contains a number of unimplemented bits. IP.7, IP.6, and IP.5 are reserved in the 80C51. User software should not write 1s to these positions, since they may be used in other 8051 Family products.

How Interrupts Are Handled

The interrupt flags are sampled at S5P2 of every machine cycle. The samples are polled during the following machine cycle. If one of the flags was in a set condition at S5P2 of the preceding cycle, the polling cycle will find it and the interrupt system will generate an LCALL to the appropriate service routine, provided this hardware-generated LCALL is not blocked by any of the following conditions:

1. An interrupt of equal or higher priority level is already in progress.
2. The current (polling) cycle is not the final cycle in the execution of the instruction in progress.
3. The instruction in progress is RETI or any write to the IE or IP registers.

Any of these three conditions will block the generation of the LCALL to the interrupt service routine. Condition 2 ensures that the instruction in progress will be completed before vectoring to any service routine. Condition 3 ensures that if the instruction in progress is RETI or any access to IE or IP, then at least one more instruction will be executed before any interrupt is vectored to.

The polling cycle is repeated with each machine cycle, and the values polled are the values that were present at S5P2 of the previous machine cycle. Note that if an interrupt flag is active but not being responded to for one of the above conditions, if the flag is not still active when the blocking condition is removed, the denied interrupt will not be serviced. In other words, the fact that the interrupt flag was once active but not serviced is not remembered. Every polling cycle is new.

MSB							LSB
EA	X	X	ES	ET1	EX1	ET0	EX0

BIT	SYMBOL	FUNCTION
IE.7	EA	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.
IE.5	—	Reserved.
IE.4	ES	Enables or disables the Serial Port interrupt. If ES=0, the Serial Port interrupt is disabled.
IE.3	ET1	Enables or disables the Timer 1 Overflow interrupt. If ET1=0, the Timer 1 interrupt is disabled.
IE.2	EX1	Enables or disables External Interrupt 1. If EX1=0, External interrupt 1 is disabled.
IE.1	ET0	Enables or disables the Timer 0 Overflow interrupt. If ET0=0, the Timer 0 interrupt is disabled.
IE.0	EX0	Enables or disables External interrupt 0. If EX0=0, External interrupt 0 is disabled.

SU00544

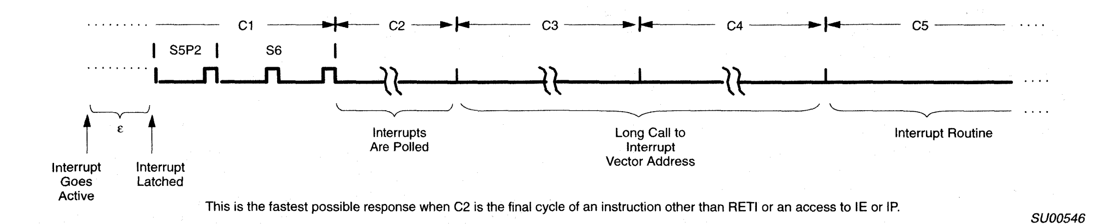
Figure 18. Interrupt Enable Register (IE)

MSB							LSB
X	X	X	PS	PT1	PX1	PT0	PX0

BIT	SYMBOL	FUNCTION
IP.7	—	Reserved.
IP.6	—	Reserved.
IP.5	—	Reserved.
IP.4	PS	Defines the Serial Port interrupt priority level. PS=1 programs it to the higher priority level.
IP.3	PT1	Defines the Timer 1 interrupt priority level. PT1=1 programs it to the higher priority level.
IP.2	PX1	Defines the External Interrupt 1 priority level. PX1=1 programs it to the higher priority level.
IP.1	PT0	Enables or disables the Timer 0 interrupt priority level. PT0=1 programs it to the higher priority level.
IP.0	PX0	Defines the External Interrupt 0 priority level. PX0=1 programs it to the higher priority level.

SU00545

Figure 19. Interrupt Priority Register (IP)



This is the fastest possible response when C2 is the final cycle of an instruction other than RETI or an access to IE or IP.

Figure 20. Interrupt Response Timing Diagram

The polling cycle/LCALL sequence is illustrated in Figure 20.

Note that if an interrupt of higher priority level goes active prior to S5P2 of the machine cycle labeled C3 in Figure 20, then in accordance with the above rules it will be vectored to during C5 and C6, without any instruction of the lower priority routine having been executed.

Thus the processor acknowledges an interrupt request by executing a hardware-generated LCALL to the appropriate servicing routine. In some cases it also clears the flag that generated the interrupt, and in other cases it doesn't. It never clears the Serial Port flag. This has to be done in the user's software. It clears an external interrupt flag (IE0 or IE1) only if it was transition-activated. The hardware-generated LCALL pushes the contents of the Program Counter on to the stack (but it does not save the PSW) and reloads the PC with an address that depends on the source of the interrupt being vectored to, as shown below:

Source	Vector Address
IE0	0003H
TF0	000BH
IE1	0013H
TF1	001BH
RI+TI	0023H

Execution proceeds from that location until the RETI instruction is encountered. The RETI instruction informs the processor that this interrupt routine is no longer in progress, then pops the top two bytes from the stack and reloads the Program Counter. Execution of the interrupted program continues from where it left off.

Note that a simple RET instruction would also have returned execution to the interrupted program, but it would have left the interrupt control system thinking an interrupt was still in progress, making future interrupts impossible.

External Interrupts

The external sources can be programmed to be level-activated or transition-activated by setting or clearing bit IT1 or IT0 in Register TCON. If ITx = 0, external interrupt x is triggered by a detected low at the INTx pin. If ITx = 1, external interrupt x is edge triggered. In this mode if successive samples of the INTx pin show a high in one cycle and a low in the next cycle, interrupt request flag IEx in TCON is set. Flag bit IEx then requests the interrupt.

Since the external interrupt pins are sampled once each machine cycle, an input high or low should hold for at least 12 oscillator periods to ensure sampling. If the external interrupt is transition-activated, the external source has to hold the request pin high for at least one cycle, and then hold it low for at least one cycle. This is done to ensure that the transition is seen so that interrupt request flag IEx will be set. IEx will be automatically cleared by the CPU when the service routine is called.

If the external interrupt is level-activated, the external source has to hold the request active until the requested interrupt is actually generated. Then it has to deactivate the request before the interrupt service routine is completed, or else another interrupt will be generated.

Response Time

The $\overline{INT0}$ and $\overline{INT1}$ levels are inverted and latched into IE0 and IE1 at S5P2 of every machine cycle. The values are not actually polled by the circuitry until the next machine cycle. If a request is active and conditions are right for it to be acknowledged, a hardware subroutine call to the requested service routine will be the next instruction to be executed. The call itself takes two cycles. Thus, a minimum of three complete machine cycles elapse between activation of an external interrupt request and the beginning of execution of the first instruction of the service routine. Figure 20 shows interrupt response timings.

A longer response time would result if the request is blocked by one of the 3 previously listed conditions. If an interrupt of equal or higher priority level is already in progress, the additional wait time obviously depends on the nature of the other interrupt's service routine. If the instruction in progress is not in its final cycle, the additional wait time cannot be more the 3 cycles, since the longest instructions (MUL and DIV) are only 4 cycles long, and if the instruction in progress is RETI or an access to IE or IP, the additional wait time cannot be more than 5 cycles (a maximum of one more cycle to complete the instruction in progress, plus 4 cycles to complete the next instruction if the instruction is MUL or DIV).

Thus, in a single-interrupt system, the response time is always more than 3 cycles and less than 9 cycles.

Single-Step Operation

The 80C51 interrupt structure allows single-step execution with very little software overhead. As previously noted, an interrupt request will not be responded to while an interrupt of equal priority level is still in progress, nor will it be responded to after RETI until at least one other instruction has been executed. Thus, once an interrupt routine has been entered, it cannot be re-entered until at least one instruction of the interrupted program is executed. One way to use this feature for single-step operation is to program one of the external interrupts (e.g., $\overline{INT0}$) to be level-activated. The service routine for the interrupt will terminate with the following code:

```
JNB   P3.2,$  ;Wait Till INT0 Goes High
JB    P3.2,$  ;Wait Till INT0 Goes Low
RETI          ;Go Back and Execute One Instruction
```

Now if the $\overline{INT0}$ pin, which is also the P3.2 pin, is held normally low, the CPU will go right into the External Interrupt 0 routine and stay there until $\overline{INT0}$ is pulsed (from low to high to low). Then it will execute RETI, go back to the task program, execute one instruction, and immediately re-enter the External Interrupt 0 routine to await the next pulsing of P3.2. One step of the task program is executed each time P3.2 is pulsed.

Reset

The reset input is the RST pin, which is the input to a Schmitt Trigger. 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 generating an internal reset, with the timing shown in Figure 21.

The external reset signal is asynchronous to the internal clock. The RST pin is sampled during State 5 Phase 2 of every machine cycle. The port pins will maintain their current activities for 19 oscillator periods after a logic 1 has been sampled at the RST pin; that is, for 19 to 31 oscillator periods after the external reset signal has been applied to the RST pin.

The internal reset algorithm writes 0s to all the SFRs except the port latches, the Stack Pointer, and SBUF. The port latches are initialized to FFH, the Stack Pointer to 07H, and SBUF is indeterminate. Table 1 lists the SFR reset values. The internal RAM is not affected by reset. On power up the RAM content is indeterminate.

Table 1. 80C51 SFR Reset Values

REGISTER	RESET VALUE
PC	000H
ACC	00H
B	00H
PSW	00H
SP	07H
DPTR	0000H
P0–P3	FFH
IP	XXX00000B
IE	0XX00000B
TMOD	00H
TCON	00H
TH0	00H
TL0	00H
TH1	00H
TL1	00H
SCON	00H
SBUF	Indeterminate
PCON (NMOS)	0XXXXXXXB
PCON (CMOS)	0XXX0000B

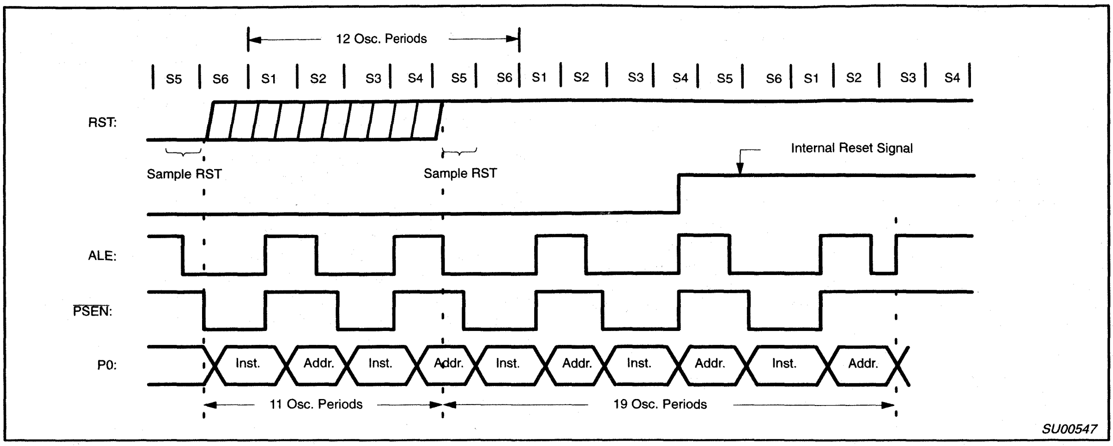


Figure 21. Reset Timing

Power-on Reset

An automatic reset can be obtained when V_{CC} is turned on by connecting the RST pin to V_{CC} through a 10µf capacitor and to V_{SS} through an 8.2k resistor, providing the V_{CC} rise time does not exceed 1 millisecond and the oscillator start-up time does not exceed 10 milliseconds. This power-on reset circuit is shown in Figure 22. The CMOS devices do not require the 8.2k pulldown resistor, although its presence does no harm.

When power is turned on, the circuit holds the RST pin high for an amount of time that depends on the value of the capacitor and the rate at which it charges. To ensure a good reset, the RST pin must be high long enough to allow the oscillator time to start-up (normally a few ms) plus two machine cycles.

Note that the port pins will be in a random state until the oscillator has started and the internal reset algorithm has written 1s to them.

With this circuit, reducing V_{CC} quickly to 0 causes the RST pin voltage to momentarily fall below 0V. However, this voltage is internally limited, and will not harm the device.

Power-Saving Modes of Operation

For applications where power consumption is critical the CMOS version provides power reduced modes of operation as a standard feature. The power down mode in NMOS devices is no longer a standard feature.

CMOS Power Reduction Mode

CMOS versions have two power reducing modes, Idle and Power Down. The input through which backup power is supplied during these operations is V_{CC}. Figure 23 shows the internal circuitry which implements these features. In the Idle modes (IDL = 1), the oscillator continues to run and the Interrupt, Serial Port, and Timer blocks continue to be clocked, but the clock signal is gated off to the CPU. In Power Down (PD = 1), the oscillator is frozen. The Idle and Power Down Modes are activated by setting bits in Special Function Register PCON. The address of this register is 87H. Figure 24 details its contents.

In the NMOS devices the PCON register only contains SMOD. The other four bits are implemented only in the CMOS devices. User software should never write 1s to unimplemented bits, since they may be used in other 80C51 Family products.

Idle Mode

An instruction that sets PCON.0 causes that to be the last instruction executed before going into the Idle mode, the internal clock signal is gated off to the CPU but not to the Interrupt, Timer, and Serial Port functions. The CPU status is preserved in its entirety; the Stack Pointer, Program Counter, Program Status Word, Accumulator, and all other registers maintain their data during Idle. The port pins hold the logical states they had at the time Idle was activated. ALE and $\overline{\text{PSEN}}$ hold at logic high levels.

There are two ways to terminate the Idle. Activation of any enabled interrupt will cause PCON.0 to be cleared by hardware, terminating the Idle mode. The interrupt will be serviced, and following RETI, the next instruction to be executed will be the one following the instruction that put the device into Idle.

The flag bits GF0 and GF1 can be used to give an indication if an interrupt occurred during normal operation or during an Idle. For example, an instruction that activates Idle can also set one or both flag bits. When Idle is terminated by an interrupt, the interrupt service routine can examine the flag bits. The other way of terminating the Idle mode is with a hardware reset. Since the clock oscillator is still running, the hardware reset needs to be held active for only two machine cycles (24 oscillator periods) to complete the reset.

The signal at the RST pin clears the IDL bit directly and asynchronously. At this time the CPU resumes program execution from where it left off; that is, at the instruction following the one that invoked the Idle Mode. As shown in Figure 21, two or three machine cycles of program execution may take place before the internal reset algorithm takes control. On-chip hardware inhibits access to the internal RAM during this time, but access to the port pins is not inhibited, so, the insertion of 3 NOP instructions is recommended following the instruction that invokes idle mode. To eliminate the possibility of unexpected outputs at the port pins, the instruction following the one that invokes Idle should not be one that writes to a port pin or to external Data RAM.

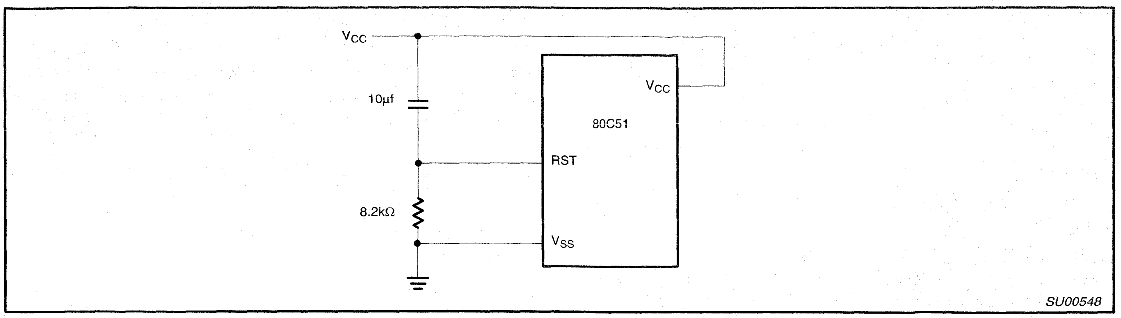


Figure 22. Power-On Reset Circuit

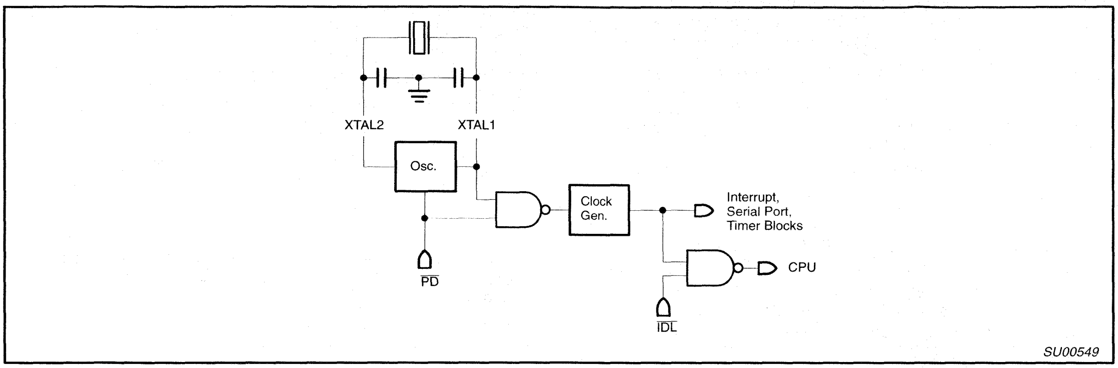


Figure 23. Idle and Power Down Hardware

MSB LSB

SMOD	—	—	—	GF1	GF0	PD	IDL

BIT	SYMBOL	FUNCTION
PCON.7	SMOD	Double Baud rate bit. When set to a 1 and Timer 1 is used to generate baud rate, and the Serial Port is used in modes 1, 2, or 3.
PCON.6	—	Reserved.
PCON.5	—	Reserved.
PCON.4	—	Reserved.
PCON.3	GF1	General-purpose flag bit.
PCON.2	GF0	General-purpose flag bit.
PCON.1	PD	Power-Down bit. Setting this bit activates power-down operation.
PCON.0	IDL	Idle mode bit. Setting this bit activate idle mode operation.

If 1s are written to PD and IDL at the same time, PD takes precedence. The reset value of PCON is (0XXX0000). In the NMOS devices, the PCON register only contains SMOD. The other four bits are implemented only in the CMOS devices. User software should never write 1s to unimplemented bits, since they may be used in future products.

SU00550

Figure 24. Power Control (PCON) Register

Power-Down Mode

An instruction that sets PCON.1 causes that to be the last instruction executed before going into the Power Down mode. In the Power Down mode, the on-chip oscillator is stopped. With the clock frozen, all functions are stopped, the contents of the on-chip RAM and Special Function Registers are maintained. The port pins output the values held by their respective SFRs. The ALE and $\overline{\text{PSEN}}$ output are held low.

The only exit from Power Down is a hardware reset. Reset redefines all the SFRs, but does not change the on-chip RAM.

In the Power Down mode of operation, V_{CC} can be reduced to as low as 2V. Care must be taken, however, to ensure that V_{CC} is not reduced before the Power Down mode is invoked, and that V_{CC} is restored to its normal operating level, before the Power Down mode is terminated. The reset that terminates Power Down also frees the oscillator. The reset should not be activated before V_{CC} is restored to its normal operating level, and must be held active long enough to allow the oscillator to restart and stabilize (normally less than 10ms).

ONCE Mode

The ONCE ("on-circuit emulation") mode facilitates testing and debugging of systems using the device without the device having to be removed from the circuit. The ONCE mode is invoked by:

1. Pull ALE low while the device in in reset and $\overline{\text{PSEN}}$ is high;
2. Hold ALE low as RST is deactivated.

While the device is in the ONCE mode, the Port 0 pins go into a float state, and the other port pins and ALE and $\overline{\text{PSEN}}$ are weakly pulled high. The oscillator circuit remains active. While the device is in this mode, an emulator or test CPU can be used to drive the circuit. Normal operation is restored after a normal reset is applied.

The On-Chip Oscillators

CMOS Versions

The on-chip oscillator circuitry for the 80C51, shown in Figure 25, consists of a single stage linear inverter intended for use as a crystal-controlled, positive reactance oscillator in the same manner as the NMOS parts. However, there are some important differences.

One difference is that the 80C51 is able to turn off its oscillator under software control (by writing a 1 to the PD bit in PCON). Another difference is that, in the 80C51, the internal clocking circuitry is driven by the signal at XTAL1, whereas in the NMOS versions it is by the signal at XTAL2.

The feedback resistor R_f in Figure 25 consists of paralleled n- and p-channel FETs controlled by the PD bit, such that R_f is opened when PD = 1. The diodes D1 and D2, which act as clamps to V_{CC} and V_{SS}, are parasitic to the R_f FETs. The oscillator can be used with the same external components as the NMOS versions, as shown in Figure 26. Typically, C1 = C2 = 30pF when the feedback element is a quartz crystal, and C1 = C2 = 47pF when a ceramic resonator is used.

When a crystal is used at frequencies above 25MHz, C1 and C2 should be in the range of 20pF to 25pF.

To drive the CMOS parts with an external clock source, apply the external clock signal to XTAL1, and leave XTAL2 float, as shown in Figure 27.

The reason for this change from the way the NMOS part is driven can be seen by comparing Figure 25. In the NMOS devices the internal timing circuits are driven by the signal at XTAL2. In the CMOS devices the internal timing circuits are driven by the signal at XTAL1.

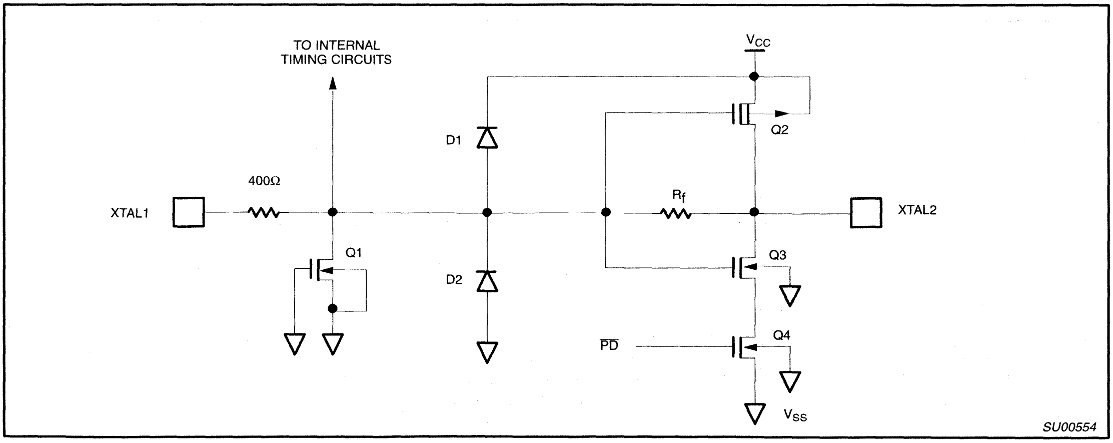


Figure 25. On-Chip Oscillator Circuitry in the CMOS Version of the 80C51 Family

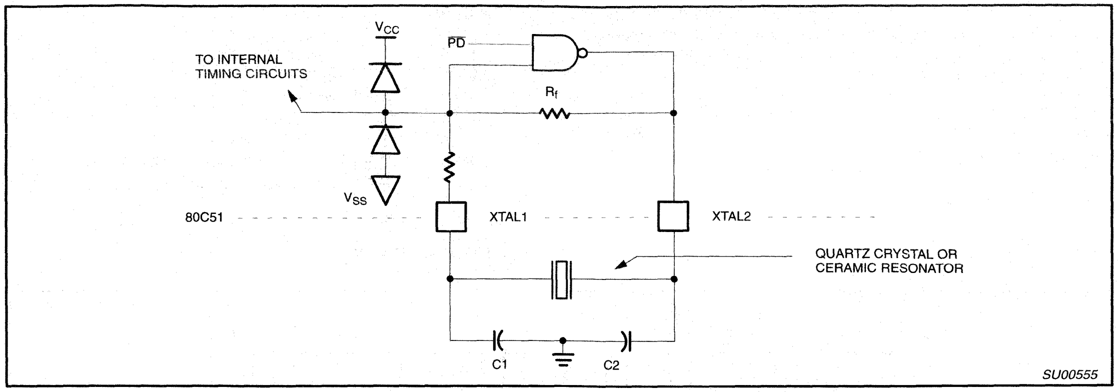


Figure 26. Using the CMOS On-Chip Oscillator

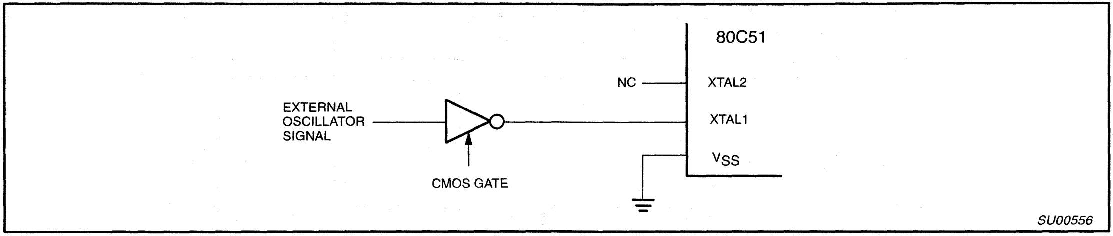


Figure 27. Driving the CMOS Family Parts with an External Clock Source

Internal Timing

Figures 28 through 31 show when the various strobe and port signals are clocked internally. The figures do not show rise and fall times of the signals, nor do they show propagation delays between the XTAL2 signal and events at other pins.

Rise and fall times are dependent on the external loading that each pin must drive. They are often taken to be something in the neighborhood of 10ns, measured between 0.8V and 2.0V.

Propagation delays are different for different pins. For a given pin they vary with pin loading, temperature, V_{CC}, and manufacturing lot. If the XTAL2 waveform is taken as the timing reference, prop delays may vary up to ±200%.

The AC Timings section of the data sheets do not reference any timing to the XTAL2 waveform. Rather, they relate the critical edges of control and input signals to each other. The timings published in the data sheets include the effects of propagation delays under the specified test conditions.

80C51 Pin Descriptions

ALE/PROG: Address Latch Enable output pulse for latching the low byte of the address during accesses to external memory. ALE is emitted at a constant rate of 1/6 of the oscillator frequency, for external timing or clocking purposes, even when there are no accesses to external memory. (However, one ALE pulse is skipped during each access to external Data Memory.) This pin is also the program pulse input (PROG) during EPROM programming.

$\overline{\text{PSEN}}$: Program Store Enable is the read strobe to external Program Memory. When the device is executing out of external Program Memory, $\overline{\text{PSEN}}$ is activated twice each machine cycle (except that two $\overline{\text{PSEN}}$ activations are skipped during accesses to external Data Memory). $\overline{\text{PSEN}}$ is not activated when the device is executing out of internal Program Memory.

$\overline{\text{EA}}$/V_{PP}: When $\overline{\text{EA}}$ is held high the CPU executes out of internal Program Memory (unless the Program Counter exceeds 0FFFH in the 80C51). Holding $\overline{\text{EA}}$ low forces the CPU to execute out of external memory regardless of the Program Counter value. In the 80C31, $\overline{\text{EA}}$ must be externally wired low. In the EPROM devices, this pin also receives the programming supply voltage (V_{PP}) during EPROM programming.

XTAL1: Input to the inverting oscillator amplifier.

XTAL2: Output from the inverting oscillator amplifier.

Port 0: Port 0 is an 8-bit open drain bidirectional port. As an open drain output port, it can sink eight LS TTL loads. Port 0 pins that have 1s written to them float, and in that state will function as high impedance inputs. Port 0 is also the multiplexed low-order address and data bus during accesses to external memory. In this application it uses strong internal pullups when emitting 1s. Port 0 emits code bytes during program verification. In this application, external pullups are required.

Port 1: Port 1 is an 8-bit bidirectional I/O port with internal pullups. Port 1 pins that have 1s written to them are pulled high by the internal pullups, and in that state can be used as inputs. As inputs, port 1 pins that are externally being pulled low will source current because of the internal pullups.

Port 2: Port 2 is an 8-bit bidirectional I/O port with internal pullups. Port 2 emits the high-order address byte during accesses to external memory that use 16-bit addresses. In this application, it uses the strong internal pullups when emitting 1s.

Port 3: Port 3 is an 8-bit bidirectional I/O port with internal pullups. It also serves the functions of various special features of the 80C51 Family as follows:

Port Pin	Alternate Function
P3.0	RxD (serial input port)
P3.1	TxD (serial output port)
P3.2	$\overline{\text{INT0}}$ (external interrupt 0)
P3.3	$\overline{\text{INT1}}$ (external interrupt 1)
P3.4	T0 (timer 0 external input)
P3.5	T1 (timer 1 external input)
P3.6	$\overline{\text{WR}}$ (external data memory write strobe)
P3.7	$\overline{\text{RD}}$ (external data memory read strobe)

V_{CC}: Supply voltage

V_{SS}: Circuit ground potential

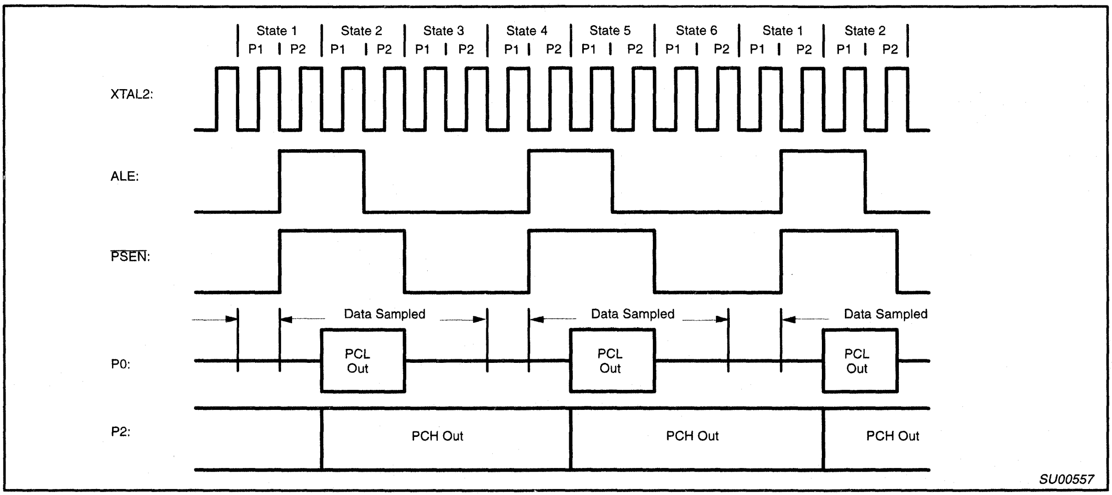


Figure 28. External Program Memory Fetches

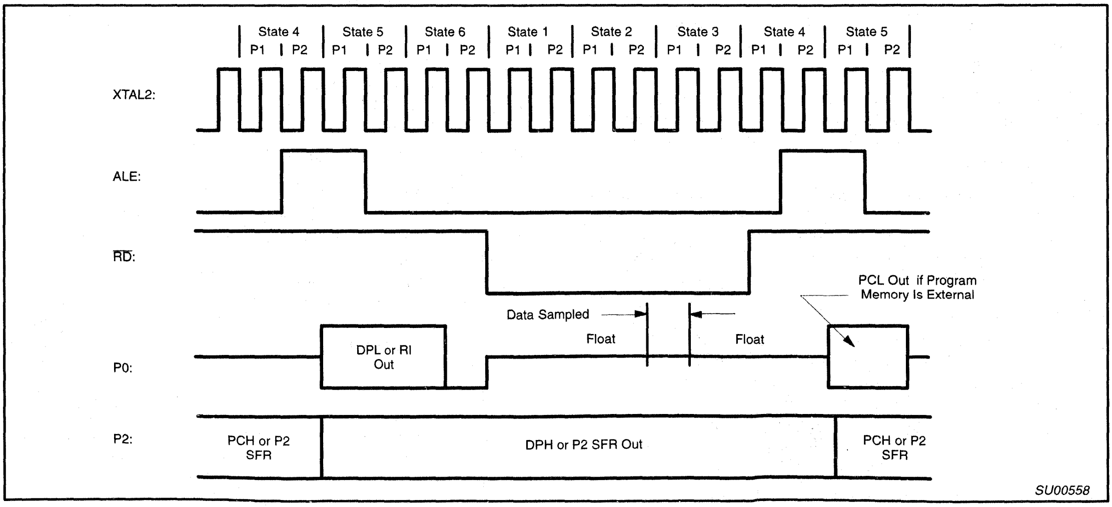


Figure 29. External Data Memory Read Cycle

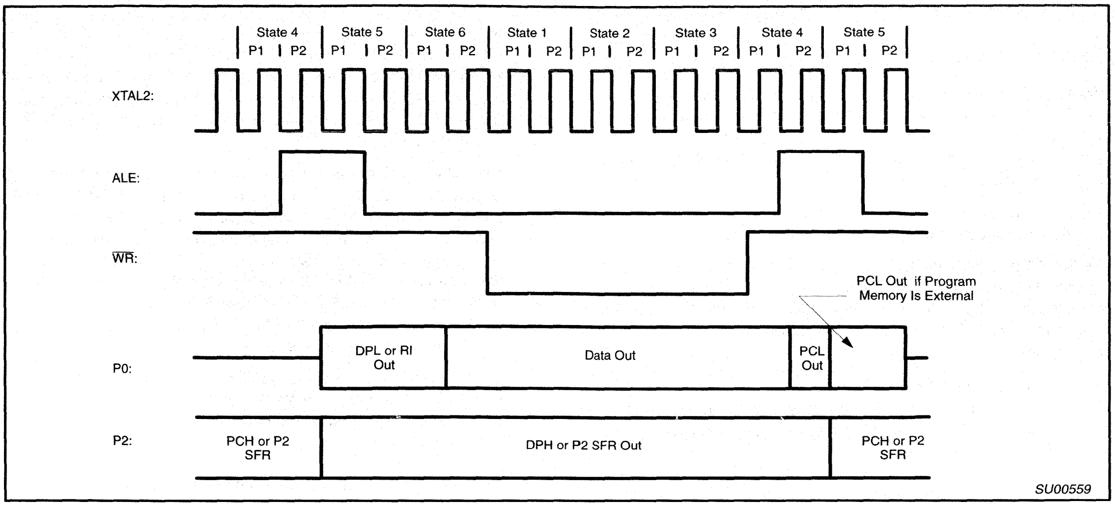


Figure 30. External Data Memory Write Cycle

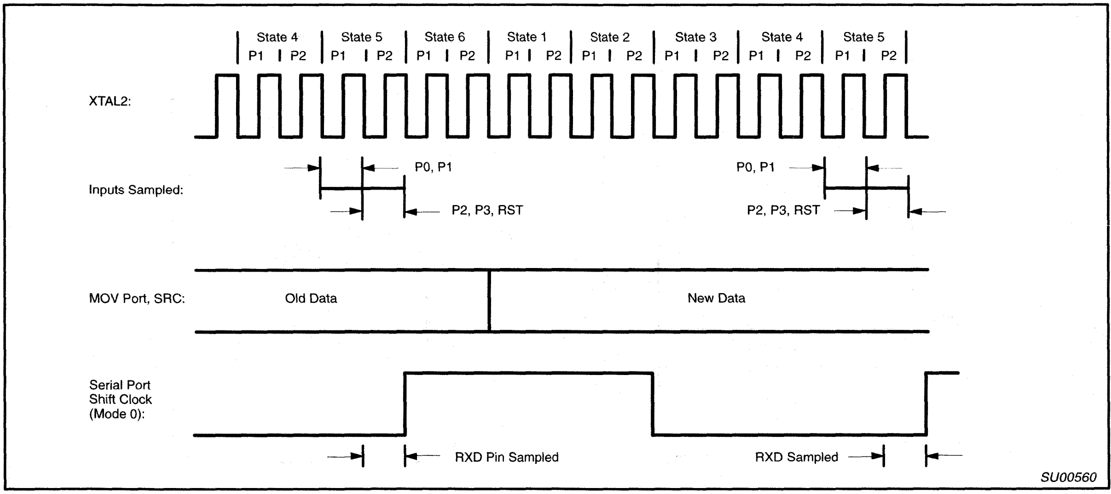


Figure 31. Port Operation

PROGRAMMER'S GUIDE AND INSTRUCTION SET

Memory Organization

Program Memory

The 80C51 has separate address spaces for program and data memory. The Program memory can be up to 64k bytes long. The lower 4k can reside on-chip. Figure 1 shows a map of the 80C51 program memory.

The 80C51 can address up to 64k bytes of data memory to the chip. The MOVX instruction is used to access the external data memory.

The 80C51 has 128 bytes of on-chip RAM, plus a number of Special Function Registers (SFRs). The lower 128 bytes of RAM can be accessed either by direct addressing (MOV data addr) or by indirect addressing (MOV @Ri). Figure 2 shows the Data Memory organization.

Direct and Indirect Address Area

The 128 bytes of RAM which can be accessed by both direct and indirect addressing can be divided into three segments as listed below and shown in Figure 3.

Register Banks 0-3: Locations 0 through 1FH (32 bytes). The device after reset defaults to register bank 0. To use the other register banks, the user must select them in software. Each register bank contains eight 1-byte registers 0 through 7. Reset initializes the stack pointer to location 07H, and it is incremented once to start from location 08H, which is the first register (R0) of the second register bank. Thus, in order to use more than one register bank, the SP should be initialized to a different location of the RAM where it is not used for data storage (i.e., the higher part of the RAM).

Bit Addressable Area: 16 bytes have been assigned for this segment, 20H-2FH. Each one of the 128 bits of this segment can be directly addressed (0-7FH). The bits can be referred to in two ways, both of which are acceptable by most assemblers. One way is to refer to their address (i.e., 0-7FH). The other way is with reference to bytes 20H to 2FH. Thus, bits 0-7 can also be referred to as bits 20.0-20.7, and bits 8-FH are the same as 21.0-21.7, and so on. Each of the 16 bytes in this segment can also be addressed as a byte.

Scratch Pad Area: 30H through 7FH are available to the user as data RAM. However, if the stack pointer has been initialized to this area, enough bytes should be left aside to prevent SP data destruction.

Figure 2 shows the different segments of the on-chip RAM.

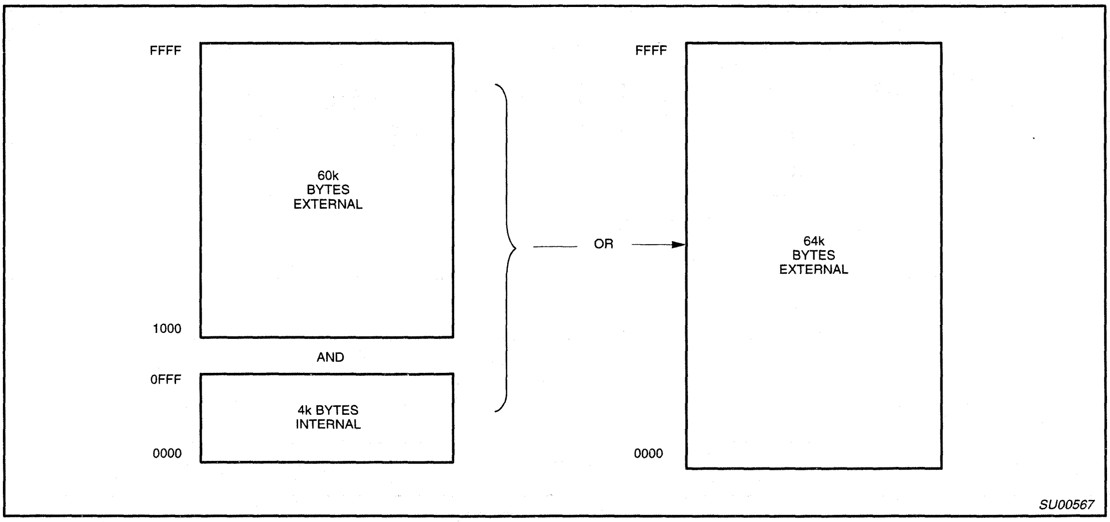


Figure 1. 80C51 Program Memory

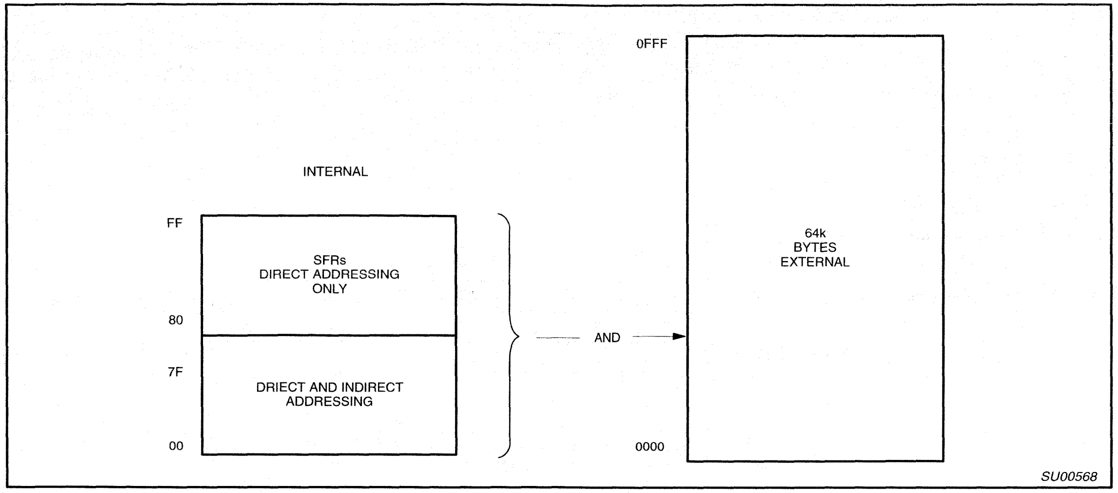


Figure 2. 80C51 Data Memory

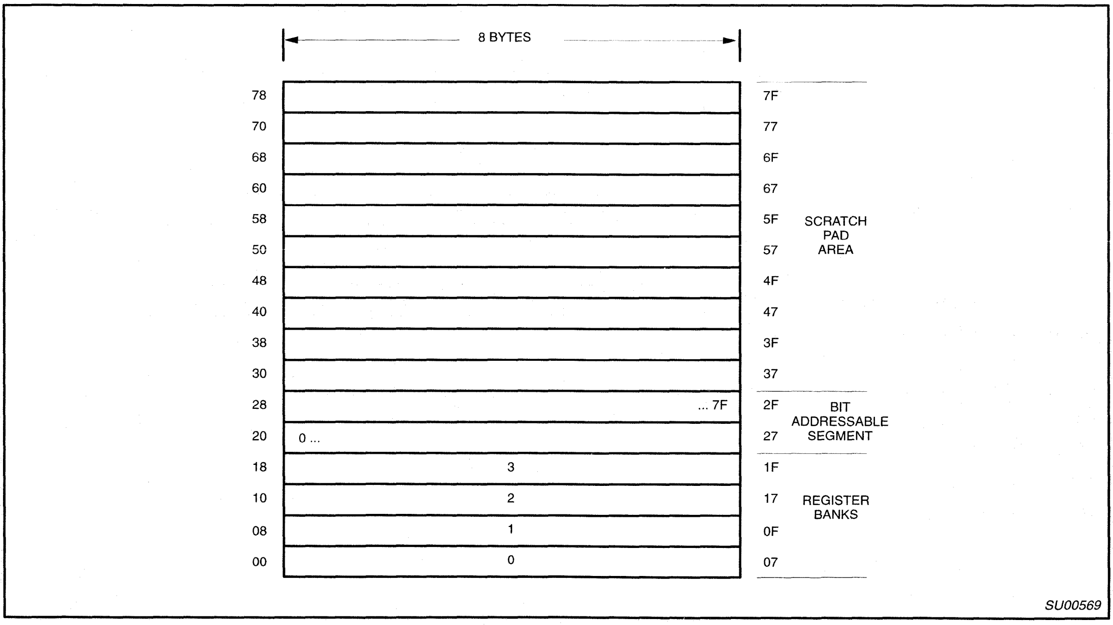


Figure 3. 128 Bytes of RAM Direct and Indirect Addressable

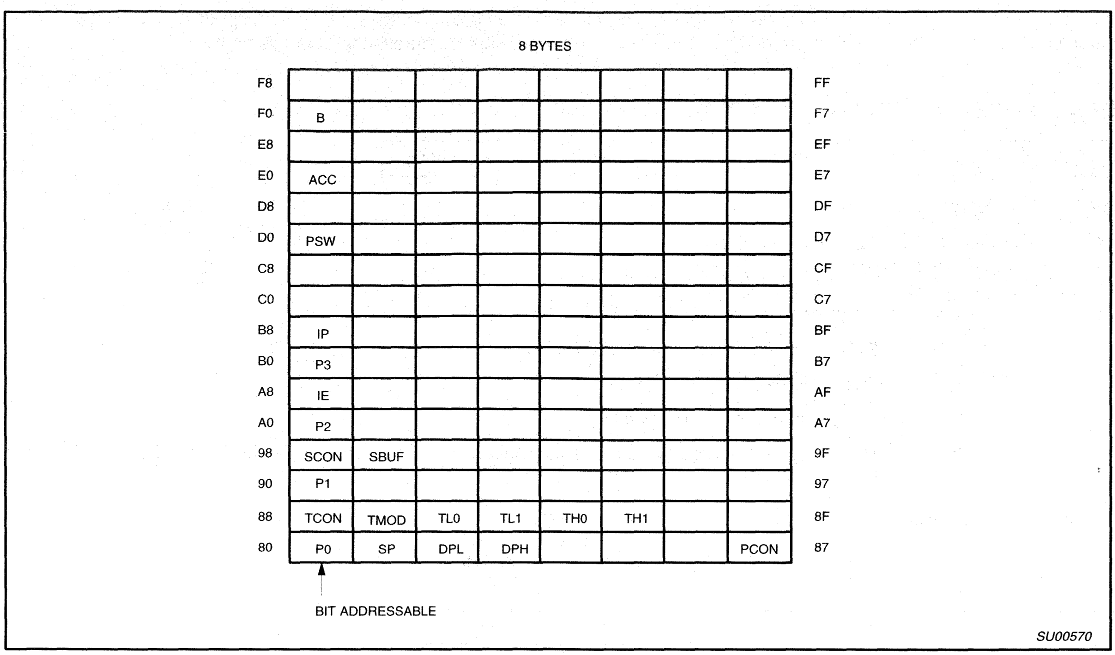
Table 1. 80C51 Special Function Registers

SYMBOL	DESCRIPTION	DIRECT ADDRESS	BIT ADDRESS, SYMBOL, OR ALTERNATIVE PORT FUNCTION								RESET VALUE
			MSB							LSB	
ACC*	Accumulator	E0H	E7	E6	E5	E4	E3	E2	E1	E0	00H
B*	B register	F0H	F7	F6	F5	F4	F3	F2	F1	F0	00H
DPTR	Data pointer (2 bytes)										
DPH	Data pointer high	83H									00H
DPL	Data pointer low	82H									00H
			AF	AE	AD	AC	AB	AA	A9	A8	
IE*	Interrupt enable	A8H	EA	–	–	ES	ET1	EX1	ET0	EX0	0x000000B
			BF	BE	BD	BC	BB	BA	B9	B8	
IP*	Interrupt priority	B8H	–	–	–	PS	PT1	PX1	PT0	PX0	xx000000B
			87	86	85	84	83	82	81	80	
P0*	Port 0	80H	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0	FFH
			97	96	95	94	93	92	91	90	
P1*	Port 1	90H	–	–	–	–	–	–	T2EX	T2	FFH
			A7	A6	A5	A4	A3	A2	A1	A0	
P2*	Port 2	A0H	A15	A14	A13	A12	A11	A10	A9	A8	FFH
			B7	B6	B5	B4	B3	B2	B1	B0	
P3*	Port 3	B0H	$\overline{RD}$	$\overline{WR}$	T1	T0	$\overline{INT1}$	$\overline{INT0}$	TxD	Rxd	FFH
PCON[1]	Power control	87H	SMOD	–	–	–	GF1	GF0	PD	IDL	0xxxxxxxB
			D7	D6	D5	D4	D3	D2	D1	D0	
PSW*	Program status word	D0H	CY	AC	F0	RS1	RS0	OV	–	P	00H
SBUF	Serial data buffer	99H									xxxxxxxxB
			9F	9E	9D	9C	9B	9A	99	98	
SCON*	Serial controller	98H	SM0	SM1	SM2	REN	TB8	RB8	TI	RI	00H
SP	Stack pointer	81H									07H
			8F	8E	8D	8C	8B	8A	89	88	
TCON*	Timer control	88H	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	
TH0	Timer high 0	8CH									00H
TH1	Timer high 1	8DH									00H
TL0	Timer low 0	8AH									00H
TL1	Timer low 1	8BH									00H
TMOD	Timer mode	89H	GATE	$C/\overline{T}$	M1	M0	GATE	$C/\overline{T}$	M1	M0	00H

NOTES:

* Bit addressable

1. Bits GF1, GF0, PD, and IDL of the PCON register are not implemented on the NMOS 8051/8031.


8 BYTES
BIT ADDRESSABLE
SU00570

F8									FF
F0	B								F7
E8									EF
E0	ACC								E7
D8									DF
D0	PSW								D7
C8									CF
C0									C7
B8	IP								BF
B0	P3								B7
A8	IE								AF
A0	P2								A7
98	SCON	SBUF							9F
90	P1								97
88	TCON	TMOD	TL0	TL1	TH0	TH1			8F
80	P0	SP	DPL	DPH				PCON	87

Figure 4. SFR Memory Map

Those SFRs that have their bits assigned for various functions are listed in this section. A brief description of each bit is provided for quick reference. For more detailed information refer to the Architecture Chapter of this book.

PSW: PROGRAM STATUS WORD. BIT ADDRESSABLE.

CY	AC	F0	RS1	RS0	OV	–	P

CY PSW.7 Carry Flag.

AC PSW.6 Auxiliary Carry Flag.

F0 PSW.5 Flag 0 available to the user for general purpose.

RS1 PSW.4 Register Bank selector bit 1 (SEE NOTE 1).

RS0 PSW.3 Register Bank selector bit 0 (SEE NOTE 1).

OV PSW.2 Overflow Flag.

– PSW.1 Usable as a general purpose flag.

P PSW.0 Parity flag. Set/cleared by hardware each instruction cycle to indicate an odd/even number of '1' bus in the accumulator.

NOTE:

1. The value presented by RS0 and RS1 selects the corresponding register bank.

RS1	RS0	REGISTER BANK	ADDRESS
0	0	0	00H-07H
0	1	1	08H-0FH
1	0	2	10H-17H
1	1	3	18H-1FH

PCON: POWER CONTROL REGISTER. NOT BIT ADDRESSABLE.

SMOD	–	–	–	GF1	GF0	PD	IDL

SMOD Double baud rate bit. If Timer 1 is used to generate baud rate and SMOD = 1, the baud rate is doubled when the Serial Port is used in modes 1, 2, or 3.

– Not implemented, reserved for future use.*

– Not implemented reserved for future use.*

– Not implemented reserved for future use.*

GF1 General purpose flag bit.

GF0 General purpose flag bit.

PD Power Down Bit. Setting this bit activates Power Down operation in the 80C51. (Available only in CMOS.)

IDL Idle mode bit. Setting this bit activates Idle Mode operation in the 80C51. (Available only in CMOS.)

If 1s are written to PD and IDL at the same time, PD takes precedence.

* User software should not write 1s to reserved bits. These bits may be used in future 8051 products to invoke new features.

INTERRUPTS:

To use any of the interrupts in the 80C51 Family, the following three steps must be taken.

1. **Set the EA (enable all) bit in the IE register to 1.**
2. **Set the corresponding individual interrupt enable bit in the IE register to 1.**
3. **Begin the interrupt service routine at the corresponding Vector Address of that interrupt. See Table below.**

INTERRUPT SOURCE	VECTOR ADDRESS
IE0	0003H
TF0	000BH
IE1	0013H
TF1	001BH
RI & TI	0023H

In addition, for external interrupts, pins INT0 and INT1 (P3.2 and P3.3) must be set to 1, and depending on whether the interrupt is to be level or transition activated, bits IT0 or IT1 in the TCON register may need to be set to 1.

ITx = 0 level activated

ITx = 1 transition activated

IE: INTERRUPT ENABLE REGISTER. BIT ADDRESSABLE.

If the bit is 0, the corresponding interrupt is disabled. If the bit is 1, the corresponding interrupt is enabled.

EA	–	–	ES	ET1	EX1	ET0	EX0

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	Not implemented, reserved for future use.*
—	IE.5	Not implemented, reserved for future use.*
ES	IE.4	Enable or disable the serial port interrupt.
ET1	IE.3	Enable or disable the Timer 1 overflow interrupt.
EX1	IE.2	Enable or disable External Interrupt 1.
ET0	IE.1	Enable or disable the Timer 0 overflow interrupt.
EX0	IE.0	Enable or disable External Interrupt 0.

* User software should not write 1s to reserved bits. These bits may be used in future 80C51 products to invoke new features.

ASSIGNING HIGHER PRIORITY TO ONE OR MORE INTERRUPTS:

In order to assign higher priority to an interrupt the corresponding bit in the IP register must be set to 1.

Remember that while an interrupt service is in progress, it cannot be interrupted by a lower or same level interrupt.

PRIORITY WITHIN LEVEL:

Priority within level is only to resolve simultaneous requests of the same priority level.

From high to low, interrupt sources are listed below:

IE0
TF0
IE1
TF1
RI or TI

IP: INTERRUPT PRIORITY REGISTER. BIT ADDRESSABLE.

If the bit is 0, the corresponding interrupt has a lower priority and if the bit is 1 the corresponding interrupt has a higher priority.

–	–	–	PS	PT1	PX1	PT0	PX0

–	IP.7	Not implemented, reserved for future use.*
–	IP.6	Not implemented, reserved for future use.*
–	IP.5	Not implemented, reserved for future use.*
PS	IP.4	Defines the Serial Port interrupt priority level.
PT1	IP.3	Defines the Timer 1 interrupt priority level.
PX1	IP.2	Defines External Interrupt 1 priority level.
PT0	IP.1	Defines the Timer 0 interrupt priority level.
PX0	IP.0	Defines the External Interrupt 0 priority level.

* User software should not write 1s to reserved bits. These bits may be used in future 80C51 products to invoke new features.

TCON: TIMER/COUNTER CONTROL REGISTER. BIT ADDRESSABLE.

TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

TF1 TCON.7 Timer 1 overflow flag. Set by hardware when the Timer/Counter 1 overflows. Cleared by hardware as processor vectors to the interrupt service routine.

TR1 TCON.6 Timer 1 run control bit. Set/cleared by software to turn Timer/Counter 1 ON/OFF.

TF0 TCON.5 Timer 0 overflow flag. Set by hardware when the Timer/Counter 0 overflows. Cleared by hardware as processor vectors to the service routine.

TR0 TCON.4 Timer 0 run control bit. Set/cleared by software to turn Timer/Counter 0 ON/OFF.

IE1 TCON.3 External Interrupt 1 edge flag. Set by hardware when External Interrupt edge is detected. Cleared by hardware when interrupt is processed.

IT1 TCON.2 Interrupt 1 type control bit. Set/cleared by software to specify falling edge/low level triggered External Interrupt.

IE0 TCON.1 External Interrupt 0 edge flag. Set by hardware when External Interrupt edge detected. Cleared by hardware when interrupt is processed.

IT0 TCON.0 Interrupt 0 type control bit. Set/cleared by software to specify falling edge/low level triggered External Interrupt.

TMOD: TIMER/COUNTER MODE CONTROL REGISTER. NOT BIT ADDRESSABLE.

GATE	C/$\overline{T}$	M1	M0	GATE	C/$\overline{T}$	M1	M0
Timer 1				Timer 0			

GATE When TRx (in TCON) is set and GATE = 1, TIMER/COUNTERx will run only while INTx pin is high (hardware control). When GATE = 0, TIMER/COUNTERx will run only while TRx = 1 (software control).

C/$\overline{T}$ Timer or Counter selector. Cleared for Timer operation (input from internal system clock). Set for Counter operation (input from Tx input pin).

M1 Mode selector bit. (NOTE 1)

M0 Mode selector bit. (NOTE 1)

NOTE 1:

M1	M0	Operating Mode	
0	0	0	13-bit Timer (8048 compatible)
0	1	1	16-bit Timer/Counter
1	0	2	8-bit Auto-Reload Timer/Counter
1	1	3	(Timer 0) TL0 is an 8-bit Timer/Counter controlled by the standart Timer 0 control bits. TH0 is an8-bit Timer and is controlled by Timer 1 control bits.
1	1	3	(Timer 1) Timer/Counter 1 stopped.

TIMER SET-UP

Tables 2 through 5 give some values for TMOD which can be used to set up Timer 0 in different modes.

It is assumed that only one timer is being used at a time. If it is desired to run Timers 0 and 1 simultaneously, in any mode, the value in TMOD for Timer 0 must be ORed with the value shown for Timer 1 (Tables 5 and 6).

For example, if it is desired to run Timer 0 in mode 1 GATE (external control), and Timer 1 in mode 2 COUNTER, then the value that must be loaded into TMOD is 69H (09H from Table 2 ORed with 60H from Table 5).

Moreover, it is assumed that the user, at this point, is not ready to turn the timers on and will do that at a different point in the program by setting bit TRx (in TCON) to 1.

TIMER/COUNTER 0

Table 2. As a Timer:

MODE	TIMER 0 FUNCTION	TMOD	
		INTERNAL CONTROL (NOTE 1)	EXTERNAL CONTROL (NOTE 2)
0	13-bit Timer	00H	08H
1	16-bit Timer	01H	09H
2	8-bit Auto-Reload	02H	0AH
3	Two 8-bit Timers	03H	0BH

Table 3. As a Counter:

MODE	COUNTER 0 FUNCTION	TMOD	
		INTERNAL CONTROL (NOTE 1)	EXTERNAL CONTROL (NOTE 2)
0	13-bit Timer	04H	0CH
1	16-bit Timer	05H	0DH
2	8-bit Auto-Reload	06H	0EH
3	One 8-bit Counter	07H	0FH

NOTES:
1. The timer is turned ON/OFF by setting/clearing bit TR0 in the software.
2. The Timer is turned ON/OFF by the 1-to-0 transition on $\overline{\text{INT0}}$ (P3.2) when TR0 = 1 (hardware control).

TIMER/COUNTER 1

Table 4. As a Timer:

MODE	TIMER 1 FUNCTION	TMOD	
		INTERNAL CONTROL (NOTE 1)	EXTERNAL CONTROL (NOTE 2)
0	13-bit Timer	00H	80H
1	16-bit Timer	10H	90H
2	8-bit Auto-Reload	20H	A0H
3	Does not run	30H	B0H

Table 5. As a Counter:

MODE	COUNTER 1 FUNCTION	TMOD	
		INTERNAL CONTROL (NOTE 1)	EXTERNAL CONTROL (NOTE 2)
0	13-bit Timer	40H	C0H
1	16-bit Timer	50H	D0H
2	8-bit Auto-Reload	60H	E0H
3	Not available	–	–

NOTES:

1. The timer is turned ON/OFF by setting/clearing bit TR1 in the software.
2. The Timer is turned ON/OFF by the 1-to-0 transition on $\overline{\text{INT1}}$ (P3.2) when TR1 = 1 (hardware control).

SCON: SERIAL PORT CONTROL REGISTER. BIT ADDRESSABLE.

SM0	SM1	SM2	REN	TB8	RB8	TI	RI

SM0	SCON.7	Serial Port mode specifier. (NOTE 1)
SM1	SCON.6	Serial Port mode specifier. (NOTE 1)
SM2	SCON.5	Enables the multiprocessor communication feature in modes 2 & 3. In mode 2 or 3, if SM2 is set to 1 then RI will not be activated if the received 9th data bit (RB8) is 0. In mode 1, if SM2 = 1 then RI will not be activated if a valid stop bit was not received. In mode 0, SM2 should be 0. (See Table 6.)
REN	SCON.4	Set/Cleared by software to Enable/Disable reception.
TB8	SCON.3	The 9th bit that will be transmitted in modes 2 & 3. Set/Cleared by software.
RB8	SCON.2	In modes 2 & 3, is the 9th data bit that was received. In mode 1, if SM2 = 0, RB8 is the stop bit that was received. In mode 0, RB8 is not used.
TI	SCON.1	Transmit interrupt flag. Set by hardware at the end of the 8th bit time in mode 0, or at the beginning of the stop bit in the other modes. Must be cleared by software.
RI	SCON.0	Receive interrupt flag. Set by hardware at the end of the 8th bit time in mode 0, or halfway through the stop bit time in the other modes (except see SM2). Must be cleared by software.

NOTE 1:

SM0	SM1	Mode	Description	Baud Rate
0	0	0	Shift Register	$F_{OSC.}/12$
0	1	1	8-bit UART	Variable
1	0	2	9-bit UART	$F_{OSC.}/64$ or $F_{OSC.}/32$
1	1	3	9-bit UART	Variable

SERIAL PORT SET-UP:

Table 6.

MODE	SCON	SM2 VARIATION
0 1 2 3	10H 50H 90H D0H	Single Processor Environment (SM2 = 0)
0 1 2 3	NA 70H B0H F0H	Multiprocessor Environment (SM2 = 1)

GENERATING BAUD RATES

Serial Port in Mode 0:

Mode 0 has a fixed baud rate which is 1/12 of the oscillator frequency. To run the serial port in this mode none of the Timer/Counters need to be set up. Only the SCON register needs to be defined.

$$\text{Baud Rate} = \frac{\text{Osc Freq}}{12}$$

Serial Port in Mode 1:

Mode 1 has a variable baud rate. The baud rate is generated by Timer 1.

USING TIMER/COUNTER 1 TO GENERATE BAUD RATES:

For this purpose, Timer 1 is used in mode 2 (Auto-Reload). Refer to Timer Setup section of this chapter.

$$\text{Baud Rate} = \frac{K \times \text{Osc Freq}}{32 \times 12 \times [256 - (TH1)]}$$

If SMOD = 0, then K = 1.
If SMOD = 1, then K = 2 (SMOD is in the PCON register).

Most of the time the user knows the baud rate and needs to know the reload value for TH1.

$$TH1 = 256 - \frac{K \times \text{Osc Freq}}{384 \times \text{baud rate}}$$

TH1 must be an integer value. Rounding off TH1 to the nearest integer may not produce the desired baud rate. In this case, the user may have to choose another crystal frequency.

Since the PCON register is not bit addressable, one way to set the bit is logical ORing the PCON register (i.e., ORL PCON,#80H). The address of PCON is 87H.

SERIAL PORT IN MODE 2:

The baud rate is fixed in this mode and is 1/32 or 1/64 of the oscillator frequency, depending on the value of the SMOD bit in the PCON register.

In this mode none of the Timers are used and the clock comes from the internal phase 2 clock.

SMOD = 1, Baud Rate = 1/32 Osc Freq.

SMOD = 0, Baud Rate = 1/64 Osc Freq.

To set the SMOD bit: ORL PCON,#80H. The address of PCON is 87H.

SERIAL PORT IN MODE 3:

The baud rate in mode 3 is variable and sets up exactly the same as in mode 1.

80C51 FAMILY INSTRUCTION SET

Table 7. 80C51 Instruction Set Summary

Interrupt Response Time: Refer to Hardware Description Chapter.

Instructions that Affect Flag Settings(1)

Instruction	Flag C	Flag OV	Flag AC	Instruction	Flag C	Flag OV	Flag AC
ADD	X	X	X	CLR C	0		
ADDC	X	X	X	CPL C	X		
SUBB	X	X	X	ANL C,bit	X		
MUL	0	X		ANL C,/bit	X		
DIV	0	X		ORL C,bit	X		
DA	X			ORL C,/bit	X		
RRC	X			MOV C,bit	X		
RLC	X			CJNE	X		
SETB C	1						

(1)Note that operations on SFR byte address 208 or bit addresses 209-215 (i.e., the PSW or bits in the PSW) will also affect flag settings.

Notes on instruction set and addressing modes:

Rn Register R7-R0 of the currently selected Register Bank.

direct 8-bit internal data location's address. This could be an Internal Data RAM location (0-127) or a SFR [i.e., I/O port, control register, status register, etc. (128-255)].

@Ri 8-bit internal data RAM location (0-255) addressed indirectly through register R1 or R0.

#data 8-bit constant included in the instruction.

#data 16 16-bit constant included in the instruction

addr 16 16-bit destination address. Used by LCALL and LJMP. A branch can be anywhere within the 64k-byte Program Memory address space.

addr 11 11-bit destination address. Used by ACALL and AJMP. The branch will be within the same 2k-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.

bit Direct Addressed bit in Internal Data RAM or Special Function Register.

MNEMONIC		DESCRIPTION	BYTE	OSCILLATOR PERIOD
ARITHMETIC OPERATIONS				
ADD	A,Rn	Add register to Accumulator	1	12
ADD	A,direct	Add direct byte to Accumulator	2	12
ADD	A,@Ri	Add indirect RAM to Accumulator	1	12
ADD	A,#data	Add immediate data to Accumulator	2	12
ADDC	A,Rn	Add register to Accumulator with carry	1	12
ADDC	A,direct	Add direct byte to Accumulator with carry	2	12
ADDC	A,@Ri	Add indirect RAM to Accumulator with carry	1	12
ADDC	A,#data	Add immediate data to A_{CC} with carry	2	12
SUBB	A,Rn	Subtract Register from A_{CC} with borrow	1	12
SUBB	A,direct	Subtract direct byte from A_{CC} with borrow	2	12
SUBB	A,@Ri	Subtract indirect RAM from A_{CC} with borrow	1	12
SUBB	A,#data	Subtract immediate data from A_{CC} with borrow	2	12
INC	A	Increment Accumulator	1	12
INC	Rn	Increment register	1	12

All mnemonics copyrighted © Intel Corporation 1980

Table 7. 80C51 Instruction Set Summary (Continued)

MNEMONIC		DESCRIPTION	BYTE	OSCILLATOR PERIOD
ARITHMETIC OPERATIONS (Continued)				
INC	direct	Increment direct byte	2	12
INC	@Ri	Increment indirect RAM	1	12
DEC	A	Decrement Accumulator	1	12
DEC	Rn	Decrement Register	1	12
DEC	direct	Decrement direct byte	2	12
DEC	@Ri	Decrement indirect RAM	1	12
INC	DPTR	Increment Data Pointer	1	24
MUL	AB	Multiply A and B	1	48
DIV	AB	Divide A by B	1	48
DA	A	Decimal Adjust Accumulator	1	12
LOGICAL OPERATIONS				
ANL	A,Rn	AND Register to Accumulator	1	12
ANL	A,direct	AND direct byte to Accumulator	2	12
ANL	A,@Ri	AND indirect RAM to Accumulator	1	12
ANL	A,#data	AND immediate data to Accumulator	2	12
ANL	direct,A	AND Accumulator to direct byte	2	12
ANL	direct,#data	AND immediate data to direct byte	3	24
ORL	A,Rn	OR register to Accumulator	1	12
ORL	A,direct	OR direct byte to Accumulator	2	12
ORL	A,@Ri	OR indirect RAM to Accumulator	1	12
ORL	A,#data	OR immediate data to Accumulator	2	12
ORL	direct,A	OR Accumulator to direct byte	2	12
ORL	direct,#data	OR immediate data to direct byte	3	24
XRL	A,Rn	Exclusive-OR register to Accumulator	1	12
XRL	A,direct	Exclusive-OR direct byte to Accumulator	2	12
XRL	A,@Ri	Exclusive-OR indirect RAM to Accumulator	1	12
XRL	A,#data	Exclusive-OR immediate data to Accumulator	2	12
XRL	direct,A	Exclusive-OR Accumulator to direct byte	2	12
XRL	direct,#data	Exclusive-OR immediate data to direct byte	3	24
CLR	A	Clear Accumulator	1	12
CPL	A	Complement Accumulator	1	12
RL	A	Rotate Accumulator left	1	12
RLC	A	Rotate Accumulator left through the carry	1	12
RR	A	Rotate Accumulator right	1	12
RRC	A	Rotate Accumulator right through the carry	1	12
SWAP	A	Swap nibbles within the Accumulator	1	12
DATA TRANSFER				
MOV	A,Rn	Move register to Accumulator	1	12
MOV	A,direct	Move direct byte to Accumulator	2	12
MOV	A,@Ri	Move indirect RAM to Accumulator	1	12

Table 7. 80C51 Instruction Set Summary (Continued)

MNEMONIC		DESCRIPTION	BYTE	OSCILLATOR PERIOD
DATA TRANSFER (Continued)				
MOV	A,#data	Move immediate data to Accumulator	2	12
MOV	Rn,A	Move Accumulator to register	1	12
MOV	Rn,direct	Move direct byte to register	2	24
MOV	RN,#data	Move immediate data to register	2	12
MOV	direct,A	Move Accumulator to direct byte	2	12
MOV	direct,Rn	Move register to direct byte	2	24
MOV	direct,direct	Move direct byte to direct	3	24
MOV	direct,@Ri	Move indirect RAM to direct byte	2	24
MOV	direct,#data	Move immediate data to direct byte	3	24
MOV	@Ri,A	Move Accumulator to indirect RAM	1	12
MOV	@Ri,direct	Move direct byte to indirect RAM	2	24
MOV	@Ri,#data	Move immediate data to indirect RAM	2	12
MOV	DPTR,#data16	Load Data Pointer with a 16-bit constant	3	24
MOVC	A,@A+DPTR	Move Code byte relative to DPTR to A_{CC}	1	24
MOVC	A,@A+PC	Move Code byte relative to PC to A_{CC}	1	24
MOVX	A,@Ri	Move external RAM (8-bit addr) to A_{CC}	1	24
MOVX	A,@DPTR	Move external RAM (16-bit addr) to A_{CC}	1	24
MOVX	A,@Ri,A	Move A_{CC} to external RAM (8-bit addr)	1	24
MOVX	@DPTR,A	Move A_{CC} to external RAM (16-bit addr)	1	24
PUSH	direct	Push direct byte onto stack	2	24
POP	direct	Pop direct byte from stack	2	24
XCH	A,Rn	Exchange register with Accumulator	1	12
XCH	A,direct	Exchange direct byte with Accumulator	2	12
XCH	A,@Ri	Exchange indirect RAM with Accumulator	1	12
XCHD	A,@Ri	Exchange low-order digit indirect RAM with A_{CC}	1	12
BOOLEAN VARIABLE MANIPULATION				
CLR	C	Clear carry	1	12
CLR	bit	Clear direct bit	2	12
SETB	C	Set carry	1	12
SETB	bit	Set direct bit	2	12
CPL	C	Complement carry	1	12
CPL	bit	Complement direct bit	2	12
ANL	C,bit	AND direct bit to carry	2	24
ANL	C,/bit	AND complement of direct bit to carry	2	24
ORL	C,bit	OR direct bit to carry	2	24
ORL	C,/bit	OR complement of direct bit to carry	2	24
MOV	C,bit	Move direct bit to carry	2	12
MOV	bit,C	Move carry to direct bit	2	24
JC	rel	Jump if carry is set	2	24
JNC	rel	Jump if carry not set	2	24

Table 7. 80C51 Instruction Set Summary (Continued)

MNEMONIC		DESCRIPTION	BYTE	OSCILLATOR PERIOD
BOOLEAN VARIABLE MANIPULATION (Continued)				
JB	rel	Jump if direct bit is set	3	24
JNB	rel	Jump if direct bit is not set	3	24
JBC	bit,rel	Jump if direct bit is set and clear bit	3	24
PROGRAM BRANCHING				
ACALL	addr11	Absolute subroutine call	2	24
LCALL	addr16	Long subroutine call	3	24
RET		Return from subroutine	1	24
RETI		Return from interrupt	1	24
AJMP	addr11	Absolute jump	2	24
LJMP	addr16	Long jump	3	24
SJMP	rel	Short jump (relative addr)	2	24
JMP	@A+DPTR	Jump indirect relative to the DPTR	1	24
JZ	rel	Jump if Accumulator is zero	2	24
JNZ	rel	Jump if Accumulator is not zero	2	24
CJNE	A,direct,rel	Compare direct byte to A_{CC} and jump if not equal	3	24
CJNE	A,#data,rel	Compare immediate to A_{CC} and jump if not equal	3	24
CJNE	RN,#data,rel	Compare immediate to register and jump if not equal	3	24
CJNE	@Ri,#data,rel	Compare immediate to indirect and jump if not equal	3	24
DJNZ	Rn,rel	Decrement register and jump if not zero	2	24
DJNZ	direct,rel	Decrement direct byte and jump if not zero	3	24
NOP		No operation	1	12

INSTRUCTION DEFINITIONS

ACALL addr11

Function: Absolute Call

Description: ACALL unconditionally calls a subroutine located at the indicated address. The instruction increments the PC twice to obtain the address of the following instruction, then pushes the 16-bit result onto the stack (low-order byte first) and increments the Stack Pointer twice. The destination address is obtained by successively concatenating the five high-order bits of the incremented PC, opcode bits 7-5, and the second byte of the instruction. The subroutine called must therefore start within the same 2k block of the program memory as the first byte of the instruction following ACALL. No flags are affected.

Example: Initially SP equals 07H. The label "SUBRTN" is at program memory location 0345 H. After executing the instruction,

ACALL SUBRTN

at location 0123H, SP will contain 09H, internal RAM locations 08H and 09H will contain 25H and 01H, respectively, and the PC will contain 0345H.

Bytes: 2

Cycles: 2

Encoding:

a10 a9 a8 1	0 0 0 1		a7 a6 a5 a4	a3 a2 a1 a0

Operation: ACALL
$(PC) \leftarrow (PC) + 2$
$(SP) \leftarrow (SP) + 1$
$((SP)) \leftarrow (PC_{7-0})$
$(SP) \leftarrow (SP) + 1$
$((SP)) \leftarrow (PC_{15-8})$
$(PC_{10-0}) \leftarrow$ page address

ADD A,<src-byte>

Function: Add

Description: ADD adds the byte variable indicated to the Accumulator, leaving the result in the Accumulator. The carry and auxiliary-carry flags are set, respectively, if there is a carry-out from bit 7 or bit 3, and cleared otherwise. When adding unsigned integers, the carry flag indicates an overflow occurred.

OV is set if there is a carry-out of bit 6 but not out of bit 7, or a carry-out of bit 7 but not bit 6; otherwise OV is cleared. When adding signed integers, OV indicates a negative number produced as the sum of two positive operands, or a positive sum from two negative operands.

Four source operand addressing modes are allowed: register, direct, register-indirect, or immediate.

Example: The Accumulator holds 0C3H (11000011B) and register 0 holds 0AAH (10101010B). The instruction,

ADD A,R0

will leave 6DH (01101101B) in the Accumulator with the AC flag cleared and both the Carry flag and OV set to 1.

ADD A,Rn

Bytes: 1

Cycles: 1

Encoding:

0 0 1 0	1 r r r

Operation: ADD
(A) ← (A) + (R_n)

ADD A,direct

Bytes: 2

Cycles: 1

Encoding:

0 0 1 0	0 1 0 1	direct address

Operation: ADD
(A) ← (A) + (direct)

ADD A,@Ri

Bytes: 1

Cycles: 1

Encoding:

0 0 1 0	0 1 1 i

Operation: ADD
(A) ← (A) + $((R_i))$

ADD A,#data

Bytes: 2

Cycles: 1

Encoding:

0 0 1 0	0 1 0 0	immediate data

Operation: ADD
(A) ← (A) + #data

ADDC A,<src-byte>

Function: Add with Carry

Description: ADDC simultaneously adds the byte variable indicated, the carry flag and the Accumulator contents, leaving the result in the Accumulator. The carry and auxiliary-carry flags are set, respectively, if there is a carry-out from bit 7 or bit 3, and cleared otherwise. When adding unsigned integers, the carry flag indicates an overflow occurred.

OV is set if there is a carry-out of bit 6 but not out of bit 7, or a carry-out of bit 7 but not out of bit 6; otherwise OV is cleared. When adding signed integers, OV indicates a negative number produced as the sum of two positive operands, or a positive sum from two negative operands.

Four source operand addressing modes are allowed: register, direct, register-indirect, or immediate.

Example: The Accumulator holds 0C3H (11000011B) and register 0 holds 0AAH (10101010B) with the carry flag set. The instruction,

ADDC A,R0

will leave 6EH (01101110B) in the Accumulator with AC cleared and both the Carry flag and OV set to 1.

ADDC A,Rn

Bytes: 1

Cycles: 1

Encoding:

0 0 1 1	1 r r r

Operation: ADDC
(A) ← (A) + (C) + (R_n)

ADDC A,direct

Bytes: 2

Cycles: 1

Encoding:

0 0 1 1	0 1 0 1	direct address

Operation: ADDC
(A) ← (A) + (C) + (direct)

ADDC A,@Ri

Bytes: 1

Cycles: 1

Encoding:

0 0 1 1	0 1 1 i

Operation: ADDC
(A) ← (A) + (C) + ((R_i))

ADDC A,#data

Bytes: 2

Cycles: 1

Encoding:

0 0 1 1	0 1 0 0	immediate data

Operation: ADDC
(A) ← (A) + (C) + #data

AJMP addr11

Function: Absolute Jump

Description: AJMP transfers program execution to the indicated address, which is formed at run-time by concatenating the high-order five bits of the PC (*after* incrementing the PC twice), opcode bits 7-5, and the second byte of the instruction. The destination must therefore be within the same 2k block of program memory as the first byte of the instruction following AJMP.

Example: The label "JMPADR" is at program memory location 0123H. The instruction,

AJMP JMPADR

is at location 0345H and will load the PC with 0123H.

Bytes: 2

Cycles: 2

Encoding:

a10 a9 a8 0	0 0 0 1		a7 a6 a5 a4	a3 a2 a1 a0

Operation: AJMP
(PC) ← (PC) + 2
$(PC_{10\text{-}0})$ ← page address

ANL <dest-byte>,<src-byte>

Function: Logical-AND for byte variables

Description: ANL performs the bitwise logical-AND operation between the variables indicated and stores the results in the destination variable. No flags are affected.

The two operands allow six addressing mode combinations. When the destination is the Accumulator, the source can use register, direct, register-indirect, or immediate addressing; when the destination is a direct address, the source can be the Accumulator or immediate data.

Note: When this instruction is used to modify an output port, the value used as the original port data will be read from the output data latch, *not* the input pins.

Example: If the Accumulator holds 0C3H (11000011B) and register 0 holds 55H (01010101B) then the instruction,

ANL A,R0

will leave 41H (01000001B) in the Accumulator.

When the destination is a directly addressed byte, this instruction will clear combinations of bits in any RAM location or hardware register. The mask byte determining the pattern of bits to be cleared would either be a constant contained in the instruction or a value computed in the Accumulator at run-time. The instruction,

ANL P1,#01110011B

will clear bits 7, 3, and 2 of output port 1.

ANL A,Rn

Bytes: 1

Cycles: 1

Encoding:

0 1 0 1	1 r r r

Operation: ANL
$(A) \leftarrow (A) \wedge (R_n)$

ANL A,direct

Bytes: 2

Cycles: 1

Encoding:

0 1 0 1	0 1 0 1	direct address

Operation: ANL
$(A) \leftarrow (A) \wedge (\text{direct})$

ANL A,@Ri

Bytes: 1

Cycles: 1

Encoding:

0 1 0 1	0 1 1 i

Operation: ANL
$(A) \leftarrow (A) \wedge ((R_i))$

ANL A,#data

Bytes: 2

Cycles: 1

Encoding:

0 1 0 1	0 1 0 0	immediate data

Operation: ANL
$(A) \leftarrow (A) \wedge \#\text{data}$

ANL direct,A

Bytes: 2

Cycles: 1

Encoding:

0 1 0 1	0 0 1 0	direct address

Operation: ANL
$(A) \leftarrow (\text{direct}) \wedge (A)$

ANL direct,#data

Bytes: 3

Cycles: 2

Encoding:

0 1 0 1	0 0 1 1	direct address	immediate data

Operation: ANL
$(\text{direct}) \leftarrow (\text{direct}) \wedge \#\text{data}$

ANL C,<src-bit>

Function: Logical-AND for bit variables

Description: If the Boolean value of the source bit is a logical 0 then clear the carry flag; otherwise leave the carry flag in its current state. A slash ("/") preceding the operand in the assembly language indicates that the logical complement of the addressed bit is used as the source value, *but the source bit itself is not affected.* No other flags are affected.

Only direct addressing is allowed for the source operand.

Example: Set the carry flag if, and only if, P1.0 = 1, ACC.7 = 1, and OV = 0:

```
MOV    C,P1.0  ;LOAD CARRY WITH INPUT PIN STATE
ANL    C,ACC.7;AND CARRY WITH ACCUM. BIT 7
ANL    C,/OV   ;AND WITH INVERSE OF OVERFLOW FLAG
```

ANL C,bit

Bytes: 2

Cycles: 2

Encoding:

1 0 0 0	0 0 1 0	bit address

Operation: ANL

$(C) \leftarrow (C) \wedge (bit)$

ANL C,/bit

Bytes: 2

Cycles: 2

Encoding:

1 0 1 1	0 0 0 0	bit address

Operation: ANL

$(C) \leftarrow (C) \wedge \neg (bit)$

CJNE <dest-byte>,<src-byte>,rel

Function: Compare and Jump if Not Equal

Description: CJNE compares the magnitudes of the first two operands, and branches if their values are not equal. The branch destination is computed by adding the signed relative-displacement in the last instruction byte to the PC, after incrementing the PC to the start of the next instruction. The carry flag is set if the unsigned integer value of <dest-byte> is less than the unsigned integer value of <src-byte>; otherwise, the carry is cleared. Neither operand is affected.

The first two operands allow four addressing mode combinations: the Accumulator may be compared with any directly addressed byte or immediate data, and any indirect RAM location or working register can be compared with an immediate constant.

Example: The Accumulator contains 34H. Register 7 contains 56H. The first instruction in the sequence,

```
            CJNE     R7,#60H,NOT_EQ
;           ...      ....          ;       R7 = 60H.
NOT_EQ      JC       REQ_LOW       ;       IF R7 < 60H.
;           ...      ....          ;       R7 > 60H.
```

sets the carry flag and branches to the instruction at label NOT_EQ. By testing the carry flag, this instruction determines whether R7 is greater or less than 60H.

If the data being presented to Port 1 is also 34H, then the instruction,

```
WAIT:  CJNE  A,P1,WAIT
```

clears the carry flag and continues with the next instruction in sequence, since the Accumulator does equal the data read from P1. (If some other value was being input on P1, the program will loop at this point until the P1 data changes to 34H.)

CJNE A,direct,rel

Bytes: 3

Cycles: 2

Encoding:

1 0 1 1	0 1 0 1	direct address	rel. address

Operation:

```
(PC) ← (PC) + 3
IF (A) < > (direct)
THEN
            (PC) ← (PC) + relative offset
IF (A) <  (direct)
THEN.
            (C) ← 1
ELSE
            (C) ← 0
```

CJNE A,#data,rel

Bytes: 3

Cycles: 2

Encoding:

1 0 1 1	0 1 0 0	immediate data	rel. address

Operation:

```
(PC) ← (PC) + 3
IF (A) < > data
THEN
        (PC) ← (PC) + relative offset
IF (A) <  data
THEN
        (C) ← 1
ELSE
        (C) ← 0
```

CJNE Rn,#data,rel

Bytes: 3

Cycles: 2

Encoding:

1 0 1 1	1 r r r	immediate data	rel. address

Operation:

```
(PC) ← (PC) + 3
IF ($R_n$) < > data
THEN
        (PC) ← (PC) + relative offset
IF ($R_n$) <  data
THEN
        (C) ← 1
ELSE
        (C) ← 0
```

CJNE @Ri,#data,rel

Bytes: 3

Cycles: 2

Encoding:

1 0 1 1	0 1 1 i	immediate data	rel. address

Operation:

```
(PC) ← (PC) + 3
IF (($R_i$)) < > data
THEN
        (PC) ← (PC) + relative offset
IF (($R_i$)) <  data
THEN
        (C) ← 1
ELSE
        (C) ← 0
```

CLR A

Function: Clear Accumulator

Description: The Accumulator is cleared (all bits reset to zero). No flags are affected.

Example: The Accumulator contains 5CH (01011100B). The instruction,

CLR A

will leave the Accumulator set to 00H (00000000B).

Bytes: 1

Cycles: 1

Encoding:

1 1 1 0	0 1 0 0

Operation: CLR
(A) ← 0

CLR bit

Function: Clear bit

Description: The indicated bit is cleared (reset to zero). No other flags are affected. CLR can operate on the carry flag or any directly addressable bit.

Example: Port 1 has previously been written with 5DH (01011101B). The instruction,

CLR P1.2

will leave the port set to 59H (01011001B).

CLR C

Bytes: 1

Cycles: 1

Encoding:

1 1 0 0	0 0 1 1

Operation: CLR
(C) ← 0

CLR bit

Bytes: 2

Cycles: 1

Encoding:

1 1 0 0	0 0 1 0	bit address

Operation: CLR
(bit) ← 0

CPL A

Function: Complement Accumulator

Description: Each bit of the Accumulator is logically complemented (one's complement). Bits which previously contained a one are changed to a zero and vice-versa. No flags are affected.

Example: The Accumulator contains 5CH (01011100B). The instruction,

CPL A

will leave the Accumulator set to 0A3H (10100011B).

Bytes: 1

Cycles: 1

Encoding:

1 1 1 1	0 1 0 0

Operation: CPL
(A) ← ⌉ (A)

CPL bit

Function: Complement bit

Description: The bit variable specified is complemented. A bit which had been a one is changed to zero and vice-versa. No other flags are affected. CPL can operate on the carry or any directly addressable bit.

Note: When this instruction is used to modify an output pin, the value used as the original data will be read from the output data latch, *not* the input pin.

Example: Port 1 has previously been written with 5DH (01011101B). The instruction sequence,

CPL P1.1

CPL P1.2

will leave the port set to 5BH (01011011B).

CPL C

Bytes: 1

Cycles: 1

Encoding:

1 0 1 1	0 0 1 1

Operation: CPL
(C) ← ⌉ (C)

CPL bit

Bytes: 2

Cycles: 1

Encoding:

1 0 1 1	0 0 1 0	bit address

Operation: CPL
(bit) ← ⌉ (bit)

DA A

Function: Decimal-adjust Accumulator for Addition

Description: DA A adjusts the eight-bit value in the Accumulator resulting from the earlier addition of two variable (each in packed-BCD format), producing two four-bit digits. Any ADD or ADDC instruction may have been used to perform the addition.

If Accumulator bits 3-0 are greater than nine (xxx1010-xxx1111), or if the AC flag is one, six is added to the Accumulator, producing the proper BCD digit in the low-order nibble. This internal addition would set the carry flag if a carry-out of the low-order four-bit field propagated through all high-order bits, but it would not clear the carry flag otherwise.

If the carry flag is now set, or if the four high-order bits now exceed nine (1010xxx-111xxxx), these high-order bits are incremented by six, producing the proper BCD digit in the high-order nibble. Again, this would set the carry flag if there was a carry-out of the high-order bits, but wouldn't clear the carry. The carry flag thus indicates if the sum of the original two BCD variables is greater than 100, allowing multiple precision decimal addition. OV is not affected.

All of this occurs during the one instruction cycle. Essentially, this instruction performs the decimal conversion by adding 00H, 06H, 60H, or 66H to the Accumulator, depending on initial Accumulator and PSW conditions.

Note: DA A *cannot* simply convert a hexadecimal number in the Accumulator to BCD notation, nor does DA A apply to decimal subtraction.

Example: The Accumulator holds the value 56H (01010110B) representing the packed BCD digits of the decimal number 56. Register 3 contains the value 67H (01100111B) representing the packed BCD digits of the decimal number 67. The carry flag is set.. The instruction sequence,

```
ADDC      A,R3
DA        A
```

will first perform a standard two's-complement binary addition, resulting in the value 0BEH (10111110B) in the Accumulator. The carry and auxiliary carry flags will be cleared.

The Decimal Adjust instruction will then alter the Accumulator to the value 24H (00100100B), indicating the packed BCD digits of the decimal number 24, the low-order two digits of the decimal sum of 56, 67, and the carry-in. The carry flag will be set by the Decimal Adjust instruction, indicating that a decimal overflow occurred. The true sum 56, 67, and 1 is 124.

BCD variables can be incremented or decremented by adding 01H or 99H. If the Accumulator initially holds 30H (representing the digits of 30 decimal), the the instruction sequence,

```
ADD     A,#99H
DA      A
```

will leave the carry set and 29H in the Accumulator, since 30 + 99 = 129. The low-order byte of the sum can be interpreted to mean 30 – 1 = 29.

Bytes: 1

Cycles: 1

Encoding:

1 1 0 1	0 1 0 0

Operation:

```
DA
–contents of Accumulator are BCD
IF   $[[(A_{3-0}) > 9] \vee [(AC) = 1]]$
     THEN$(A_{3-0}) \leftarrow (A_{3-0}) + 6$
         AND
IF   $[[(A_{7-4}) > 9] \vee [(C) = 1]]$
     THEN$(A_{7-4}) \leftarrow (A_{7-4}) + 6$
```

DEC byte

Function: Decrement

Description: The variable indicated is decremented by 1. An original value of 00H will underflow to 0FFH. No flags are affected. Four operand addressing modes are allowed: accumulator, register, direct, or register-indirect.

Note: When this instruction is used to modify an output port, the value used as the original data will be read from the output data latch, *not* the input pin.

Example: Register 0 contains 7FH (01111111B). Internal RAM locations 7EH and 7FH contain 00H and 40H, respectively. The instruction sequence,

```
DEC    @R0
DEC    R0
DEC    @R0
```

will leave register 0 set to 7EH and internal RAM locations 7EH and 7FH set to 0FFH and 3FH.

DEC A

Bytes: 1

Cycles: 1

Encoding:

0 0 0 1	0 1 0 0

Operation: DEC
$(A) \leftarrow (A) - 1$

DEC Rn

Bytes: 1

Cycles: 1

Encoding:

0 0 0 1	1 r r r

Operation: DEC
$(R_n) \leftarrow (R_n) - 1$

DEC direct

Bytes: 2

Cycles: 1

Encoding:

0 0 0 1	0 1 0 1	direct address

Operation: DEC
$(\text{direct}) \leftarrow (\text{direct}) - 1$

DEC @Ri

Bytes: 1

Cycles: 1

Encoding:

0 0 0 1	0 1 1 i

Operation: DEC
$((R_i)) \leftarrow ((R_i)) - 1$

DIV AB

Function: Divide

Description: DIV AB divides the unsigned eight-bit integer in the Accumulator by the unsigned eight-bit integer in register B.

The Accumulator receives the integer part of the quotient; register B receives the integer remainder. The carry and OV flags will be cleared.

Exception: if B had originally contained 00H, the values returned in the Accumulator and B-register will be undefined and the overflow flag will be set. The carry flag is cleared in any case.

Example: The Accumulator contains 251 (0FBH or 11111011B) and B contains 18 (12H or 00010010B). The instruction,

DIV AB

will leave 13 in the Accumulator (0DH or 00001101B) and the value 17 (11H or 00010001B) in B, since 251 = (13 x 18) + 17. Carry and OV will both be cleared.

Bytes: 1

Cycles: 4

Encoding:

1 0 0 0	0 1 0 0

Operation: DIV

$(A)_{15-8} \leftarrow (A)/(B)$

$(B)_{7-0}$

DJNZ <byte>,<rel-addr>

Function: Decrement and Jump if Not Zero

Description: DJNZ decrements the location indicated by 1, and branches to the address indicated by the second operand if the resulting value is not zero. An original value of 00H will underflow to 0FFH. No flags are affected. The branch destination would be computed by adding the signed relative-displacement value in the last instruction byte to the PC, after incrementing the PC to the first byte of the following instruction.

The location decremented may be a register or directly addressed byte.

Note: When this instruction is used to modify an output port, the value used as the original port data will be read from the output data latch, *not* the input pins.

Example: Internal RAM locations 40H, 50H, and 60H contain the values 01H, 70H, and 15H, respectively. The instruction sequence,

```
DJNZ    40H,LABEL_1
DJNZ    50H,LABEL_2
DJNZ    60H,LABEL_3
```

will cause a jump to the instruction at LABEL_2 with the values 00h, 6FH, and 15H in the three RAM locations. The first jump was *not* taken because the result was zero.

This instruction provides a simple was of executing a program loop a given number of times, or for adding a moderate time delay (from 2 to 512 machine cycles) with a single instruction. The instruction sequence,

```
         MOV     R2,#8
TOGGLE:  CPL     P1.7
         DJNZ    R2,TOGGLE
```

will toggle P1.7 eight times, causing four output pulses to appear at bit 7 of output Port 1. Each pulse will last three machine cycles, two for DJNZ and one to alter the pin.

DJNZ Rn,rel

Bytes: 2

Cycles: 2

Encoding:

1 1 0 1	1 r r r	rel. address

Operation:

```
DJNZ
(PC) ← (PC) + 2
$(R_n)$ ← $(R_n)$ – 1
IF $(R_n)$ > 0 or $(R_n)$ < 0
    THEN
            (PC) ← (PC) + rel
```

DJNZ direct,rel

Bytes: 3

Cycles: 2

Encoding:

1 1 0 1	0 1 0 1	direct data	rel. address

Operation:

```
DJNZ
(PC) ← (PC) + 2
(direct) ← (direct) – 1
IF (direct) > 0 or (direct) < 0
    THEN
        (PC) ← (PC) + rel
```

INC <byte>

Function: Increment

Description: INC increments the indicated variable by 1. An original value of 0FFH will overflow to 00H. No flags are affected. Three addressing modes are allowed: register, direct, or register-indirect.

Note: When this instruction is used to modify an output port, the value used as the original port data will be read from the output data latch, *not* the input pins.

Example: Register 0 contains 7EH (01111110B). Internal RAM locations 7EH and 7FH contain 0FFH and 40H, respectively. The instruction sequence,

```
INC     @R0
INC     R0
INC     @R0
```

will leave register 0 set to 7FH and internal RAM locations 7EH and 7FH holding (respectively) 00H and 41H.

INC A

Bytes: 1

Cycles: 1

Encoding:

0 0 0 0	0 1 0 0

Operation: INC
$(A) \leftarrow (A) + 1$

INC Rn

Bytes: 1

Cycles: 1

Encoding:

0 0 0 0	1 r r r

Operation: INC
$(R_n) \leftarrow (R_n) + 1$

INC direct

Bytes: 2

Cycles: 1

Encoding:

0 0 0 0	0 1 0 1	direct address

Operation: INC
$(direct) \leftarrow (direct) + 1$

INC @Ri

Bytes: 1

Cycles: 1

Encoding:

0 0 0 0	0 1 1 i

Operation: INC
$((R_i)) \leftarrow ((R_i)) + 1$

INC DPTR

Function: Increment Data Pointer

Description: Increment the 16-bit data pointer by 1. A 16-bit increment (modulo 2^{16}) is performed; an overflow of the low-order byte of the data pointer (DPL) from 0FFH to 00H will increment the high-order byte (DPH). No flags are affected.

This is the only 16-bit register which can be incremented.

Example: Registers DPH and DPL contain 12H and 0FEH, respectively. The instruction sequence,

```
INC     DPTR
INC     DPTR
INC     DPTR
```

will change DPH and DPL to 13H and 01H.

Bytes: 1

Cycles: 2

Encoding:

1 0 1 0	0 0 1 1

Operation:

```
INC
(DPTR) ← (DPTR) + 1
```

JB bit,rel

Function: Jump if Bit set

Description: If the indicated bit is a one, jump to the address indicated; otherwise proceed with the next instruction. The branch destination is computed by adding the signed relative-displacement in the third instruction byte to the PC, after incrementing the PC to the first byte of the next instruction. *The bit tested is not modified.* No flags are affected.

Example: The data present at input port 1 is 11001010B. The Accumulator holds 56 (01010110B). The instruction sequence,

```
JB      P1.2,LABEL1
JB      ACC.2,LABEL2
```

will cause program execution to branch to the instruction at label LABEL2.

Bytes: 3

Cycles: 2

Encoding:

0 0 1 0	0 0 0 0	bit address	rel. address

Operation:

```
JB
(PC) ← (PC) + 3
IF (bit) = 1
   THEN
        (PC) ← (PC) + rel
```

JBC bit,rel

Function: Jump if Bit is set and Clear bit

Description: If the indicated bit is a one, branch to the address indicated; otherwise proceed with the next instruction. *The bit will not be cleared if it is already a zero.* The branch destination is computed by adding the signed relative-displacement in the third instruction byte to the PC, after incrementing the PC to the first byte of the next instruction. No flags are affected.

Note: When this instruction is used to test an output pin, the value used as the original data will read from the output data latch, not the input pin.

Example: The Accumulator holds 56H (01010110B). The instruction sequence,

```
JBC     ACC.3,LABEL1
JBC     ACC.2,LABEL2
```

will cause program execution to continue at the instruction identified by the LABEL2, with the Accumulator modified to 52H (01010010B).

Bytes: 3

Cycles: 2

Encoding:

0 0 0 1	0 0 0 0	bit address	rel. address

Operation:

```
JBC
(PC) ← (PC) + 3
IF (bit) = 1
   THEN
(bit) ← 0
(PC) ← (PC) + rel
```

JC rel

Function: Jump if Carry is set

Description: If the carry flag is set, branch to the address indicated; otherwise proceed with the next instruction. The branch destination is computed by adding the signed relative-displacement in the second instruction byte to the PC, after incrementing the PC twice. No flags are affected.

Example: The carry flag is cleared. The instruction sequence,

```
JC      LABEL1
CPL     C
JC      LABEL2
```

will set the carry and cause program execution to continue at the instruction identified by the label LABEL2.

Bytes: 2

Cycles: 2

Encoding:

0 1 0 0	0 0 0 0	rel. address

Operation:

```
JC
(PC) ← (PC) + 2
IF (C) = 1
   THEN
              (PC) ← (PC) + rel
```

JMP @A+DPTR

Function: Jump indirect

Description: Add the eight-bit unsigned contents of the Accumulator with the sixteen-bit data pointer, and load the resulting sum to the program counter. This will be the address for subsequent instruction fetches. Sixteen-bit addition is performed (modulo 2^{16}): a carry-out from the low-order eight bits propagates through the higher-order bits. Neither the Accumulator nor the Data Pointer is altered. No flags are affected.

Example: An even number from 0 to 6 is in the Accumulator. The following sequence of instructions will branch to one of four AJMP instructions in a jump table starting at JMP_TBL:

```
           MOV    DPTR,#JMP_TBL
           JMP    @A+DPTR
JMP_TBL:   AJMP   LABEL0
           AJMP   LABEL1
           AJMP   LABEL2
           AJMP   LABEL3
```

If the Accumulator equals 04H when starting this sequence, execution will jump to label LABEL2. Remember that AJMP is a two-byte instruction, so the jump instructions start at every other address.

Bytes: 1

Cycles: 2

Encoding:

0 1 1 1	0 0 1 1

Operation:

```
JMP
(PC) ← (A) + (DPTR)
```

JNB bit,rel

Function: Jump if Bit Not set

Description: If the indicated bit is a zero, branch to the indicated address; otherwise proceed with the next instruction. The branch destination is computed by adding the signed relative-displacement in the third instruction byte to the PC, after incrementing the PC to the first byte of the next instruction. *The bit tested is not modified.* No flags are affected.

Example: The data present at input port 1 is 11001010B. The Accumulator holds 56H (01010110B). The instruction sequence,

```
JNB    P1.3,LABEL1
JNB    ACC.3,LABEL2
```

will cause program execution to continue at the instruction at label LABEL2.

Bytes: 3

Cycles: 2

Encoding:

0 0 1 1	0 0 0 0	bit address	rel. address

Operation:

```
JNB
(PC) ← (PC) + 3
IF (bit) = 0
   THEN
(PC) ← (PC) + rel
```

JNC rel

Function: Jump if Carry Not set

Description: If the carry flag is a zero, branch to the address indicated; otherwise proceed with the next instruction. The branch destination is computed by adding the signed relative-displacement in the second instruction byte to the PC, after incrementing the PC twice to point to the next instruction. The carry flag is not modified.

Example: The carry flag is set. The instruction sequence,

```
JNC     LABEL1
CPL     C
JNC     LABEL2
```

will clear the carry and cause program execution to continue at the instruction identified by the label LABEL2.

Bytes: 2

Cycles: 2

Encoding:

0 1 0 1	0 0 0 0		rel. address

Operation:

```
JNC
(PC) ← (PC) + 2
IF (C) = 0
   THEN
(PC) ← (PC) + rel
```

JNZ rel

Function: Jump if Accumulator Not Zero

Description: If any bit of the Accumulator is a one, branch to the indicated address; otherwise proceed with the next instruction. The branch destination is computed by adding the signed relative-displacement in the second instruction byte to the PC, after incrementing the PC twice. The Accumulator is not modified. No flags are affected.

Example: The Accumulator originally holds 00H. The instruction sequence,

```
JNZ     LABEL1
INC     A
JNZ     LABEL2
```

will set the Accumulator to 01H and continue at label LABEL2.

Bytes: 2

Cycles: 2

Encoding:

0 1 1 1	0 0 0 0		rel. address

Operation:

```
JNZ
(PC) ← (PC) + 2
IF A ≠ 0
   THEN  (PC) ← (PC) + rel
```

JZ rel

Function: Jump if Accumulator Zero

Description: If all bits of the Accumulator are zero, branch to the indicated address; otherwise proceed with the next instruction. The branch destination is computed by adding the signed relative-displacement in the second instruction byte to the PC, after incrementing the PC twice. The Accumulator is not modified. No flags are affected.

Example: The Accumulator originally holds 01H. The instruction sequence,

```
JZ      LABEL1
DEC     A
JZ      LABEL2
```

will change the Accumulator to 00H and cause program execution to continue at the instruction identified by the label LABEL2.

Bytes: 2

Cycles: 2

Encoding:

0 1 1 0	0 0 0 0	rel. address

Operation:

```
JZ
$(PC) \leftarrow (PC) + 2$
IF $A = 0$
   THEN $(PC) \leftarrow (PC) + rel$
```

LCALL addr16

Function: Long Call

Description: LCALL calls a subroutine located at the indicated address. The instruction adds three to the program counter to generate the address of the next instruction and then pushes the 16-bit result onto the stack (low byte first), incrementing the Stack Pointer by two. The high-order and low-order bytes of the PC are then loaded, respectively, with the second and third bytes of the LCALL instruction. Program execution continues with the instruction at this address. The subroutine may therefore begin anywhere in the full 64k-byte program memory address space. No flags are affected.

Example: Initially the Stack Pointer equals 07H. The label "SUBRTN" is assigned to program memory location 1234H. After executing the instruction,

```
LCALL   SUBRTN
```

at location 0123H, the Stack Pointer will contain 09H, internal RAM locations 08H and 09H will contain 26H and 01H, and the PC will contain 1235H.

Bytes: 3

Cycles: 2

Encoding:

0 0 0 1	0 0 1 0	addr15-addr8	addr7-addr0

Operation:

```
LCALL
$(PC) \leftarrow (PC) + 3$
$(SP) \leftarrow (SP) + 1$
$((SP)) \leftarrow (PC_{7-0})$
$(SP) \leftarrow (SP) + 1$
$((SP)) \leftarrow (PC_{15-8})$
$(PC) \leftarrow addr_{15-0}$
```

LJMP addr16 (Implemented in 87C751 and 87C752 for in-circuit emulation only.)

Function: Long Jump

Description: LJMP causes an unconditional branch to the indicated address, by loading the high-order and low-order bytes of the PC (respectively) with the second and third instruction bytes. The destination may therefore be anywhere in the full 64k program memory address space. No flags are affected.

Example: The label "JMPADR" is assigned to the instruction at program memory location 1234H. The instruction,

```
LJMP    JMPADR
```

at location 0123H will load the program counter with 1234H.

Bytes: 3

Cycles: 2

Encoding:

0 0 0 0	0 0 1 0	addr15-addr8	addr7-addr0

Operation: LJMP
$(PC) \leftarrow addr_{15\text{-}0}$

MOV <dest-byte>,<src-byte>

Function: Move byte variable

Description: The byte variable indicated by the second operand is copied into the location specified by the first operand. The source byte is not affected. No other register or flag is affected.

This is by far the most flexible operation. Fifteen combinations of source and destination addressing modes are allowed.

Example: Internal RAM location 30H holds 40H. The value of RAM location 40H is 10H. The data present at input port 1 is 11001010B (0CAH). The instruction sequence,

```
MOV     R0,#30H     ;R0 < = 30H
MOV     A,@R0       ;A < = 40H
MOV     R1,A        ;R1 < = 40H
MOV     B,@R1       ;B < = 10H
MOV     @R1,P1      ;RAM (40H) < = 0CAH
MOV     P2,P1       ;P2 #0CAH
```

leaves the value 30H in register 0, 40H in both the Accumulator and register 1, 10H in register B, and 0CAH (11001010B) both in RAM location 40H and output on port 2.

MOV A,Rn

Bytes: 1

Cycles: 1

Encoding:

1 1 1 0	1 r r r

Operation: MOV
$(A) \leftarrow (R_n)$

***MOV A,direct**

Bytes: 2

Cycles: 1

Encoding:

1 1 1 0	0 1 0 1	direct address

Operation: MOV
(A) ← (direct)

MOV A,@Ri

Bytes: 1

Cycles: 1

Encoding:

1 1 1 0	0 1 1 i

Operation: MOV
(A) ← $((R_i))$

MOV A,#data

Bytes: 2

Cycles: 1

Encoding:

0 1 1 1	0 1 0 0	immediate data

Operation: MOV
(A) ← #data

MOV Rn,A

Bytes: 1

Cycles: 1

Encoding:

1 1 1 1	1 r r r

Operation: MOV
(R_n) ← (A)

MOV Rn,direct

Bytes: 2

Cycles: 2

Encoding:

1 0 1 0	1 r r r	direct address

Operation: MOV
(R_n) ← (direct)

MOV Rn,#data

Bytes: 2

Cycles: 1

Encoding:

0 1 1 1	1 r r r	immediate data

Operation: MOV
(R_n) ← #data

*MOV A,ACC is not a valid instruction.

MOV direct,A

Bytes: 2

Cycles: 1

Encoding:

1 1 1 1	0 1 0 1	direct address

Operation: MOV
(direct) ← (A)

MOV direct,Rn

Bytes: 2

Cycles: 2

Encoding:

1 0 0 0	1 r r r	direct address

Operation: MOV
(direct) ← (R_n)

MOV direct,direct

Bytes: 3

Cycles: 2

Encoding:

1 0 0 0	0 1 0 1	dir. addr. (src)	dir. addr. (dest)

Operation: MOV
(direct) ← (direct)

MOV direct,@Ri

Bytes: 2

Cycles: 2

Encoding:

1 0 0 0	0 1 1 i	direct address

Operation: MOV
(direct) ← $((R_i))$

MOV direct,#data

Bytes: 3

Cycles: 2

Encoding:

0 1 1 1	0 1 0 1	direct address	immediate data

Operation: MOV
(direct) ← #data

MOV @Ri,A

Bytes: 1

Cycles: 1

Encoding:

1 1 1 1	0 1 1 i

Operation: MOV
$((R_i))$ ← (A)

MOV @Ri,direct

Bytes: 2

Cycles: 2

Encoding:

1 0 1 0	0 1 1 i	direct address

Operation: MOV
$((R_i)) \leftarrow$ (direct)

MOV @Ri,#data

Bytes: 2

Cycles: 1

Encoding:

0 1 1 1	0 1 1 i	immediate data

Operation: MOV
$((R_i)) \leftarrow$ #data

MOV <dest-bit>,<src-bit>

Function: Move bit data

Description: The Boolean variable indicated by the second operand is copied into the location specified by the first operand. One of the operands must be the carry flag; the other may be any directly addressable bit. No other register or flag is affected.

Example: The carry flag is originally set. The data present at input Port 3 is 11000101B. The data previously written to output Port 1 is 35H (00110101B). The instruction sequence,

```
MOV     P1.3,C
MOV     C,P3.3
MOV     P1.2,C
```

will leave the carry cleared and change Port 1 to 39H (00111001B).

MOV C,bit

Bytes: 2

Cycles: 1

Encoding:

1 0 1 0	0 0 1 0	bit address

Operation: MOV
(C) ← (bit)

MOV bit,C

Bytes: 2

Cycles: 2

Encoding:

1 0 0 1	0 0 1 0	bit address

Operation: MOV
(bit) ← (C)

MOV DPTR,#data16

Function: Load Data Pointer with a 16-bit constant

Description: The Data Pointer is loaded with the 16-bit constant indicated. The 16-bit constant is loaded into the second and third bytes of the instruction. The second byte (DPH) is the high-order byte, while the third byte (DPL) holds the low-order byte. No flags are affected.

This is the only instruction which moves 16 bits of data at once.

Example: The instruction,

```
MOV    DPTR,#1234H
```

will load the value 1234H into the Data Pointer: DPH will hold 12H and DPL will hold 34H.

Bytes: 3

Cycles: 2

Encoding:

1 0 0 1	0 0 0 0	immed. data15-8	immed. data7-0

Operation:

MOV
(DPTR) ← (#data$_{15-0}$)
DPH □ DPL ← #data$_{15-8}$□ #data$_{7-0}$

MOVC A,@A+<base-reg>

Function: Move Code byte

Description: The MOVC instructions load the Accumulator with a code byte, or constant from program memory. The address of the byte fetched is the sum of the original unsigned eight-bit Accumulator contents and the contents of a sixteen-bit base register, which may be either the Data Pointer or the PC. In the latter case, the PC is incremented to the address of the following instruction before being added with the Accumulator; otherwise the base register is not altered. Sixteen-bit addition is performed so a carry-out from the low-order eight bits may propagate through higher-order bits. No flags are affected.

Example: A value between 0 and 3 is in the Accumulator. The following instructions will translate the value in the Accumulator to one of four values defined by the DB (define byte) directive:

```
REL_PC:    INC      A
           MOVC     A,@A+PC
           RET
           DB       66H
           DB       77H
           DB       88H
           DB       99H
```

If the subroutine is called with the Accumulator equal to 01H, it will return with 77H in the Accumulator. The INC A before the MOVC instruction is needed to “get around” the RET instruction above the table. If several bytes of code separated the MOVC from the table, the corresponding number would be added to the Accumulator instead.

MOVC A,@A+DPTR

Bytes: 1

Cycles: 2

Encoding:

1 0 0 1	0 0 1 1

Operation:

MOVC
(A) ← ((A) + (DPTR))

MOVC A,@A+PC

Bytes: 1

Cycles: 2

Encoding:

1 0 0 0	0 0 1 1

Operation: MOVC
(PC) ← (PC) + 1
(A) ← ((A) + (PC))

MOVX <dest-byte>,<src-byte> (Not implemented in the 8XC752 or 8XC752)

Function: Move External

Description: The MOVX instructions transfer data between the Accumulator and a byte of external data memory, hence the "X" appended to MOV. There are two types of instructions, differing in whether they provide an eight-bit or sixteen-bit indirect address to the external data RAM.

In the first type, the contents of R0 or R1 in the current register bank provide an eight-bit address multiplexed with data on P0. Eight bits are sufficient for external I/O expansion decoding or for a relatively small RAM array. For somewhat larger arrays, port pins can be used to output higher-order address bits. These pins would be controlled by an output instruction preceding the MOVX.

In the second type of MOVX instruction, The Data Pointer generates a sixteen-bit address. P2 outputs the high-order eight address bits (the contents of DPH) while P0 multiplexes the low-order eight bits (DPL) with data. The P2 Special Function Register retains its previous contents while the P2 output buffers are emitting the contents of DPH. This form is faster and more efficient when accessing very large data arrays (up to 64k bytes), since no additional instructions are needed to set up the output ports.

It is possible in some situations to mix the two MOVX types. A large RAM array with its high-order address lines driven by P2 can be addressed via the Data Pointer, or with code to output high-order address bits to P2 followed by a MOVX instruction using R0 or R1.

Example: An external 256 byte RAM using multiplexed address/data lines is connected to the 8051 Port 0. Port 3 provides control lines for the external RAM. Ports 1 and 2 are used for normal I/O. Registers 0 and 1 contain 12H and 34H. Location 34H of the external RAM holds the value 56H. The instruction sequence,

```
MOVX   A,@R1
MOVX   @R0,A
```

copies the value 56H into both the Accumulator and external RAM location 12H.

MOVX A,@Ri

Bytes: 1

Cycles: 2

Encoding:

1 1 1 0	0 0 1 i

Operation: MOVX
(A) ← ((R_i))

MOVX A,@DPTR

Bytes: 1

Cycles: 2

Encoding:

1 1 1 0	0 0 0 0

Operation: MOVX
(A) ← ((DPTR))

MOVX @Ri,A

Bytes: 1

Cycles: 2

Encoding:

1 1 1 1	0 0 1 i

Operation: MOVX
$((R_i)) \leftarrow (A)$

MOVX @DPTR,A

Bytes: 1

Cycles: 2

Encoding:

1 1 1 1	0 0 0 0

Operation: MOVX
$((DPTR)) \leftarrow (A)$

MUL AB

Function: Multiply

Description: MUL AB multiplies the unsigned eight-bit integers in the Accumulator and register B. The low-order byte of the sixteen-bit product is left in the Accumulator, and the high-order byte in B. If the product is greater than 255 (0FFH) the overflow flag is set; otherwise it is cleared. The carry flag is always cleared.

Example: Originally the Accumulator holds the value 80 (50H). Register B holds the value 160 (0A0H).The instruction,

MUL AB

will give the product 12,800 (3200H), so B is changed to 32H (00110010B) and the Accumulator is cleared. The overflow flag is set, carry is cleared.

Bytes: 1

Cycles: 4

Encoding:

1 0 1 0	0 1 0 0

Operation: MUL
$(A)_{7-0} \leftarrow (A) \text{ x } (B)$
$(B)_{15-8}$

NOP

Function: No Operation

Description: Execution continues at the following instruction. Other than the PC, no registers or flags are affected.

Example: It is desired to produce a low-going output pulse on bit 7 of Port 2 lasting exactly 5 cycles. A simple SETB/CLR sequence would generate a one-cycle pulse, so four additional cycles must be inserted. This may be done (assuming are enabled) with the instruction sequence,

```
CLR     P2.7
NOP
NOP
NOP
NOP
SETB    P2.7
```

Bytes: 1

Cycles: 1

Encoding:

0 0 0 0	0 0 0 0

Operation: NOP
(PC) ← (PC) + 1

ORL <dest-byte>,<src-byte>

Function: Logical-OR for byte variables

Description: ORL performs the bitwise logical-OR operation between the indicated variables, storing the results in the destination byte. No flags are affected.

The two operands allow six addressing mode combinations. When the destination is the Accumulator, the source can use register, direct, register-indirect, or immediate addressing; when the destination is a direct address, the source can be the Accumulator or immediate data.

Note: When this instruction is used to modify an output port, the value used as the original port data will be read from the output data latch, *not* the input pins.

Example: If the Accumulator holds 0C3H (11000011B) and R0 holds 55H (01010101B) then the instruction,

ORL A,R0

will leave the Accumulator holding the value 0D7H (11010111B).

When the destination is a directly addressed byte, the instruction can set combinations of bits in any RAM location or hardware register. The pattern of bits to be set is determined by a mask byte, which may be either a constant data value in the instruction or a variable computed in the Accumulator at run-time. The instruction,

ORL P1,#00110010B

will set bits 5, 4, and 1 of output Port 1.

ORL A,Rn

Bytes: 1

Cycles: 1

Encoding:

0 1 0 0	1 r r r

Operation: ORL
$(A) \leftarrow (A) \vee (R_n)$

ORL A,direct

Bytes: 2

Cycles: 1

Encoding:

0 1 0 0	0 1 0 1	direct address

Operation: ORL
$(A) \leftarrow (A) \vee (direct)$

ORL A,@Ri

Bytes: 1

Cycles: 1

Encoding:

0 1 0 0	0 1 1 i

Operation: ORL
$(A) \leftarrow (A) \vee ((R_i))$

ORL A,#data

Bytes: 2

Cycles: 1

Encoding:

0 1 0 0	0 1 0 0	immediate data

Operation: ORL
$(A) \leftarrow (A) \vee \#data$

ORL direct,A

Bytes: 2

Cycles: 1

Encoding:

0 1 0 0	0 0 1 0	direct address

Operation: ORL
$(direct) \leftarrow (direct) \vee (A)$

ORL direct,#data

Bytes: 3

Cycles: 2

Encoding:

0 1 0 0	0 0 1 1	direct address	immediate data

Operation: ORL
$(direct) \leftarrow (direct) \vee \#data$

ORL C,<src-bit>

Function: Logical-OR for bit variables

Description: Set the carry flag if the Boolean value is a logical 1; leave the carry in its current state otherwise. A slash ("/") preceding the operand in the assembly language indicates that the logical complement of the addressed bit is used as the source value, but the source bit itself is not affected. No other flags are affected.

Example: Set the carry flag if and only if P1.0 = 1, ACC.7 = 1, or OV = 0:

```
ORL   C,P1.0     ;LOAD CARRY WITH INPUT PIN P10
ORL   C,ACC.7    ;OR CARRY WITH THE ACC. BIT 7
ORL   C,/OV      ;OR CARRY WITH THE INVERSE OF OV.
```

ORL C,bit

Bytes: 2

Cycles: 2

Encoding:

0 1 1 1	0 0 1 0	bit address

Operation: ORL
$(C) \leftarrow (C) \vee (bit)$

ORL C,/bit

Bytes: 2

Cycles: 2

Encoding:

1 0 1 0	0 0 0 0	bit address

Operation: ORL
$(C) \leftarrow (C) \vee (\overline{bit})$

POP direct

Function: Pop from stack

Description: The contents of the internal RAM location addressed by the Stack Pointer is read, and the Stack Pointer is decremented by one. The value read is then transferred to the directly addressed byte indicated. No flags are affected.

Example: The Stack Pointer originally contains the value 32H, and internal RAM locations 30H through 32H contain the values 20H, 23H, and 01H, respectively. The instruction sequence,

```
POP     DPH
POP     DPL
```

will leave the Stack Pointer equal to the value 30H and the Data Pointer set to 0123H. At this point the instruction,

```
POP     SP
```

will leave the Stack Pointer set to 20H. Note that in this special case the Stack Pointer was decremented to 2FH before being loaded with the value popped (20H).

Bytes: 2

Cycles: 2

Encoding:

1 1 0 1	0 0 0 0		direct address

Operation: POP
(direct) ← ((SP))
(SP) ← (SP) – 1

PUSH direct

Function: Push onto stack

Description: The Stack Pointer is incremented by one. The contents of the indicated variable is then copied into the internal RAM location addressed by the Stack Pointer. Otherwise no flags are affected.

Example: On entering an interrupt routine the Stack Pointer contains 09H. The Data Pointer holds the value 0123H. The instruction sequence,

```
PUSH    DPL
PUSH    DPH
```

will leave the Stack Pointer set to 0BH and store 23H and 01H in internal RAM locations 0AH and 0BH, respectively.

Bytes: 2

Cycles: 2

Encoding:

1 1 0 0	0 0 0 0		direct address

Operation: PUSH
(SP) ← (SP) + 1
((SP)) ← (direct)

RET

Function: Return from subroutine

Description: RET pops the high- and low-order bytes of the PC successively from the stack, decrementing the Stack Pointer by two. Program execution continues at the resulting address, generally the instruction immediately following an ACALL or LCALL. No flags are affected.

Example: The Stack Pointer originally contains the value 0BH. Internal RAM locations 0AH and 0BH contain the values 23H and 01H, respectively. The instruction,

RET

will leave the Stack Pointer equal to the value 09H. Program execution will continue at location 0123H.

Bytes: 1

Cycles: 2

Encoding:

0 0 1 0	0 0 1 0

Operation: RET
$(PC_{15-8}) \leftarrow ((SP))$
$(SP) \leftarrow (SP) - 1$
$(PC_{7-0}) \leftarrow ((SP))$
$(SP) \leftarrow (SP) - 1$

RETI

Function: Return from interrupt

Description: RETI pops the high- and low-order bytes of the PC successively from the stack, and restores the interrupt logic to accept additional interrupts at the same priority level as the one just processed. The Stack Pointer is left decremented by two. No other registers are affected; the PSW is not automatically restored to its pre-interrupt status. Program execution continues at the resulting address, which is generally the instruction immediately after the point at which the interrupt request was detected. If a lower- or same-level interrupt has been pending when the RETI instruction is executed, that one instruction will be executed before the pending interrupt is processed.

Example: The Stack Pointer originally contains the value 0BH. An interrupt was detected during the instruction ending at location 0122H. Internal RAM locations 0AH and 0BH contain the values 23H and 01H, respectively. The instruction,

RETI

will leave the Stack Pointer equal to 09H and return program execution to location 0123H.

Bytes: 1

Cycles: 2

Encoding:

0 0 1 1	0 0 1 0

Operation: RETI
$(PC_{15-8}) \leftarrow ((SP))$
$(SP) \leftarrow (SP) - 1$
$(PC_{7-0}) \leftarrow ((SP))$
$(SP) \leftarrow (SP) - 1$

RL A

Function: Rotate Accumulator Left

Description: The eight bits in the Accumulator are rotated one bit to the left. Bit 7 is rotated into the bit 0 position. No flags are affected.

Example: The Accumulator holds the value 0C5H (11000101B). The instruction,

RL A

leaves the Accumulator holding the value 8BH (10001011B) with the carry unaffected.

Bytes: 1

Cycles: 1

Encoding:

0 0 1 0	0 0 1 1

Operation: RL
$(A_{n+1}) \leftarrow (A_n)$, n = 0 – 6
$(A0) \leftarrow (A7)$

RLC A

Function: Rotate Accumulator Left through the Carry flag

Description: The eight bits in the Accumulator and the carry flag are together rotated one bit to the left. Bit 7 moves into the carry flag; the original state of the carry flag moves into the bit 0 position. No other flags are affected.

Example: The Accumulator holds the value 0C5H (11000101B), and the carry is zero. The instruction,

RLC A

leaves the Accumulator holding the value 8AH (10001010B) with the carry set.

Bytes: 1

Cycles: 1

Encoding:

0 0 1 1	0 0 1 1

Operation: RLC
$(A_{n+1}) \leftarrow (A_n)$, n = 0 – 6
$(A0) \leftarrow (C)$
$(C) \leftarrow (A7)$

RR A

Function: Rotate Accumulator Right

Description: The eight bits in the Accumulator are rotated one bit to the right. Bit 0 is rotated into the bit 7 position. No flags are affected.

Example: The Accumulator holds the value 0C5H (11000101B). The instruction,

RR A

leaves the Accumulator holding the value 0E2H (11100010B) with the carry unaffected.

Bytes: 1

Cycles: 1

Encoding:

0 0 0 0	0 0 1 1

Operation: RR
$(A_n) \leftarrow (A_{n+1})$, n = 0 – 6
$(A7) \leftarrow (A0)$

RRC A

Function: Rotate Accumulator Right through the Carry flag

Description: The eight bits in the Accumulator and the carry flag are together rotated one bit to the right. Bit 0 moves into the carry flag; the original state of the carry flag moves into the bit 7 position. No other flags are affected.

Example: The Accumulator holds the value 0C5H (11000101B), and the carry is zero. The instruction,

RRC A

leaves the Accumulator holding the value 62 (01100010B) with the carry set.

Bytes: 1

Cycles: 1

Encoding:

0 0 0 1	0 0 1 1

Operation: RRC
$(A_n) \leftarrow (A_{n+1})$, n = 0 – 6
$(A7) \leftarrow (C)$
$(C) \leftarrow (A0)$

SETB <bit>

Function: Set Bit

Description: SETB sets the indicated bit to one. SETB can operate on the carry flag or any directly addressable bit. No other flags are affected.

Example: The carry flag is cleared. Output Port 1 has been written with the value 34H (00110100B). The instructions,

```
SETB   C
SETB   P1.0
```

will leave the carry flag set to 1 and change the data output on Port 1 to 35H (00110101B).

SETB C

Bytes: 1

Cycles: 1

Encoding:

1 1 0 1	0 0 1 1

Operation: SETB
(C) ← 1

SETB bit

Bytes: 2

Cycles: 1

Encoding:

1 1 0 1	0 0 1 0	bit address

Operation: SETB
(bit) ← 1

SJMP rel

Function: Short Jump

Description: Program control branches unconditionally to the address indicated. The branch destination is computed by adding the signed displacement in the second instruction byte to the PC, after incrementing the PC twice. Therefore, the range of destinations allowed is from 128 bytes preceding this instruction to 127 bytes following it.

Example: The label "RELADR" is assigned to an instruction at program memory location 0123H. The instruction,

```
SJMP   RELADR
```

will assemble into location 0100H. After the instruction is executed, the PC will contain the value 0123H.

(*Note:* Under the above conditions the instruction following SJMP will be at 102H. Therefore, the displacement byte of the instruction will be the relative offset (0123H-0102H) = 21H. Put another way, an SJMP with a displacement of 0FEH would be a one-instruction infinite loop.)

Bytes: 2

Cycles: 2

Encoding:

1 0 0 0	0 0 0 0	rel. address

Operation: SJMP
(PC) ← (PC) + 2
(PC) ← (PC) + rel

SUBB A, <src-byte>

Function: Subtract with borrow

Description: SUBB subtracts the indicated variable and the carry flag together from the Accumulator, leaving the result in the Accumulator. SUBB sets the carry (borrow) flag if a borrow is needed for bit 7, and clears C otherwise. (If C was set *before* executing a SUBB instruction, this indicates that a borrow was needed for the previous step in a multiple precision subtraction, so the carry is subtracted from the Accumulator along with the source operand.) AC is set if a borrow is needed for bit 3, and cleared otherwise. OV is set if a borrow is needed into bit 6, but not into bit 7, or into bit 7, but not bit 6.

When subtracting signed integers OV indicates a negative number produced when a negative value is subtracted from a positive value, or a positive result when a positive number is subtracted from a negative number.

The source operand allows four addressing modes: register, direct, register-indirect, or immediate.

Example: The Accumulator holds 0C9H (11001001B), register 2 holds 54H (01010100B), and the carry flag is set. The instruction,

SUBB A,R2

will leave the value 74H (01110100B) in the Accumulator, with the carry flag and AC cleared but OV set.

Notice that 0C9H minus 54H is 75H The difference between this and the above result is due to the carry (borrow) flag being set before the operation. If the state of the carry is not known before starting a single or multiple-precision subtraction, it should be explicitly cleared by a CLR C instruction

SUBB A,Rn

Bytes: 1

Cycles: 1

Encoding:

1 0 0 1	1 r r r

Operation: SUBB
$(A) \leftarrow (A) - (C) - (R_n)$

SUBB A,direct

Bytes: 2

Cycles: 1

Encoding:

1 0 0 1	0 1 0 1	direct address

Operation: SUBB
$(A) \leftarrow (A) - (C) - (\text{direct})$

SUBB A,@Ri

Bytes: 1

Cycles: 1

Encoding:

1 0 0 1	0 1 1 i

Operation: SUBB
$(A) \leftarrow (A) - (C) - (R_i)$

SUBB A,#data

Bytes: 2

Cycles: 1

Encoding:

1 0 0 1	0 1 0 0	immediate data

Operation: SUBB
$(A) \leftarrow (A) - (C) - (\#\text{data})$

SWAP A

Function: Swap nibbles within the Accumulator

Description: SWAP A interchanges the low- and high-order nibbles (four-bit fields) of the Accumulator (bits 3-0 and bits 7-4). The operation can also be thought of as a four-bit rotate instruction. No flags are affected.

Example: The Accumulator holds the value 0C5H (11000101B). The instruction,

SWAP A

leaves the Accumulator holding the value 5CH (01011100B).

Bytes: 1

Cycles: 1

Encoding:

1 1 0 0	0 1 0 0

Operation: SWAP
$(A_{3\text{-}0}) \rightleftarrows (A_{7\text{-}4})$

XCH A,<byte>

Function: Exchange Accumulator with byte variable

Description: XCH loads the Accumulator with the contents of the indicated variable, at the same time writing the original Accumulator contents to the indicated variable. The source/destination operand can use register, direct, or register-indirect addressing.

Example: R0 contains the address 20H. The Accumulator holds the value 3FH (00111111B). Internal RAM location 20H holds the value 75H (01110101B). The instruction,

XCH A,@R0

will leave the RAM location 20H holding the values 3FH (00111111B) and 75H (01110101B) in the Accumulator.

XCH A,Rn

Bytes: 1

Cycles: 1

Encoding:

1 1 0 0	1 r r r

Operation: XCH
$(A) \rightleftarrows (R_n)$

XCH A,direct

Bytes: 2

Cycles: 1

Encoding:

1 1 0 0	0 1 0 1	direct address

Operation: XCH
$(A) \rightleftarrows (\text{direct})$

XCH A,@Ri

Bytes: 1

Cycles: 1

Encoding:

1 1 0 0	0 1 1 i

Operation: XCH
$(A) \rightleftarrows ((R_i))$

XCHD A,@Ri

Function: Exchange Digit

Description: XCHD exchanges the low-order nibble of the Accumulator (bits 3-0), generally representing a hexadecimal or BCD digit, with that of the internal RAM location indirectly addressed by the specified register. The high-order nibbles (bits 7-4) of each register are not affected. No flags are affected.

Example: R0 contains the address 20H. The Accumulator holds the value 36H (00110110B). Internal RAM location 20H holds the value 75H (01110101B). The instruction,

XCHD A,@R0

will leave RAM location 20H holding the value 76H (01110110B) and 35H (00110101B) in the Accumulator.

Bytes: 1

Cycles: 1

Encoding:

1 1 0 1	0 1 1 i

Operation: XCHD
$(A_{3-0}) \rightleftarrows ((Ri_{3-0}))$

XRL <dest-byte>,<src-byte>

Function: Logical Exclusive-OR for byte variables

Description: XRL performs the bitwise logical Exclusive-OR operation between the indicated variables, storing the results in the destination. No flags are affected.

The two operands allow six addressing mode combinations. When the destination is the Accumulator, the source can use register, direct, register-indirect, or immediate addressing; when the destination is a direct address, the source can be the Accumulator or immediate data.

(*Note:* When this instruction is used to modify an output port, the value used as the original port data will be read from the output data latch, *not* the input pins.)

Example: If the Accumulator holds 0C3H (11000011B) and register 0 holds 0AAH (10101010B) then the instruction,

XRL A,R0

will leave the Accumulator holding the value 69H (01101001B).

When the destination is a directly addressed byte, this instruction can complement combinations of bits in any RAM location or hardware register. The pattern of bits to be complemented is then determined by a mask byte, either a constant contained in the instruction or a variable computed in the Accumulator at run-time. The instruction,

XRL P1,#00110001B

will complement bits 5, 4, and 0 of output Port 1.

XRL A,Rn

Bytes: 1

Cycles: 1

Encoding:

0 1 1 0	1 r r r

Operation: XRL
$(A) \leftarrow (A) \veebar (R_n)$

XRL A,direct

Bytes: 2

Cycles: 1

Encoding:

0 1 1 0	0 1 0 1	direct address

Operation: XRL
$(A) \leftarrow (A) \veebar (\text{direct})$

XRL A,@Ri

Bytes: 1

Cycles: 1

Encoding:

0 1 1 0	0 1 1 i

Operation: XRL
$(A) \leftarrow (A) \veebar ((R_i))$

XRL A,#data

Bytes: 2

Cycles: 1

Encoding:

0 1 1 0	0 1 0 0	immediate data

Operation: XRL
$(A) \leftarrow (A) \veebar \#\text{data}$

XRL direct,A

Bytes: 2

Cycles: 1

Encoding:

0 1 1 0	0 0 1 0	direct address

Operation: XRL
$(\text{direct}) \leftarrow (\text{direct}) \veebar (A)$

XRL direct,#data

Bytes: 3

Cycles: 2

Encoding:

0 1 1 0	0 0 1 1	direct address	immediate data

Operation: XRL
$(\text{direct}) \leftarrow (\text{direct}) \veebar \#\text{data}$

EPROM PRODUCTS

Programming the 87C51

The setup for programming the microcontroller is shown in Figure 1. Note that the part is running with a 4 to 6 MHz oscillator. The clock must be running because the device is executing internal address and program data transfers during the programming.

To program the 87C51, the address of the EPROM location to be programmed is applied to ports 1 and 2 as shown in Figure 1. The code byte to be programmed into this location is applied to port 0. RST, $\overline{\text{PSEN}}$, and the pins of ports 2 and 3 specified in Table 1 are held at the "Program Code Data" levels specified in the table. The ALE/$\overline{\text{PROG}}$ is then pulsed low 25 times to program the addressed location.

Encryption Table

The encryption table is a feature of the 87C51, and its derivatives, that protects the code from being easily read by anyone other than the programmer. The encryption table is 16 to 64 bytes of code, depending on the microcontroller, that are exclusive NORed with the program code data as it is read out. The first byte is XNORed with the first location read, the second with the second read, etc. through the sixteenth byte read. The seventeenth byte is XNORed with the first byte of the encryption table, the eighteenth with the second, etc. and on in sixteen-byte groups.

After the Encryption table has been programmed the user has to know its contents in order to correctly decode the program code data. The encryption table itself cannot be read out.

The encryption table is programmed in the same manner as the program memory, but using the "Pgm Encryption Table" levels specified in Table 1. After the encryption table is programmed, verification cycles will produce only encrypted information.

Security Bit

There are two security bits on the 87C51 that, when set, prevent the program data memory from being read out or programmed further. To program the security bits, repeat the programming sequence using the "Pgm Security Bit" levels specified in Table 1.

After the first security bit is programmed, further programming of the code memory or the encryption table is disabled. The other security bit can of course still be programmed. With only security bit one programmed, the memory can still be read out for program verification. After the second security bit is programmed, it is no longer possible to read out (verify) the program memory.

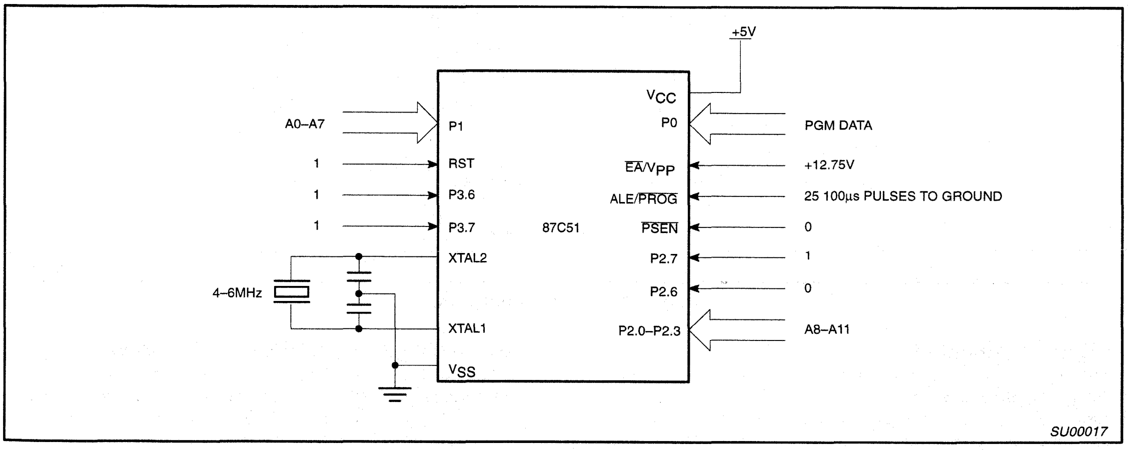


Figure 1. Programming Configuration

Table 1. EPROM Programming Modes

MODE	RST	$\overline{\text{PSEN}}$	ALE/$\overline{\text{PROG}}$	$\overline{\text{EA}}/V_{PP}$	P2.7	P2.6	P3.7	P3.6
Read signature	1	0	1	1	0	0	0	0
Program code data	1	0	0*	V_{PP}	1	0	1	1
Verify code data	1	0	1	1	0	0	1	1
Pgm encryption table	1	0	0*	V_{PP}	1	0	1	0
Pgm security bit 1	1	0	0*	V_{PP}	1	1	1	1
Pgm security bit 2	1	0	0*	V_{PP}	1	1	0	0

NOTES:
1. “0” = valid low for that pin, “1” = valid high for that pin.
2. V_{PP} = 12.75 ±0.25V.
3. V_{CC} = 5V ±10% during programming and verification.

* ALE/$\overline{\text{PROG}}$ receives 25 programming pulses while V_{PP} is held at 12.75V. Each programming pulse is low for 100ms (±10μs) and high for a minimum of 10μs.

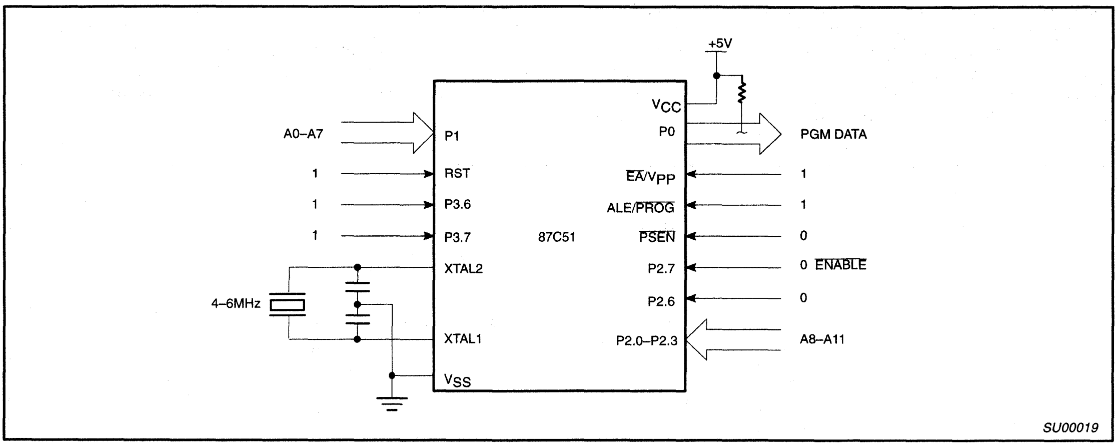


Figure 2. Program Verification

Program Verification

If security bit 2 has not been programmed the on-chip program memory can be read out for program verification. To verify the contents of the program memory, the address of the location to be read is applied to ports 1 and 2 as shown in Figure 2. The other pins are held at the “Verify Code Data” levels indicated in Table 1. The contents of the addressed location will appear on port 0. For this operation external pull-ups are required on port 0 as shown in Figure 2. Note that if the encryption table has been programmed the data presented at port 0 will be the exclusive NOR of the program byte with a byte from the encryption table.

Signature Bytes

The 87C51 contains two signature bytes that can be read and used by an EPROM programming system to identify the device. The signature bytes identify the device as an 87C51 manufactured by Philips.

The signature bytes are read by the same procedure as a normal verification of locations 030H and 031H, except that P3.6 and P3.7 need to be pulled to a logic low.

Programming the 87C750, 87C751 and 87C752

The 87C750, 87C751 and 87C752 are programmed using a Quick-pulse programming algorithm that is similar to that used for the 87C51. It differs from the 87C51 in that a serial data stream is used to place the 87C751 in the programming mode.

Figure 3 shows a block diagram of the programming configuration for the 87C751. Port pin P0.2 is used for the programming voltage supply input (V_{PP} signal). Port pin P0.1 is used for the program (PGM) signal.

Port 3 accepts the address input for the EPROM location to be programmed. Both the high and low components of the eleven-bit address are presented to the part through port 3. Multiplexing of the address components is performed using ASEL (P0.0).

Port 1 is used as a bidirectional data bus during programming and verify operations. During the programming mode, it accepts the byte to be programmed. In the verify mode, it returns the contents of the specified address location.

The X1 pin is the oscillator input and receives the master system clock. This clock should be between 1.2 and 6MHz.

The RESET pin is used to accept the serial data stream that places the 87C751 into various programming modes. This pattern consists of a 10-bit code with the LSB send first. Each bit is synchronized to the clock input X1.

To program the 87C751 the part must be put into the programming mode by presenting the proper serial code (see Table 2) to the RESET pin. To do this RESET should be held high for at least two machine cycles. Port pins P0.1 and P0.2 will be at VOH as a result of this, but they must be driven high prior to sending the serial data stream on the RESET pin. The serial data bits can now be transmitted over the RESET pin placing the 87C751 into one of the programming modes. Following the transmission of the last data bit, the reset pin should be held low.

Next the address information for the location to be programmed is placed on Port 3 and ASEL is used to perform the address multiplexing. ASEL should be driven high and then Port 3 driven with the high-order address bits. ASEL is then driven low, latching the high-order bits internally. Port 3 can now be driven with the low 8 bits of the address, completing the addressing of the location to be programmed.

A high-voltage V_{PP} level is now applied to the V_{PP} input. This sets Port 1 as an input port. The data to be programmed to the EPROM array should be placed on Port 1. A series of 25 programming pulses is now applied to the PGM pin (P0.1) to program the addressed EPROM location.

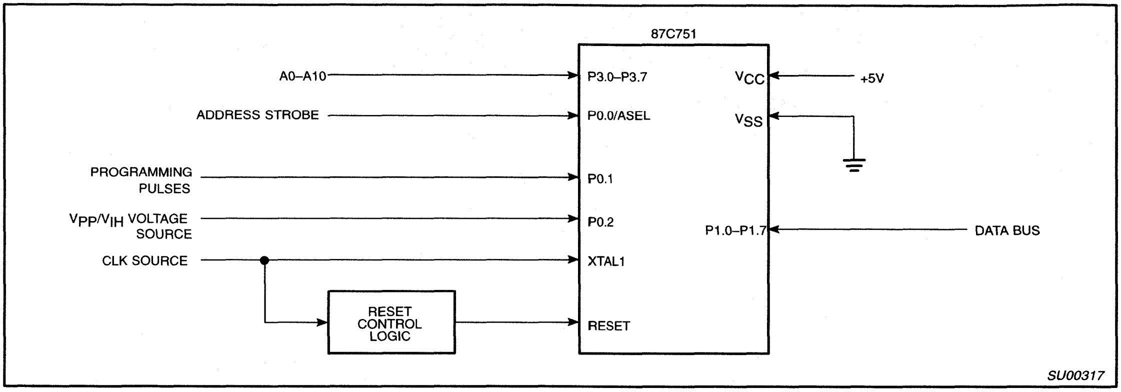


Figure 3. Programming Configuration

Table 2. 87C750, 87C751, and 87C752 Serial Codes

OPERATION	SERIAL CODE	P0.1 (PGM/)	P0.2 (V_{PP})
Program user EPROM	296H	–*	V_{PP}
Verify user EPROM	296H	V_{IH}	V_{IH}
Program key EPROM	292H	–*	V_{PP}
Verify key EPROM	292H	V_{IH}	V_{IH}
Program security bit 1	29AH	–*	V_{PP}
Program security bit 2	298H	–*	V_{PP}
Verify security bits	29AH	V_{IH}	V_{IH}
Read signature bytes	294H	V_{IH}	V_{IH}

NOTE:

* Pulsed from V_{IH} to V_{IL} and returned to V_{IH}.

Program Verification

The EPROM array can be verified by placing the part in the programming mode as described above and forcing the V_{PP} pin to the V_{OH} level. Four machine cycles after addressing a location the contents of the addressed location will appear on Port 1.

87C750, 87C751 and 87C752 Signature Bytes

The signature bytes for the 87C750, 87C751 and 87C752 are read differently and are in different locations than those on the 87C51. Due to its reduced pin count, the part has to be put into "Signature Byte Read Mode" by placing a 10-bit serial data stream on the Reset pin. The proper code and the conditions of P0.1 and P0.2, for this mode, are shown in Table 2.

Once the part has been placed into the Signature Byte Read Mode, the signature bytes can be read by the same procedure as a normal verification of locations 01EH and 01FH. The values are:

01EH	=	15H indicates the part is made by Philips
01FH	=	91H - 87C751
01FH	=	95H - 87C752

Programming Features

The 87C751 has all of the special programming features incorporated within its EPROM array that the 87C51 has. It has an encryption key table and two security bits. These function exactly as they do in the 87C51. They are programmed or verified by sending the proper code over the RESET pin (see Table 2) and then following the 87C751 programming procedure as described previously.

Section 3
80C51 Family Derivatives

80C51-Based 8-Bit Microcontrollers

CONTENTS

8XC51/80C31	8-bit CMOS (low voltage, low power, and high speed) microcontroller families	143
8XC52/54/58/80C32 8XC51FA/FB/FC/80C51FA 8XC51RA+/RB+/ RC+/RD+/80C51RA+	8-bit CMOS (low voltage, low power and high speed) microcontroller families	146
89C52/54/58 89C51RA+/RB+/RC+/RD+	8-bit CMOS microcontroller families with FLASH program memory	151
83C145; 83C845; 83C055; 87C055	Microcontrollers for TV and video (MTV)	154
P83Cx80; P87C380	Microcontrollers for monitors with DDC interface, auto-sync detection and sync proc.	156
P80CL31; P80CL51	Low voltage 8-bit microcontrollers with UART	159
P80CL410; P83CL410	Low voltage 8-bit microcontrollers with I^{2}C–bus	162
P83C434; P83C834	8-bit microcontrollers with LCD-driver	164
80C451/83C451/87C451	CMOS single-chip 8-bit microcontrollers	166
83C453/87C453	CMOS single-chip 8-bit microcontrollers	169
87C524	CMOS single-chip 8-bit microcontroller	171
87C528	CMOS single-chip 8-bit microcontroller	173
P8XC524;P80C528; P83C528	CMOS single-chip 8–bit	175
80C550/83C550/87C550	CMOS single-chip 8-bit microcontroller with A/D and watchdog timer	179
80C552/83C552	Single-chip 8-bit microcontroller with 10-bit A/D, capture/compare timer, high-speed outputs, PWM	181
87C552	Single-chip 8-bit microcontroller with 10-bit A/D, capture/compare timer, high-speed outputs, PWM	184
P83CE558/P80CE558/ P89CE558	Single-chip 8-bit microcontroller	186
P83CE559/P80CE559	Single-chip 8-bit microcontroller	188
P8xCE560	8-bit microcontroller	190
P83C562; P80C562	8-bit microcontroller	193
80C575/83C575/87C575	CMOS single-chip 8-bit microcontrollers	195
83C576/87C576	CMOS single-chip 8-bit microcontrollers	197
P80CL580; P83CL580	Low voltage 8–bit microcontrollers with UART, I^{2}C–bus and ADC	199
80C652/83C652	CMOS single-chip 8-bit microcontrollers	201
87C652	CMOS single-chip 8-bit microcontroller	205
83C654	CMOS single-chip 8-bit microcontroller	207
87C654	CMOS single-chip 8-bit microcontroller	211
P89C738; P89C739	8–bit microcontrollers	213
83C748/87C748	CMOS single-chip 8-bit microcontrollers	215
83C749/87C749	CMOS single-chip 8-bit microcontrollers	217
TPM749	Microcontroller with TrackPoint™ microcode from IBM . . .	219
83C750/87C750	CMOS single-chip 8-bit microcontrollers	221
83C751/87C751	CMOS single-chip 8-bit microcontrollers	223
83C752/87C752	CMOS single-chip 8-bit microcontroller with A/D, PWM . .	225

Section 3
80C51 Family Derivatives

(continued)

83C754/87C754 CMOS single-chip 8-bit microcontrollers 227

TPM754 Microcontroller with TrackPoint™ microcode from IBM . . . 229

P83CL781; P83CL782 Low voltage 8–bit microcontrollers with UART and I^{2}C–bus 231

80C851/83C851 CMOS single-chip 8-bit microcontroller with on-chip EEPROM 234

TELX family Low voltage 8–bit microcontrollers 236

P83CL883; P87CL883; P83CL884; P87CL884 TELX microcontrollers for CT0 handset/basestation applications 238

P83CL886; P87CL886 TELX microcontrollers for CT0 handset/basestation applications 241

P83CL887; P87CL887 TELX microcontrollers for CT0 handset/basestation applications 243

P87CL881 Low voltage 8–bit microcontroller 245

8-bit CMOS (low voltage, low power, and high speed) microcontroller families

8XC51/80C31

DESCRIPTION

The Philips 8XC51/31 is a high-performance static 80C51 design fabricated with Philips high-density CMOS technology with operation from 2.7V to 5.5V.

The 8XC51/31 contains a 4k × 8 ROM, a 128 × 8 RAM, 32 I/O lines, three 16-bit counter/timers, a six-source, four-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 clock circuits.

In addition, the device is a low power static design which offers a wide range of operating frequencies down to zero. Two software selectable modes of power reduction—idle mode and power-down mode are available. 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. Since the design is static, the clock can be stopped without loss of user data and then the execution resumed from the point the clock was stopped.

SELECTION TABLE

For applications requiring more ROM and RAM, see 8XC52/54/58/80C32, 8XC51FA/FB/FC/80C51FA, and 8XC51RA+/RB+/RC+/80C51RA+ data sheet.

ROM/EPROM Memory Size (X by 8)	RAM Size (X by 8)	Programmable Timer Counter (PCA)	Hardware Watch Dog Timer
80C31/8XC51			
0K/4K	128	No	No
80C32/8XC52/54/58			
0K/8K/16K/32K	256	No	No
80C51FA/8XC51FA/FB/FC			
0K/8K/16K/32K	256	Yes	No
80C51RA+/8XC51RA+/RB+/RC+			
0K/8K/16K/32K	512	Yes	Yes
8XC51RD+			
64K	1024	Yes	Yes

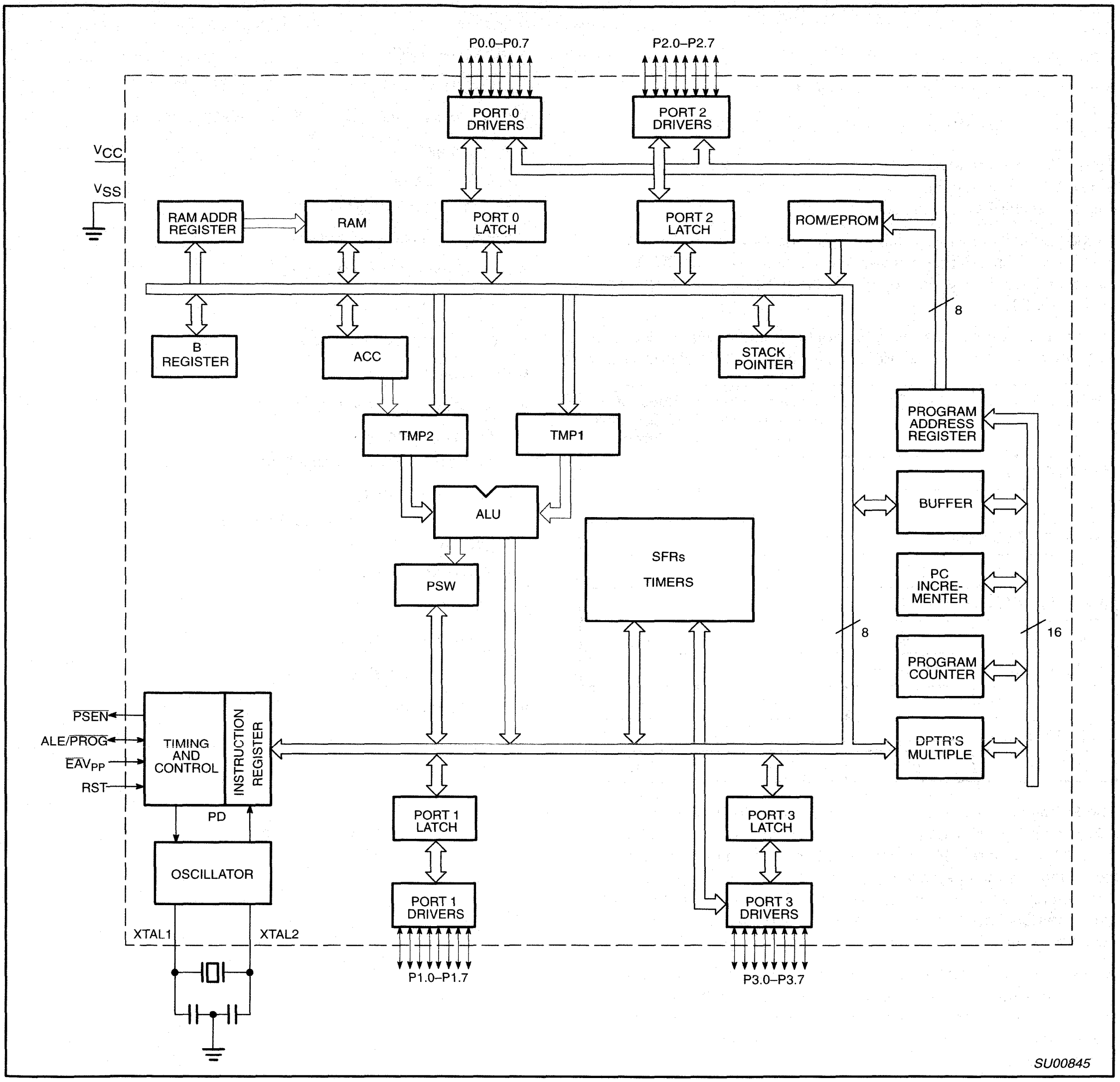
FEATURES

- 8051 Central Processing Unit
  - 4k × 8 ROM (80C51)
  - 128 × 8 RAM
  - Three 16-bit counter/timers
  - Full duplex serial channel
  - Boolean processor
  - Full static operation
  - Low voltage (2.7V to 5.5V@ 16MHz) operation
- Memory addressing capability
  - 64k ROM and 64k RAM
- Power control modes:
  - Clock can be stopped and resumed
  - Idle mode
  - Power-down mode
- CMOS and TTL compatible
- Three speed ranges at V_{CC} = 5V
  - 0 to 16MHz
  - 0 to 33MHz
- Three package styles
- Extended temperature ranges
- Dual Data Pointers
- Second DPTR register
- Security bits:
  - ROM (2 bits)
  - OTP/EPROM (3 bits)
- Encryption array–64 bytes
- 4 level priority interrupt
- 6 interrupt sources
- Four 8–bit I/O ports
- Full–duplex enhanced UART
  - Framing error detection
  - Automatic address recognition
- Programmable clock out
- Asynchronous port reset
- Low EMI (inhibit ALE)
- Wake–up from Power Down by an external interrupt f(8XC51)

80C51 AND 80C31 ORDERING INFORMATION

	MEMORY SIZE 4K × 8	ROMless	TEMPERATURE RANGE °C AND PACKAGE	VOLTAGE RANGE	FREQ. (MHz)	DWG. #
ROM	P80C51SBPN	P80C31SBPN	0 to +70, Plastic Dual In-line Package	2.7V to 5.5V	0 to 16	SOT129-1
OTP	P87C51SBPN					
ROM	P80C51SBAA	P80C31SBAA	0 to +70, Plastic Leaded Chip Carrier	2.7V to 5.5V	0 to 16	SOT187-2
OTP	P87C51SBAA					
ROM	P80C51SBBB	P80C31SBBB	0 to +70, Plastic Quad Flat Pack	2.7V to 5.5V	0 to 16	SOT307-2
OTP	P87C51SBBB					
ROM	P80C51SFPN	P80C31SFPN	–40 to +85, Plastic Dual In-line Package	2.7V to 5.5V	0 to 16	SOT129-1
OTP	P87C51SFPN					
ROM	P80C51SFAA	P80C31SFAA	–40 to +85, Plastic Leaded Chip Carrier	2.7V to 5.5V	0 to 16	SOT187-2
OTP	P87C51SFAA					
ROM	P80C51SFBB	P80C31SFBB	–40 to +85, Plastic Quad Flat Pack	2.7V to 5.5V	0 to 16	SOT307-2
OTP	P87C51SFBB					
ROM	P80C51UBAA	P80C31UBAA	0 to +70, Plastic Leaded Chip Carrier	5V	0 to 33	SOT187-2
OTP	P87C51UBAA					
ROM	P80C51UBPN	P80C31UBPN	0 to +70, Plastic Dual In-line Package	5V	0 to 33	SOT129-1
OTP	P87C51UBPN					
ROM	P80C51UBBB	P80C31UBBB	0 to +70, Plastic Quad Flat Pack	5V	0 to 33	SOT307-2
OTP	P87C51UBBB					
ROM	P80C51UFAA	P80C31UFAA	–40 to +85, Plastic Leaded Chip Carrier	5V	0 to 33	SOT187-2
OTP	P87C51UFAA					
ROM	P80C51UFPN	P80C31UFPN	–40 to +85, Plastic Dual In-line Package	5V	0 to 33	SOT129-1
OTP	P87C51UFPN					
ROM	P80C51UFBB	P80C31UFBB	–40 to +85, Plastic Quad Flat Pack	5V	0 to 33	SOT307-2
OTP	P87C51UFBB					

BLOCK DIAGRAM



8-bit CMOS (low voltage, low power and high speed) microcontroller families

8XC52/54/58/80C32
8XC51FA/FB/FC/80C51FA
8XC51RA+/RB+/RC+/RD+/80C51RA+

DESCRIPTION

Three different Single-Chip 8-Bit Microcontroller families are presented in this datasheet:

- 80C32/8XC52/8XC54/8XC58
- 80C51FA/8XC51FA/8XC51FB/8XC51FC
- 80C51RA+/8XC51RA+/8XC51RB+/8XC51RC+/8XC51RD+

For applications requiring 4K ROM/EPROM, see the 8XC51/80C31 8-bit CMOS (low voltage, low power, and high speed) microcontroller families datasheet.

All the families are Single-Chip 8-Bit Microcontrollers manufactured in advanced CMOS process and are derivatives of the 80C51 microcontroller family. All the devices have the same instruction set as the 80C51.

These devices provide architectural enhancements that make them applicable in a variety of applications for general control systems.

ROM/EPROM Memory Size (X by 8)	RAM Size (X by 8)	Programmable Timer Counter (PCA)	Hardware Watch Dog Timer
80C31/8XC51			
0K/4K	128	No	No
80C32/8XC52/54/58			
0K/8K/16K/32K	256	No	No
80C51FA/8XC51FA/FB/FC			
0K/8K/16K/32K	256	Yes	No
80C51RA+/8XC51RA+/RB+/RC+			
0K/8K/16K/32K	512	Yes	Yes
8XC51RD+			
64K	1024	Yes	Yes

The ROMless devices, 80C32, 80C51FA, and 80C51RA+ can address up to 64K of external memory. All the devices have four 8-bit I/O ports, three 16-bit timer/event counters, a multi-source, four-priority-level, nested interrupt structure, an enhanced UART and on-chip oscillator and timing circuits. For systems that require extra memory capability up to 64k bytes, each can be expanded using standard TTL-compatible memories and logic.

Its added features make it an even more powerful microcontroller for applications that require pulse width modulation, high-speed I/O and up/down counting capabilities such as motor control. It also has a more versatile serial channel that facilitates multiprocessor communications.

FEATURES

- 80C51 Central Processing Unit
- Speed up to 33MHz
- Full static operation
- Operating voltage range: 2.7V to 5.5V @ 16MHz
- Security bits:
  - ROM – 2 bits
  - OTP–EPROM – 3 bits
- Encryption array – 64 bytes
- RAM expandable to 64K bytes
- 4 level priority interrupt
- 6 or7 interrupt sources, depending on device
- Four 8-bit I/O ports
- Full-duplex enhanced UART
  - Framing error detection
  - Automatic address recognition
- Power control modes
  - Clock can be stopped and resumed
  - Idle mode
  - Power down mode
- Programmable clock out
- Second DPTR register
- Asynchronous port reset
- Low EMI (inhibit ALE)

8XC52/54/58 AND 80C32 ORDERING INFORMATION

	MEMORY SIZE 8K × 8	MEMORY SIZE 16K × 8	MEMORY SIZE 32K × 8	ROMless	TEMPERATURE RANGE °C AND PACKAGE	VOLTAGE RANGE	FREQ. (MHz)	DWG. #
ROM	P80C52SBPN	P80C54SBPN	P80C58SBPN	P80C32SBPN	0 to +70, Plastic Dual In-line Package	2.7V to 5.5V	0 to 16	SOT129-1
OTP	P87C52SBPN	P87C54SBPN	P87C58SBPN					
ROM	P80C52SBAA	P80C54SBAA	P80C58SBAA	P80C32SBAA	0 to +70, Plastic Leaded Chip Carrier	2.7V to 5.5V	0 to 16	SOT187-2
OTP	P87C52SBAA	P87C54SBAA	P87C58SBAA					
ROM	P80C52SBBB	P80C54SBBB	P80C58SBBB	P80C32SBBB	0 to +70, Plastic Quad Flat Pack	2.7V to 5.5V	0 to 16	SOT307-2
OTP	P87C52SBBB	P87C54SBBB	P87C58SBBB					
ROM	P80C52SFPN	P80C54SFPN	P80C58SFPN	P80C32SFPN	–40 to +85, Plastic Dual In-line Package	2.7V to 5.5V	0 to 16	SOT129-1
OTP	P87C52SFPN	P87C54SFPN	P87C58SFPN					
ROM	P80C52SFAA	P80C54SFAA	P80C58SFAA	P80C32SFAA	–40 to +85, Plastic Leaded Chip Carrier	2.7V to 5.5V	0 to 16	SOT187-2
OTP	P87C52SFAA	P87C54SFAA	P87C58SFAA					
ROM	P80C52SFBB	P80C54SFBB	P80C58SFBB	P80C32SFBB	–40 to +85, Plastic Quad Flat Pack	2.7V to 5.5V	0 to 16	SOT307-2
OTP	P87C52SFBB	P87C54SFBB	P87C58SFBB					
ROM	P80C52UBAA	P80C54UBAA	P80C58UBAA	P80C32UBAA	0 to +70, Plastic Leaded Chip Carrier	5V	0 to 33	SOT187-2
OTP	P87C52UBAA	P87C54UBAA	P87C58UBAA					
ROM	P80C52UBPN	P80C54UBPN	P80C58UBPN	P80C32UBPN	0 to +70, Plastic Dual In-line Package	5V	0 to 33	SOT129-1
OTP	P87C52UBPN	P87C54UBPN	P87C58UBPN					
ROM	P80C52UBBB	P80C54UBBB	P80C58UBBB	P80C32UBBB	0 to +70, Plastic Quad Flat Pack	5V	0 to 33	SOT307-2
OTP	P87C52UBBB	P87C54UBBB	P87C58UBBB					
ROM	P80C52UFAA	P80C54UFAA	P80C58UFAA	P80C32UFAA	–40 to +85, Plastic Leaded Chip Carrier	5V	0 to 33	SOT187-2
OTP	P87C52UFAA	P87C54UFAA	P87C58UFAA					
ROM	P80C52UFPN	P80C54UFPN	P80C58UFPN	P80C32UFPN	–40 to +85, Plastic Dual In-line Package	5V	0 to 33	SOT129-1
OTP	P87C52UFPN	P87C54UFPN	P87C58UFPN					
ROM	P80C52UFBB	P80C54UFBB	P80C58UFBB	P80C32UFBB	–40 to +85, Plastic Quad Flat Pack	5V	0 to 33	SOT307-2
OTP	P87C52UFBB	P87C54UFBB	P87C58UFBB					

Note: For Multi Time Programmable devices, See P89C51RX+ Flash datasheet.

8XC51FA/FB/FC AND 80C51FA ORDERING INFORMATION

	MEMORY SIZE 8K × 8	MEMORY SIZE 16K × 8	MEMORY SIZE 32K × 8	ROMless	TEMPERATURE RANGE °C AND PACKAGE	VOLTAGE RANGE	FREQ. (MHz)	DWG. #
ROM	P83C51FA–4N	P83C51FB–4N	P83C51FC–4N	P80C51FA–4N	0 to +70, 40-Pin Plastic Dual In-line Pkg.	2.7V to 5.5V	0 to 16	SOT129-1
OTP	P87C51FA–4N	P87C51FB–4N	P87C51FC–4N					
ROM	P83C51FA–4A	P83C51FB–4A	P83C51FC–4A	P80C51FA–4A	0 to +70, 44-Pin Plastic Leaded Chip Carrier	2.7V to 5.5V	0 to 16	SOT187-2
OTP	P87C51FA–4A	P87C51FB–4A	P87C51FC–4A					
ROM	P83C51FA–4B	P83C51FB–4B	P83C51FC–4B	P80C51FA–4B	0 to +70, 44-Pin Plastic Quad Flat Pack	2.7V to 5.5V	0 to 16	SOT307-2
OTP	P87C51FA–4B	P87C51FB–4B	P87C51FC–4B					
ROM	P83C51FA–5N	P83C51FB–5N	P83C51FC–5N	P80C51FA–5N	–40 to +85, 40-Pin Plastic Dual In-line Pkg.	2.7V to 5.5V	0 to 16	SOT129-1
OTP	P87C51FA–5N	P87C51FB–5N	P87C51FC–5N					
ROM	P83C51FA–5A	P83C51FB–5A	P83C51FC–5A	P80C51FA–5A	–40 to +85, 44-Pin Plastic Leaded Chip Carrier	2.7V to 5.5V	0 to 16	SOT187-2
OTP	P87C51FA–5A	P87C51FB–5A	P87C51FC–5A					
ROM	P83C51FA–5B	P83C51FB–5B	P83C51FC–5B	P80C51FA–5B	–40 to +85, 44-Pin Plastic Quad Flat Pack	2.7V to 5.5V	0 to 16	SOT307-2
OTP	P87C51FA–5B	P87C51FB–5B	P87C51FC–5B					
ROM	P83C51FA–IN	P83C51FB–IN	P83C51FC–IN	P80C51FA–IN	0 to +70, 40-Pin Plastic Dual In-line Pkg.	5V	0 to 33	SOT129-1
OTP	P87C51FA–IN	P87C51FB–IN	P87C51FC–IN					
ROM	P83C51FA–IA	P83C51FB–IA	P83C51FC–IA	P80C51FA–IA	0 to +70, 44-Pin Plastic Leaded Chip Carrier	5V	0 to 33	SOT187-2
OTP	P87C51FA–IA	P87C51FB–IA	P87C51FC–IA					
ROM	P83C51FA–IB	P83C51FB–IB	P83C51FC–IB	P80C51FA–IB	0 to +70, 44-Pin Plastic Quad Flat Pack	5V	0 to 33	SOT307-2
OTP	P87C51FA–IB	P87C51FB–IB	P87C51FC–IB					
ROM	P83C51FA–JN	P83C51FB–JN	P83C51FC–JN	P80C51FA–JN	–40 to +85, 40-Pin Plastic Dual In-line Pkg.	5V	0 to 33	SOT129-1
OTP	P87C51FA–JN	P87C51FB–JN	P87C51FC–JN					
ROM	P83C51FA–JA	P83C51FB–JA	P83C51FC–JA	P80C51FA–JA	–40 to +85, 44-Pin Plastic Leaded Chip Carrier	5V	0 to 33	SOT187-2
OTP	P87C51FA–JA	P87C51FB–JA	P87C51FC–JA					
ROM	P83C51FA–JB	P83C51FB–JB	P83C51FC–JB	P80C51FA–JB	–40 to +85, 44-Pin Plastic Quad Flat Pack	5V	0 to 33	SOT307-2
OTP	P87C51FA–JB	P87C51FB–JB	P87C51FC–JB					

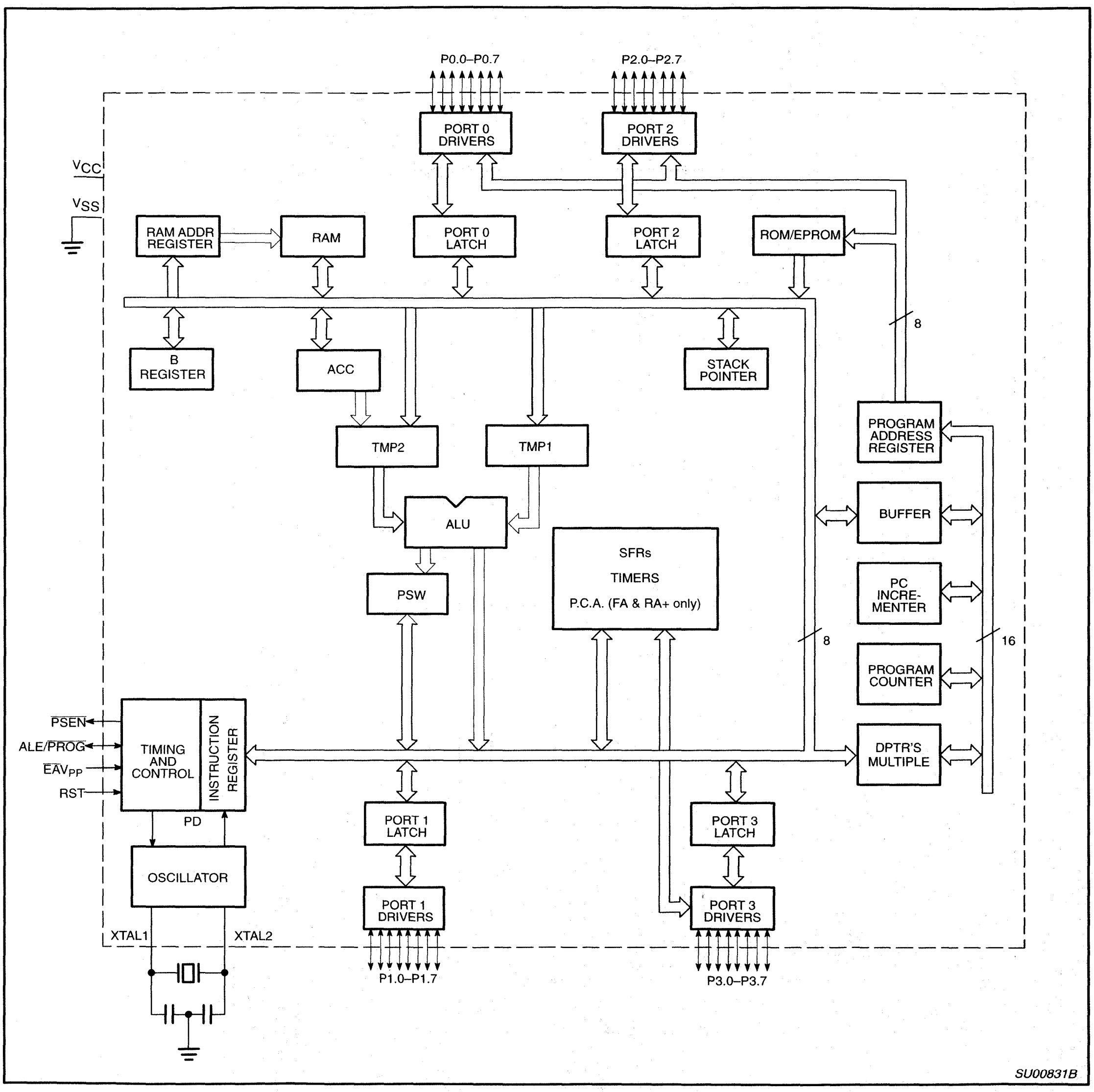
Note: For Multi Time Programmable devices, See P89C51RX+ Flash datasheet.

87C51RA+/RB+/RC+/RD+ AND 80C51RA+ ORDERING INFORMATION

	MEMORY SIZE 8K × 8	MEMORY SIZE 16K × 8	MEMORY SIZE 32K × 8	MEMORY SIZE 64K × 8	ROMless	TEMPERATURE RANGE °C AND PACKAGE	VOLTAGE RANGE	FREQ. (MHz)	DWG. #
ROM	P83C51RA+4N	P83C51RB+4N	P83C51RC+4N	P83C51RD+4N	P80C51RA+4N	0 to +70, 40-Pin Plastic Dual In-line Pkg.	2.7V to 5.5V	0 to 16	SOT129-1
OTP	P87C51RA+4N	P87C51RB+4N	P87C51RC+4N	P87C51RD+4N					
ROM	P83C51RA+4A	P83C51RB+4A	P83C51RC+4A	P83C51RD+4A	P80C51RA+4A	0 to +70, 44-Pin Plastic Leaded Chip Carrier	2.7V to 5.5V	0 to 16	SOT187-2
OTP	P87C51RA+4A	P87C51RB+4A	P87C51RC+4A	P87C51RD+4A					
ROM	P83C51RA+4B	P83C51RB+4B	P83C51RC+4B	P83C51RD+4B	P80C51RA+4B	0 to +70, 44-Pin Plastic Quad Flat Pack	2.7V to 5.5V	0 to 16	SOT307-2
OTP	P87C51RA+4B	P87C51RB+4B	P87C51RC+4B	P87C51RD+4B					
ROM	P83C51RA+5N	P83C51RB+5N	P83C51RC+5N	P83C51RD+5N	P80C51RA+5N	–40 to +85, 40-Pin Plastic Dual In-line Pkg.	2.7V to 5.5V	0 to 16	SOT129-1
OTP	P87C51RA+5N	P87C51RB+5N	P87C51RC+5N	P87C51RD+5N					
ROM	P83C51RA+5A	P83C51RB+5A	P83C51RC+5A	P83C51RD+5A	P80C51RA+5A	–40 to +85, 44-Pin Plastic Leaded Chip Carrier	2.7V to 5.5V	0 to 16	SOT187-2
OTP	P87C51RA+5A	P87C51RB+5A	P87C51RC+5A	P87C51RD+5A					
ROM	P83C51RA+5B	P83C51RB+5B	P83C51RC+5B	P83C51RD+5B	P80C51RA+5B	–40 to +85, 44-Pin Plastic Quad Flat Pack	2.7V to 5.5V	0 to 16	SOT307-2
OTP	P87C51RA+5B	P87C51RB+5B	P87C51RC+5B	P87C51RD+5B					
ROM	P83C51RA+IN	P83C51RB+IN	P83C51RC+IN	P83C51RD+IN	P80C51RA+IN	0 to +70, 40-Pin Plastic Dual In-line Pkg.	5V	0 to 33	SOT129-1
OTP	P87C51RA+IN	P87C51RB+IN	P87C51RC+IN	P87C51RD+IN					
ROM	P83C51RA+IA	P83C51RB+IA	P83C51RC+IA	P83C51RD+IA	P80C51RA+IA	0 to +70, 44-Pin Plastic Leaded Chip Carrier	5V	0 to 33	SOT187-2
OTP	P87C51RA+IA	P87C51RB+IA	P87C51RC+IA	P87C51RD+IA					
ROM	P83C51RA+IB	P83C51RB+IB	P83C51RC+IB	P83C51RD+IB	P80C51RA+IB	0 to +70, 44-Pin Plastic Quad Flat Pack	5V	0 to 33	SOT307-2
OTP	P87C51RA+IB	P87C51RB+IB	P87C51RC+IB	P87C51RD+IB					
ROM	P83C51RA+JN	P83C51RB+JN	P83C51RC+JN	P83C51RD+JN	P80C51RA+JN	–40 to +85, 40-Pin Plastic Dual In-line Pkg.	5V	0 to 33	SOT129-1
OTP	P87C51RA+JN	P87C51RB+JN	P87C51RC+JN	P87C51RD+JN					
ROM	P83C51RA+JA	P83C51RB+JA	P83C51RC+JA	P83C51RD+JA	P80C51RA+JA	–40 to +85, 44-Pin Plastic Leaded Chip Carrier	5V	0 to 33	SOT187-2
OTP	P87C51RA+JA	P87C51RB+JA	P87C51RC+JA	P87C51RD+JA					
ROM	P83C51RA+JB	P83C51RB+JB	P83C51RC+JB	P83C51RD+JB	P80C51RA+JB	–40 to +85, 44-Pin Plastic Quad Flat Pack	5V	0 to 33	SOT307-2
OTP	P87C51RA+JB	P87C51RB+JB	P87C51RC+JB	P87C51RD+JB					

Note: For Multi Time Programmable devices, See P89C51RX+ Flash datasheet.

BLOCK DIAGRAM



8-bit CMOS microcontroller families with FLASH program memory

89C52/54/58
89C51RA+/RB+/RC+/RD+

DESCRIPTION

Two different Single-Chip 8-Bit Microcontroller families are presented in this datasheet:

- 89C52/89C54/89C58
- 89C51RA+/89C51RB+/89C51RC+/89C51RD+

The 89C5X and 89C51RX+ families contain a non–volatile FLASH program memory (up to 64K bytes in the 89C51RD+) that is both parallel programmable and In–System Programmable. In–System Programming allows devices to alter their own program memory, in the actual end product, under software control. This opens up a range of applications that can include the ability to field update the application firmware.

Both families are Single-Chip 8-Bit Microcontrollers manufactured in advanced CMOS process and are derivatives of the 80C51 microcontroller family. All the devices have the same instruction set as the 80C51.

FLASH/ EPROM Memory Size (X by 8)	RAM Size (X by 8)	Programmable Timer Counter (PCA)	Hardware Watch Dog Timer
89C52/54/58			
8K/16K/32K	256	No	No
89C51RA+/RB+/RC+			
8K/16K/32K	512	Yes	Yes
89C51RD+			
64K	1024	Yes	Yes

The devices also have four 8-bit I/O ports, three 16-bit timer/event counters, a multi-source, four-priority-level, nested interrupt structure, an enhanced UART and on-chip oscillator and timing circuits. For systems that require extra memory capability up to 64k bytes, each can be expanded using standard TTL-compatible memories and logic.

The added features of the P89C51RX+ Family makes them even more powerful microcontrollers for applications that require pulse width modulation, high-speed I/O and up/down counting capabilities such as motor control.

FEATURES

- 80C51 Central Processing Unit
- On–chip FLASH Program Memory
- Speed up to 33MHz
- Full static operation
- RAM expandable externally to 64K bytes
- 4 level priority interrupt
- 6 or7 interrupt sources, depending on device
- Four 8-bit I/O ports
- Full-duplex enhanced UART
  - Framing error detection
  - Automatic address recognition
- Power control modes
  - Clock can be stopped and resumed
  - Idle mode
  - Power down mode
- Programmable clock out
- Second DPTR register
- Asynchronous port reset
- Low EMI (inhibit ALE)

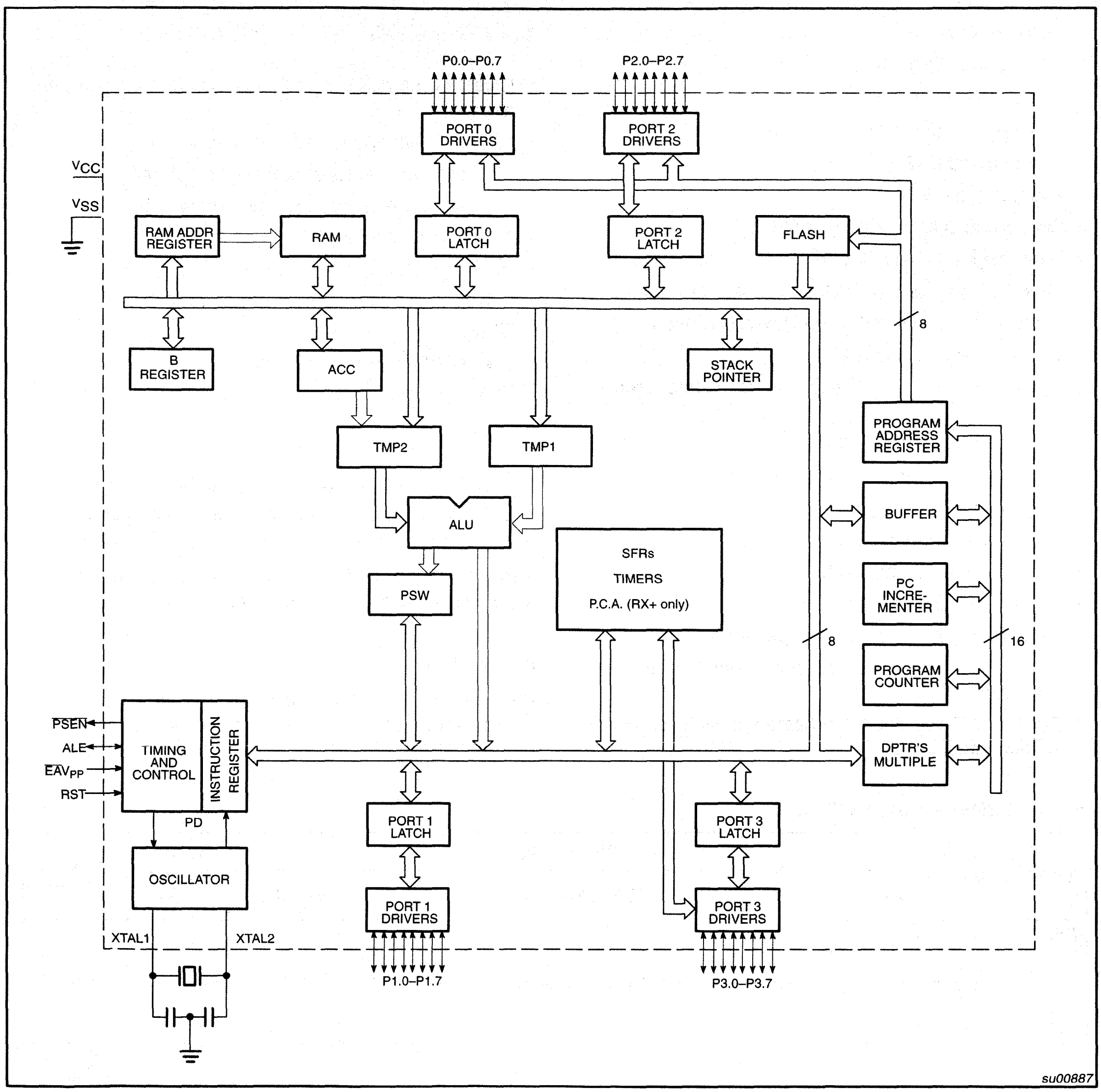
89C52/54/58 AND ORDERING INFORMATION

	MEMORY SIZE 8K × 8	MEMORY SIZE 16K × 8	MEMORY SIZE 32K × 8	TEMPERATURE RANGE °C AND PACKAGE	VOLTAGE RANGE	FREQ. (MHz)	DWG. #
FLASH	P89C52UBAA	P89C54UBAA	P89C58UBAA	0 to +70, Plastic Leaded Chip Carrier	5V	0 to 33	SOT187-2
FLASH	P89C52UBPN	P89C54UBPN	P89C58UBPN	0 to +70, Plastic Dual In-line Package	5V	0 to 33	SOT129-1
FLASH	P89C52UBBB	P89C54UBBB	P89C58UBBB	0 to +70, Plastic Quad Flat Pack	5V	0 to 33	SOT307-2
FLASH	P89C52UFA A	P89C54UFA A	P89C58UFA A	Plastic Leaded Chip Carrier	5V	0 to 33	SOT187-2
FLASH	P89C52UFPN	P89C54UFPN	P89C58UFPN	Plastic Dual In-line Package	5V	0 to 33	SOT129-1
FLASH	P89C52UFBB	P89C54UFBB	P89C58UFBB	Plastic Quad Flat Pack	5V	0 to 33	SOT307-2

87C51RA+/RB+/RC+/RD+ ORDERING INFORMATION

	MEMORY SIZE 8K × 8	MEMORY SIZE 16K × 8	MEMORY SIZE 32K × 8	MEMORY SIZE 64K × 8	TEMPERATURE RANGE °C AND PACKAGE	VOLTAGE RANGE	FREQ. (MHz)	DWG. #
FLASH	P89C51RA+IN	P89C51RB+IN	P89C51RC+IN	P89C51RD+IN	0 to +70, 40-Pin Plastic Dual In-line Pkg.	5V	0 to 33	SOT129-1
FLASH	P89C51RA+IA	P89C51RB+IA	P89C51RC+IA	P89C51RD+IA	0 to +70, 44-Pin Plastic Leaded Chip Carrier	5V	0 to 33	SOT187-2
FLASH	P89C51RA+IB	P89C51RB+IB	P89C51RC+IB	P89C51RD+IB	0 to +70, 44-Pin Plastic Quad Flat Pack	5V	0 to 33	SOT307-2
FLASH	P89C51RA+JN	P89C51RB+JN	P89C51RC+JN	P89C51RD+JN	40-Pin Plastic Dual In-line Pkg.	5V	0 to 33	SOT129-1
FLASH	P89C51RA+JA	P89C51RB+JA	P89C51RC+JA	P89C51RD+JA	44-Pin Plastic Leaded Chip Carrier	5V	0 to 33	SOT187-2
FLASH	P89C51RA+JB	P89C51RB+JB	P89C51RC+JB	P89C51RD+JB	44-Pin Plastic Quad Flat Pack	5V	0 to 33	SOT307-2

BLOCK DIAGRAM



Microcontrollers for TV and video (MTV)

83C145; 83C845 83C055; 87C055

1 FEATURES

- Masked ROM sizes:
  - 8 kbytes (83C845)
  - 12 kbytes (83C145)
  - 16 kbytes (83C055)
  - 16 kbytes OTP (87C055)
- RAM: 256 bytes
- On Screen Display (OSD) controller
- Three digital video outputs
- Multiplexer/mixer and background intensity controls
- Flexible formatting with OSD New Line option
- 128 × 10 bits display RAM
- Designed for reduced Radio Frequency Interference (RFI)
- Character generator ROM:
  - character format 18 lines × 14 dots
  - 60 visible characters
  - 4 special characters
- Eight text shadowing modes
- Text colour selectable per character
- Background colour selectable per word
- Background colour versus video selectable per character
- Eight 6-bit Pulse Width Modulators (PWM) for analog voltage integration
- One 14-bit PWM for high-precision voltage integration
- Digital-to-analog converter and comparator with 3 inputs multiplexer
- Nine dedicated I/Os plus 28 port bits (15 port bits with alternative uses)
- 4 high current open-drain port outputs
- 12 high voltage (+12 V) open-drain outputs
- Programmable video input and output polarities
- 80C51 instruction set
- No external memory capability
- Plastic shrink dual in-line package (0.07 inch centre pins)
- High-speed CMOS technology
- Power supply: 5 V ±10%.

2 DESCRIPTION

The 83C055, Microcontroller for Television and Video (MTV) applications, is a derivative of Philips' industry standard 80C51 microcontroller.

The 83C055 is intended for use as the central control mechanism in a television receiver or tuner.

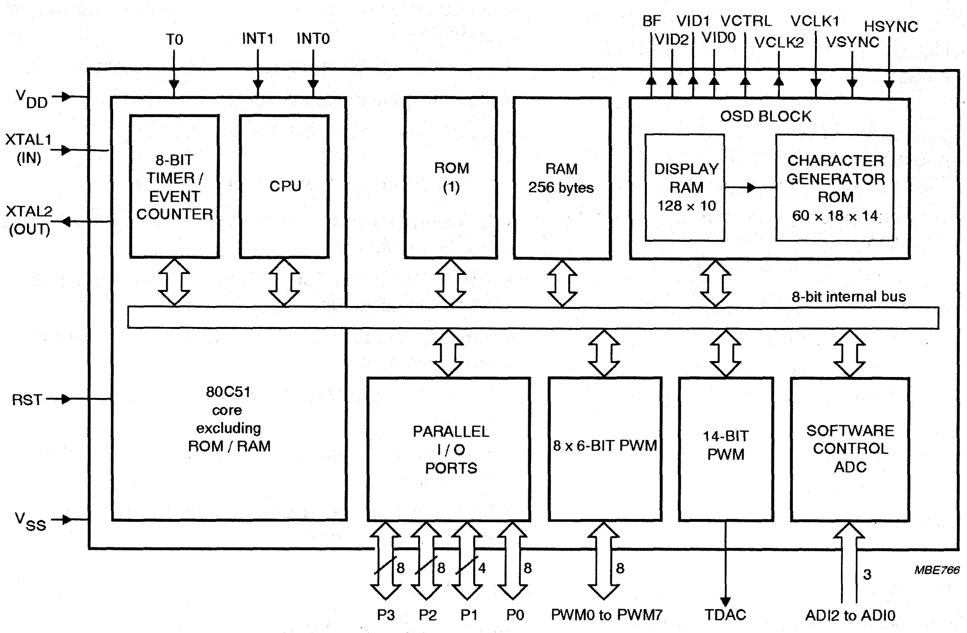
3 APPLICATIONS

Providing tuner functions and an OSD facility, it represents a next generation replacement for the currently available parts.

4 ORDERING INFORMATION

TYPE NUMBER	PACKAGE			TEMP. RANGE (°C)	FREQ. (MHz)
	NAME	DESCRIPTION	VERSION		
P83C055BBP	SDIP42	plastic shrink dual in-line package; 42 leads (600 mil)	SOT270-1	0 to +70	3.5 to 12
P87C055BBP					
P83C145BBP					
P83C845BBP					

5 BLOCK DIAGRAM



(1) ROM sizes: see Table 1.

Fig.1 Block diagram.

1 FEATURES

- 80C51 type core
- On-chip oscillator with a maximum frequency of 16 MHz (maximum 0.75 μs instruction cycle)
- A DDC interface:
  - That fully supports DDC1 with specific hardware
  - That is DDC2B, DDC2B+, DDC2AB (ACCESS.bus) compliant, based on a dedicated hardware I²C-bus interface.
  - Contains a specific AUX-RAM buffer with programmable size (128 or 256 bytes) that can be used for DDC operation and shared as system RAM
- Automatic mode detection by hardware to capture the following information:
  - HSYNC frequency with 12-bit resolution
  - VSYNC frequency with 12-bit resolution
  - HSYNC and VSYNC polarity
  - HSYNC and VSYNC presence; needed for the VESA Device Power Management Signalling (DPMS) standard
- On-chip sync processor comprising:
  - Composite sync separation
  - Free running mode
  - Clamping
  - Pattern generation
- Two specific ports for the software I²C-bus interface
- 4 analog voltage outputs derived from an 8-bit Digital-to-Analog Converter (DAC)
- Ten 8-bit Pulse Width Modulation (PWM) outputs for digital control application
- One 14-bit PWM output for digital control application
- One 4-bit Analog-to-Digital Converter (ADC) with 2 input channels (for keyboard interface)
- LED driver port (Port 0); eight port lines with 10 mA drive capability
- One 8-bit port only for I/O function
- 20 derivative I/O ports with the specific port type configuration in each alternative function
- Watchdog Timer with a programmable interval
- On-chip Power-on-reset for low power detection
- Special Idle and Power-down modes for reduced power operation
- Optimized for Electromagnetic Compatibility (EMC)
- Operating temperature: −25 to +85 °C
- Single power supply: 4.4 to 5.5 V.

1.1 Differences from the 80C51 core

- No external memory connection; signals $\overline{\text{EA}}$, ALE and $\overline{\text{PSEN}}$ are not present.
- Port 1, Port 2 and Port 3 (P3.0 to P3.3 only) mixed with other derivative functions.
- Timer 0/Counter 0 and Timer 1/Counter 1: external input is removed.
- External interrupt 0/INT0 replaced by Mode detection function.
- Standard serial interface (UART) and its control register are removed.
- Wake-up from Power-down mode is also possible by means of an interrupt.

1.2 Memory

Table 1 ROM/RAM sizes

DEVICE	MEMORY	
	ROM	RAM
P83C880	8 kbytes	512 bytes
P83C180	16 kbytes	512 bytes
P83C280	24 kbytes	512 bytes
P83C380	32 kbytes	512 bytes
P87C380 (OTP)	16 kbytes	512 bytes

2 GENERAL DESCRIPTION

The P83Cx80; P87C380 denotes the following types: P83C880, P83C180, P83C280, P83C380 and P87C380, hereafter referred to as the P83C880, are monitor microcontrollers of the 80C51 family, with DDC (DDC1, DDC2B, DDC2B+ and DDC2AB) interface to the PC host. The internal hardware can separate composite sync signals and detect the various display modes. The digital/analog voltage outputs can be used to control the video and deflection functions the monitor.

This data sheet details the specific properties of the P83C880, P83C180, P83C280, P83C380 and P87C380. The shared characteristics of the 80C51 family of microcontrollers are described in *"Data Handbook IC20"*, which should be read in conjunction with this data sheet.

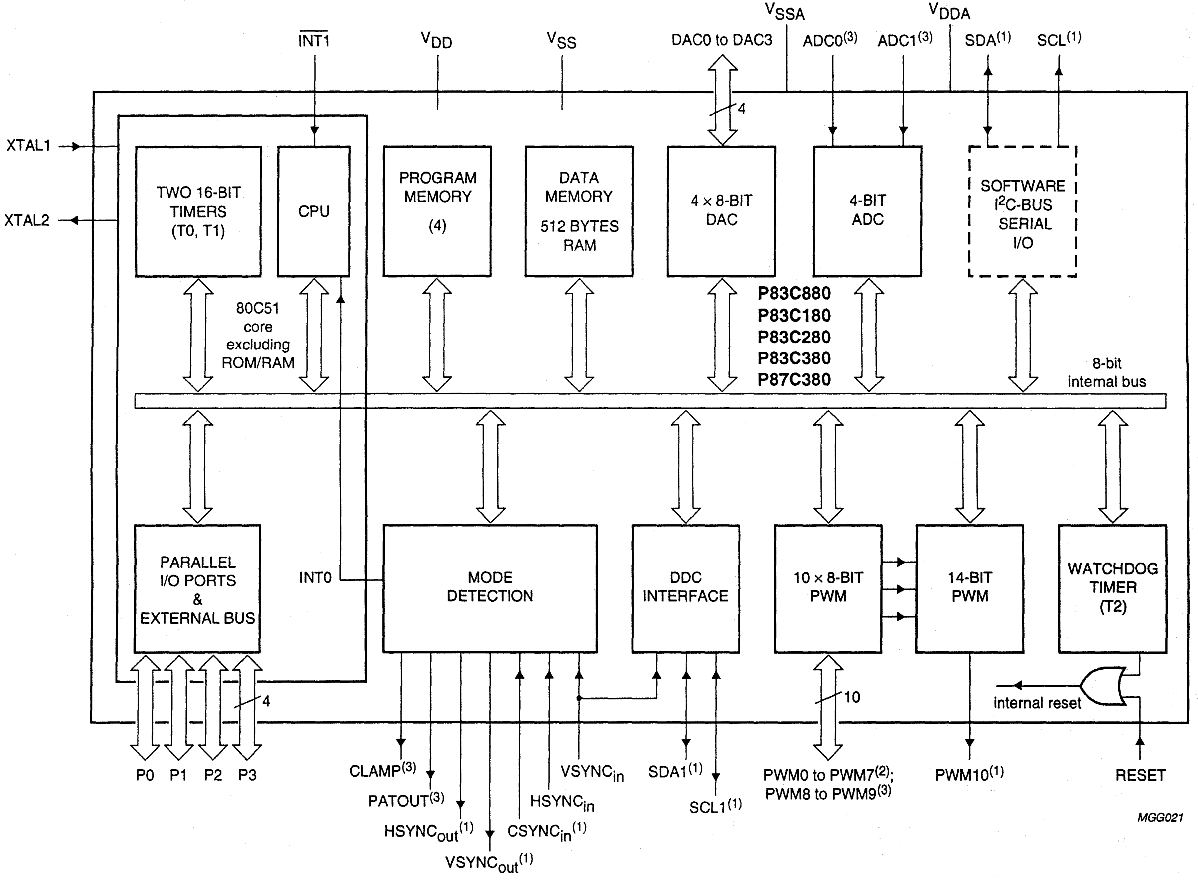
3 ORDERING INFORMATION

TYPE NUMBER	PACKAGE[1]			TEMPERATURE RANGE (°C)
	NAME	DESCRIPTION	VERSION	
P83C880	SDIP42	plastic shrink dual in-line package; 42 leads (600 mil)	SOT270-1	−25 to +85
P83C180				
P83C280				
P83C380				
P87C380 (OTP)				

Note

1. For emulation the package CLCC84 is used.

4 BLOCK DIAGRAM



(1) Alternative function of Port 1.

(2) Alternative function of Port 2.

(3) Alternative function of Port 3.

(4) **ROM**: 8 kbytes (P83C880); 16 kbytes (P83C180); 24 kbytes (P83C280); 32 kbytes (P83C380). **EPROM**: 16 kbytes only in the P87C180A.

Fig.1 Block diagram.

Low voltage 8-bit microcontrollers with UART

P80CL31; P80CL51

1 FEATURES

- Full static 80C51 Central Processing Unit
- 8-bit CPU, ROM, RAM, I/O in a 40-lead DIP, 40-lead VSO or 44-lead QFP package
- 128 bytes on-chip RAM Data Memory
- 4 kbytes on-chip ROM Program Memory for P80CL51
- External memory expandable up to 128 kbytes: RAM up to 64 kbytes and ROM up to 64 kbytes
- Four 8-bit ports; 32 I/O lines
- Two 16-bit Timer/Event counters
- On-chip oscillator suitable for RC, LC, quartz crystal or ceramic resonator
- Thirteen source, thirteen vector, nested interrupt structure with two priority levels
- Full duplex serial port (UART)
- Enhanced architecture with:
  - non-page oriented instructions
  - direct addressing
  - four 8-byte RAM register banks
  - stack depth limited only by available internal RAM (maximum 128 bytes)
  - multiply, divide, subtract and compare instructions
- Reduced power consumption through Power-down and Idle modes
- Wake-up via external interrupts at Port 1
- Frequency range: 0 to 16 MHz (P80C51: 3.5 MHz min.)
- Supply voltage: 1.8 to 6.0 V (P80C51: 5.0 V ±10%)
- Very low current consumption
- Operating ambient temperature range: –40 to +85 °C.

2 GENERAL DESCRIPTION

The P80CL31; P80CL51 (hereafter generally referred to as the P80CLx1) is manufactured in an advanced CMOS technology. The P80CLx1 has the same instruction set as the 80C51, consisting of over 100 instructions:
49 one-byte, 46 two-byte, and 16 three-byte. The device operates over a wide range of supply voltages and has low power consumption; there are two software selectable modes for power reduction: Idle and Power-down.
For emulation purposes, the P85CL000 (piggy-back version) with 256 bytes of RAM is recommended.

This data sheet details the specific properties of the P80CL31; P80CL51. For details of the 80C51 core see *"Data Handbook IC20"*.

2.1 Versions: P80CL31 and P80C51

The P80CL31 is the ROMless version of the P80CL51. The mask options on the P80CL31 are fixed as follows:

- All ports have option '1S' (standard, HIGH after reset)
- Oscillator option: Oscillator 3
- Power-on-reset option: OFF.

The P80C51 is a restricted-voltage range version of the P80CL51. The operating voltage is 5.0 V ±10%.

3 APPLICATIONS

The P80CLx1 is especially suited for real-time applications such as instrumentation, industrial control, intelligent computer peripherals and consumer products.
The P80CLx1 also functions as an arithmetic processor having facilities for both binary and BCD arithmetic plus bit-handling capabilities.

4 ORDERING INFORMATION

TYPE NUMBER[(1)]		PACKAGE		
ROMless	ROM	NAME	DESCRIPTION	VERSION
P80CL31HFP	P80CL51HFP	DIP40	plastic dual in-line package; 40 leads (600 mil)	SOT129-1
P80CL31HFT	P80CL51HFT	VSO40	plastic very small outline package; 40 leads	SOT158-1
P80CL31HFH	P80CL51HFH	QFP44	plastic quad flat package; 44 leads (lead length 1.3 mm); body 10 × 10 × 1.75 mm	SOT307-2
–	P80C51HFP	DIP40	plastic dual in-line package; 40 leads (600 mil)	SOT129-1
–	P80C51HFT	VSO40	plastic very small outline package; 40 leads	SOT158-1
–	P80C51HFH	QFP44	plastic quad flat package; 44 leads (lead length 1.3 mm); body 10 × 10 × 1.75 mm	SOT307-2

Note

1. Refer to the Order Entry Form (OEF) for this device for the full type number, including options/program.

5 BLOCK DIAGRAM

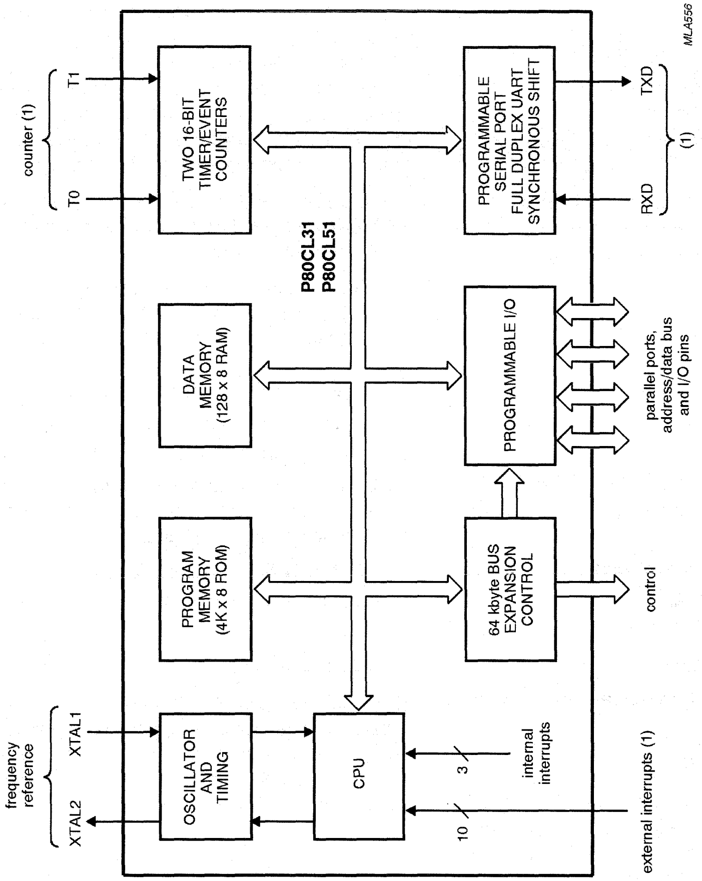

counter (1)
T1
T0
TWO 16-BIT TIMER/EVENT COUNTERS
PROGRAMMABLE SERIAL PORT FULL DUPLEX UART SYNCHRONOUS SHIFT
TXD
RXD
(1)
MLA556
P80CL31
P80CL51
DATA MEMORY (128 x 8 RAM)
PROGRAMMABLE I/O
parallel ports, address/data bus and I/O pins
PROGRAM MEMORY (4K x 8 ROM)
64 kbyte BUS EXPANSION CONTROL
control
frequency reference
XTAL1
XTAL2
OSCILLATOR AND TIMING
CPU
3
internal interrupts
10
external interrupts (1)

Fig.1 Block diagram.

(1) Pins shared with parallel port pins.

Low voltage 8-bit microcontrollers with I^{2}C-bus

P80CL410; P83CL410

1 FEATURES

- Full static 80C51 Central Processing Unit
- 8-bit CPU, ROM, RAM, I/O in a 40-lead DIP, 40-lead VSO or 44-lead QFP package
- 128 bytes on-chip RAM Data Memory
- 4 kbytes on-chip ROM Program Memory for P83CL410
- External memory expandable up to 128 kbytes: RAM up to 64 kbytes and ROM up to 64 kbytes
- Four 8-bit ports; 32 I/O lines
- Two 16-bit Timer/Event counters
- On-chip oscillator suitable for RC, LC, quartz crystal or ceramic resonator
- Thirteen source, thirteen vector, nested interrupt structure with two priority levels
- I^{2}C-bus interface for serial transfer on two lines
- Enhanced architecture with:
  - non-page oriented instructions
  - direct addressing
  - four 8-byte RAM register banks
  - stack depth limited only by available internal RAM (maximum 128 bytes)
  - multiply, divide, subtract and compare instructions
- Reduced power consumption through Power-down and Idle modes
- Wake-up via external interrupts at Port 1
- Frequency range: DC to 12 MHz
- Supply voltage: 1.8 to 6.0 V
- Very low current consumption
- Operating ambient temperature range: –40 to +85 °C.

2 GENERAL DESCRIPTION

The P80CL410; P83CL410 (hereafter generally referred to as the P8xCL410) is manufactured in an advanced CMOS technology that allows the device to operate at voltages down to 1.8 V and at frequencies down to DC.
The P8xCL410 has the same instruction set as the 80C51.

The P8xCL410 features 4 kbyte ROM (83CL410), 128 bytes RAM (both ROM and RAM are externally expandable to 64 kbytes), four 8-bit ports, two 16-bit timer/counter, an I^{2}C serial interface, a thirteen source two priority level nested interrupt structure, and on-chip oscillator circuitry suitable for quartz crystal, ceramic resonator, RC, or LC. The device operates over a wide range of supply voltages and has low power consumption.

The 8xCL410 has two reduced power modes that are the same as those on the standard 80C51.

The special reduced power feature of this device is that it can be stopped and then restarted. Running from an external clock source, the clock can be stopped and after a period of time restarted. The 8xCL410 will resume operation from where it was when the code stopped with no loss of internal state, RAM contents, or Special Function Register contents.
If the internal oscillator is used the device cannot be stopped and started, but the Power-down mode can be used to achieve similar power savings, without loss of on-chip RAM and Special Function Register values.
The Power-down mode can be terminated via an interrupt.

This data sheet details the specific properties of the P80CL410; P83CL410. For details of the 80C51 core see *“Data Handbook IC20”.*

For emulation purposes, the P85CL000 (piggy-back version) with 256 bytes of RAM is recommended. Details are given in Chapter 22.

2.1 ROMless version: P80CL410

The P80CL410 is the ROMless version of the P83CL410. The mask options on the P80CL410 are fixed as follows:

- All ports have option ‘1S’ (standard port, HIGH after reset), except ports P1.6 and P1.7 which have option ‘2S’ (open-drain, HIGH after reset)
- Oscillator option: Oscillator 3
- Power-on-reset option: OFF.

3 APPLICATIONS

The P8xCL410 is an 8-bit general purpose microcontroller especially suited for battery-powered applications.
The P8xCL410 also functions as an arithmetic processor having facilities for both binary and BCD arithmetic plus bit-handling capabilities.

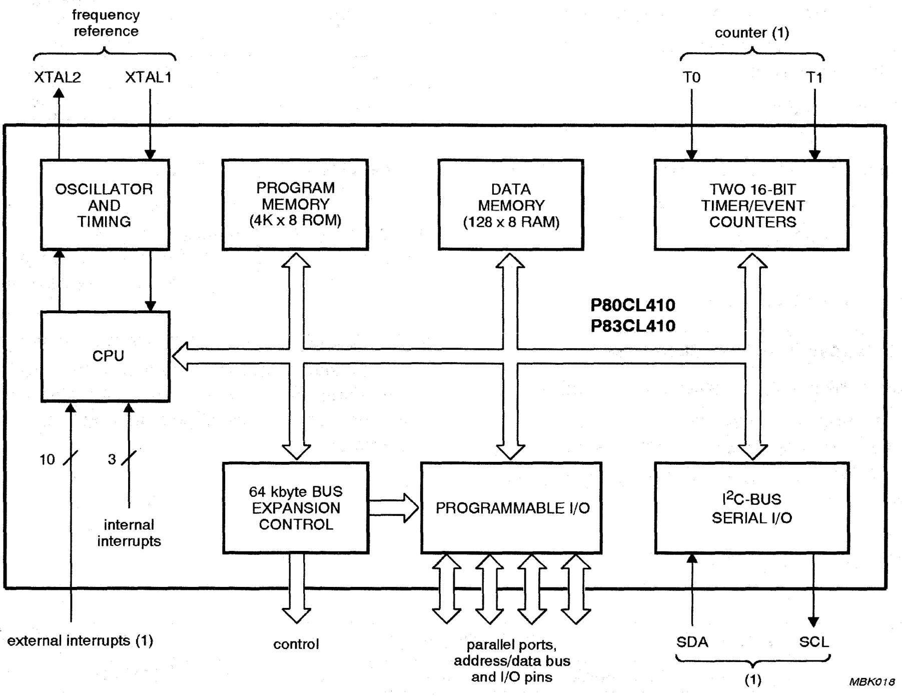
4 ORDERING INFORMATION

TYPE NUMBER[1]		PACKAGE		
ROMless	ROM	NAME	DESCRIPTION	VERSION
P80CL410HFP	P83CL410HFP	DIP40	plastic dual in-line package; 40 leads (600 mil)	SOT129-1
P80CL410HFT	P83CL410HFT	VSO40	plastic very small outline package; 40 leads	SOT158-1
–	P83CL410HFH	QFP44	plastic quad flat package; 44 leads (lead length 1.3 mm); body 10 × 10 × 1.75 mm	SOT307-2

Note

1. Refer to the Order Entry Form (OEF) for this device for the full type number, including options/program.

5 BLOCK DIAGRAM



(1) Pins shared with parallel port pins.

Fig.1 Block diagram.

8-bit microcontrollers with LCD-driver P83C434; P83C834

1 FEATURES

- 80C51 type core
- System clock derived from an internal free running Current Controlled Oscillator (CCO); no external components required. Clock frequency can be adjusted by software.
- Optimized for EMC (Electromagnetic Compatibility)
- Clock frequency f_{clk} = 1 to 12 MHz
- 12 I/O lines, quasi-bidirectional
- Gated interrupt on 8 I/O lines:
  – P0.0 to P0.3 when LOW
  – P0.4 to P0.7 when LOW or HIGH
- LCD driver clock, 32 kHz, which also provides the time base for a Real Time Clock
- 1-second interrupt by internal 15-bit counter
- On-chip Liquid Crystal Display (LCD) drivers with 26 outputs, comprising:
  – 22, 23 or 24 segment drivers
  – 1 to 4 backplanes
- LCD multiplexing rates: 1 : 1 (static), 1 : 2, 1 : 3 or 1 : 4
- Operating temperature: –40 to +85 °C
- Single power supply:
  – Operating voltage: 3.5 to 5.5 V
  – Power-down mode: 1.8 V.

1.1 Differences from the 80C51 core

- Port 0 quasi-bidirectional instead of open-drain.
- No external memory connection. Signals $\overline{EA}$, ALE and $\overline{PSEN}$ are not present.
- Port 1, Port 2 (pins P2.4 to P2.7) and Port 3 are not present.
- Timer 0 external input replaced by a direct connection to the 32 kHz oscillator output.
- Timer 1 external input is connected to pin P0.0 (Port 0); alternate function of P0.0.
- Standard serial interface and its control register is not present.
- Adjustable on-chip oscillator without external components.
- Wake-up from Power-down mode is also possible by means of an interrupt.
- Extended external interrupts.

1.2 Memory

Table 1 ROM/RAM sizes

DEVICE	MEMORY	
	ROM	RAM
P83C434	4 kbytes	128 bytes
P83C834	8 kbytes	256 bytes

2 GENERAL DESCRIPTION

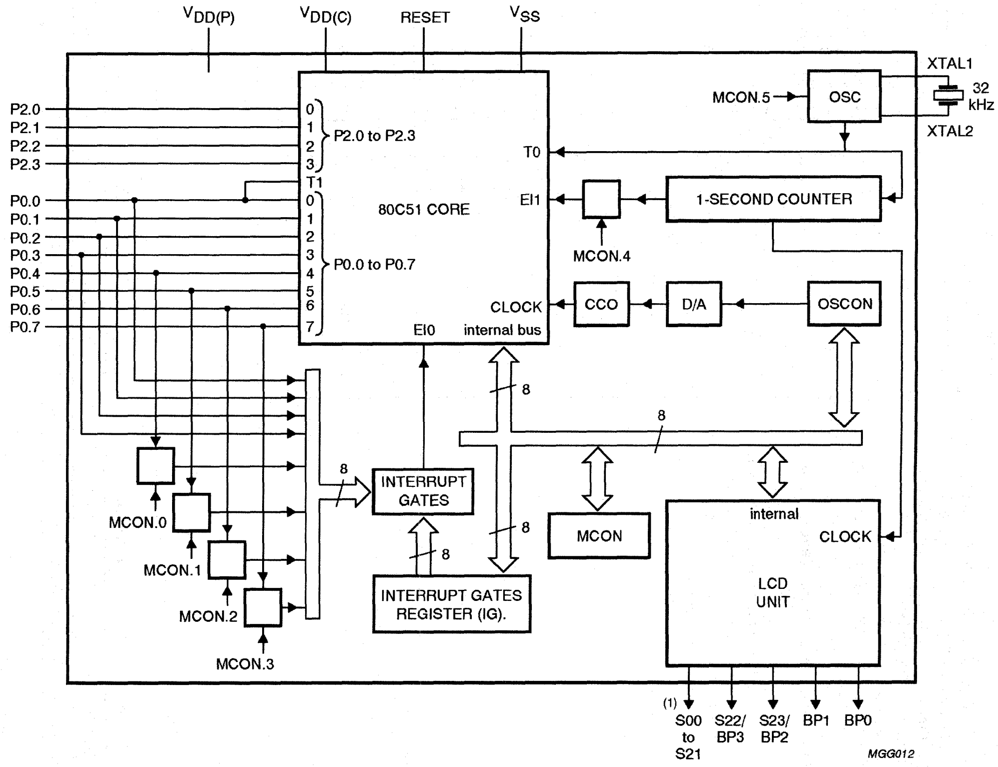
The P83C434 and P83C834 are low-cost microcontrollers of the 80C51 family, with LCD drivers.
Main application is in the user-interface (keypad, display) of consumer products, e.g. portable radios, CD-players, etc.

This data sheet details the specific properties of the P83C434 and P83C834. The shared characteristics of the 80C51 family of microcontrollers are described in *"Data Handbook IC20"*, which should be read in conjunction with this data sheet.

3 ORDERING INFORMATION

TYPE NUMBER	PACKAGE			TEMP. RANGE (°C)
	NAME	DESCRIPTION	VERSION	
P83C434CFP; P83C834CFP	SDIP42	plastic shrink dual in-line package; 42 leads (600 mil)	SOT270-1	–40 to +85
	QFP44	plastic quad flat package; 44 leads (lead length 1.3 mm); body 10 × 10 × 1.75 mm	SOT307-2	

4 BLOCK DIAGRAM



(1) Drive lines S00 and S21 are not available with the SDIP42 (SOT270-1) package.

Fig.1 Block diagram.

CMOS single-chip 8-bit microcontrollers 80C451/83C451/87C451

DESCRIPTION

The Philips 8XC451 is an I/O expanded single-chip microcontroller fabricated with Philips high-density CMOS technology. Philips epitaxial substrate minimizes latch-up sensitivity.

The 8XC451 (includes the 80C451, 87C451 and 83C451) is a functional extension of the 87C51 microcontroller with three additional I/O ports and four I/O control lines for a total of 68 pins. Four control lines associated with port 6 facilitate high-speed asynchronous I/O functions.

The 8XC451 includes a 4k × 8 ROM (83C451) EPROM (87C451), a 128 × 8 RAM, 56 I/O, two 16-bit timer/counters, a five source, two priority level, nested interrupt structure, a serial I/O port for either a full duplex UART, I/O expansion, or multi-processor communications, and on-chip oscillator and clock circuits. The 80C451 ROMless version includes all of the 83C451 features except the on-board 4k × 8 ROM.

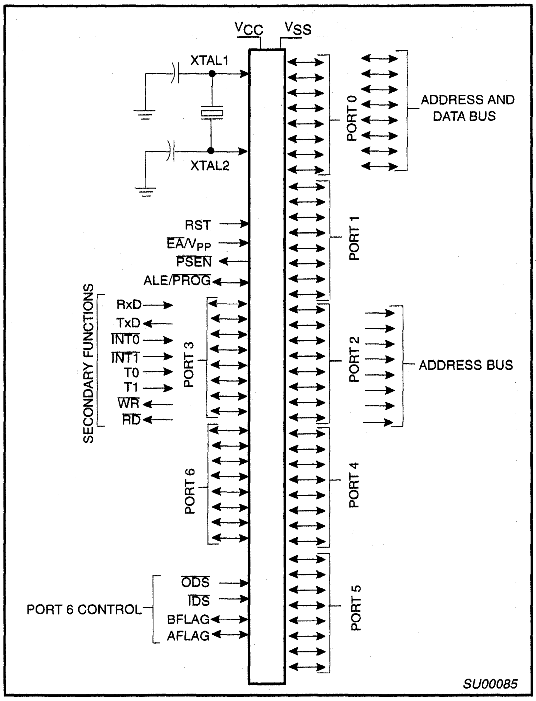
The 87C451 has 4k of EPROM on-chip as program memory and is otherwise identical to the 83C451.

The 8XC451 has two software selectable modes of reduced activity for further power reduction; idle mode and power-down mode. Idle mode freezes the CPU while allowing the RAM, timers, serial port, and interrupt system to continue functioning. Power-down mode freezes the oscillator, causing all other chip functions to be inoperative while maintaining the RAM contents.

FEATURES

- 80C51 based architecture
- Seven 8-bit I/O ports
- Port 6 features:
  - Eight data pins
  - Four control pins
  - Direct MPU bus interface
  - Parallel printer interface
- On the microcontroller:
  - 4k × 8 ROM (83C451)
    4k × 8 EPROM (87C451)
    ROMless version (80C451)
  - 128 × 8 RAM
  - Two 16-bit counter/timers
  - Two external interrupts
- External memory addressing capability
  - 64k ROM and 64k RAM
- Low power consumption:
  - Normal operation: less than 24mA at 5V, 12MHz
  - Idle mode
  - Power-down mode

LOGIC SYMBOL



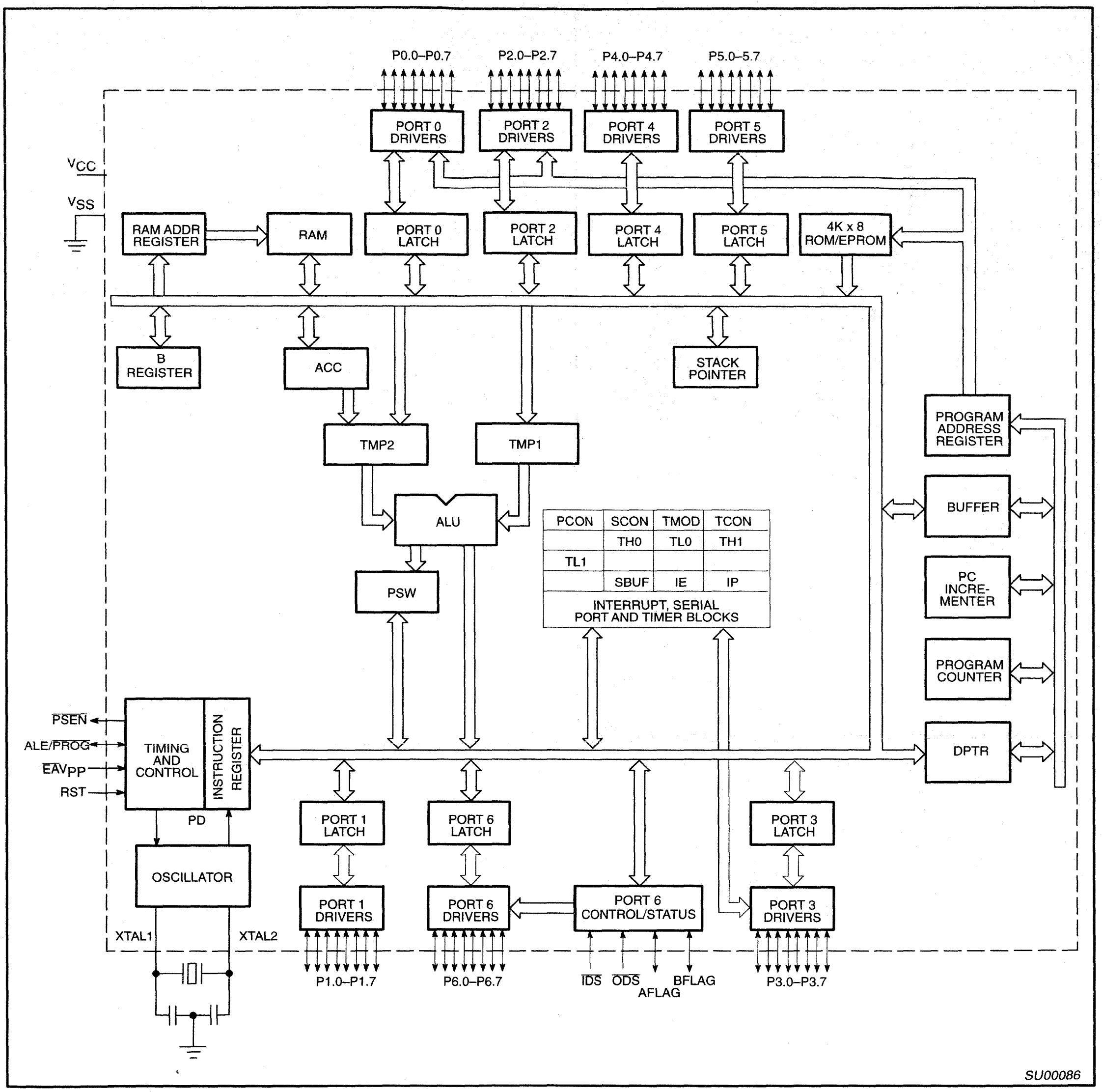
ORDERING INFORMATION

ROMless	ROM	EPROM[1]		TEMPERATURE RANGE °C AND PACKAGE	FREQ MHz	DRAWING NUMBER
SC80C451CCA68	SC83C451CCA68	SC87C451CCA68	OTP	0 to +70, Plastic Leaded Chip Carrier,	3.5 to 12	SOT188-3
SC80C451CGA68	SC83C451CGA68	SC87C451CGA68	OTP	0 to +70, Plastic Leaded Chip Carrier	3.5 to 16	SOT188-3

NOTE:

[1] OTP = One Time Programmable

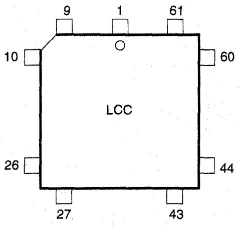
BLOCK DIAGRAM



PCON	SCON	TMOD	TCON
	TH0	TL0	TH1
TL1			
	SBUF	IE	IP

INTERRUPT, SERIAL PORT AND TIMER BLOCKS

PIN CONFIGURATION



Pin	Function	Pin	Function	Pin	Function
1	$\overline{\text{EA}}$/V_{PP}	24	P4.2	47	P5.3
2	P2.0/A8	25	P4.1	48	P5.4
3	P2.1/A9	26	P4.0	49	P5.5
4	P2.2/A10	27	P1.0	50	P5.6
5	P2.3/A11	28	P1.1	51	P5.7
6	P2.4/A12	29	P1.2	52	XTAL2
7	P2.5/A13	30	P1.3	53	XTAL1
8	P2.6/A14	31	P1.4	54	V_{SS}
9	P2.7/A15	32	P1.5	55	$\overline{\text{ODS}}$
10	P0.7/AD7	33	P1.6	56	$\overline{\text{IDS}}$
11	P0.6/AD6	34	P1.7	57	BFLAG
12	P0.5/AD5	35	RST	58	AFLAG
13	P0.4/AD4	36	P3.0/RxD	59	P6.0
14	P0.3/AD3	37	P3.1/TxD	60	P6.1
15	P0.2/AD2	38	P3.2/$\overline{\text{INT0}}$	61	P6.2
16	P0.1/AD1	39	P3.3/$\overline{\text{INT1}}$	62	P6.3
17	P0.0/AD0	40	P3.4/T0	63	P6.4
18	V_{CC}	41	P3.5/T1	64	P6.5
19	P4.7	42	P3.6/$\overline{\text{WR}}$	65	P6.6
20	P4.6	43	P3.7/$\overline{\text{RD}}$	66	P6.7
21	P4.5	44	P5.0	67	$\overline{\text{PSEN}}$
22	P4.4	45	P5.1	68	ALE/$\overline{\text{PROG}}$
23	P4.3	46	P5.2		

SU00084A

CMOS single-chip 8-bit microcontrollers | 83C453/87C453

DESCRIPTION

The Philips 8XC453 is an I/O expanded single-chip microcontroller fabricated with Philips high-density CMOS technology. Philips epitaxial substrate minimizes latch-up sensitivity.

The 8XC453 is a functional extension of the 87C51 microcontroller with three additional I/O ports and four I/O control lines. The 8XC453 is available in 68-pin LCC packages. Four control lines associated with port 6 facilitate high-speed asynchronous I/O functions.

The 87C453 includes an 8k × 8 EPROM, a 256 × 8 RAM, 56 I/O lines, two 16-bit timer/counters, a seven source, two priority level, nested interrupt structure, a serial I/O port for either a full duplex UART, I/O expansion, or multi-processor communications, and on-chip oscillator and clock circuits.

The 87C453 has two software selectable modes of reduced activity for further power reduction; idle mode and power-down mode. Idle mode freezes the CPU while allowing the RAM, timers, serial port, and interrupt system to continue functioning. Power-down mode freezes the oscillator, causing all other chip functions to be inoperative while maintaining the RAM contents.

FEATURES

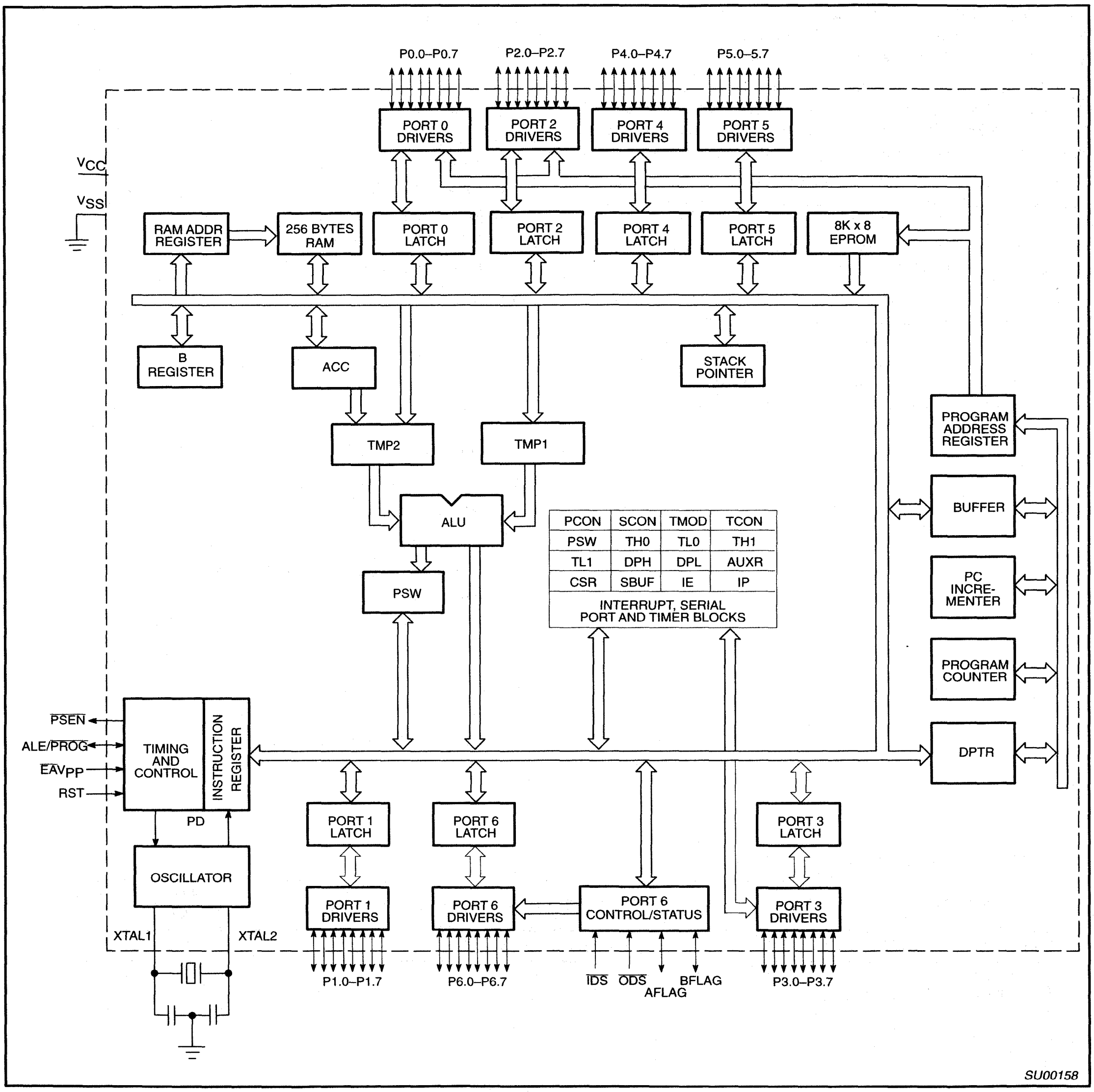
- 80C51 based architecture
- Seven 8-bit I/O ports
- Port 6 features:
  - Eight data pins
  - Four control pins
  - Direct MPU bus interface
  - ISA Bus Interface
  - Parallel printer interface
  - IBF and $\overline{\text{OBF}}$ interrupts
  - A flag latch on host write
- On the microcontroller:
  - 8k × 8 EPROM
    Quick pulse programming algorithm
    Two-level program security system
  - 256 × 8 RAM
  - Two 16-bit counter/timers
  - Two external interrupts
- External memory addressing capability
  - 64k ROM and 64k RAM
- Low power consumption:
  - Normal operation: less than 24mA at 5V, 16MHz
  - Idle mode
  - Power-down mode
- Reduced EMI
- Full-duplex enhanced UART
  - Framing error detection
  - Automatic address recognition

ORDERING INFORMATION

EPROM[1]		ROM	TEMPERATURE °C AND PACKAGE	FREQ. (MHz)	PKG. DWG #
P87C453EBAA	OTP	P83C453EBAA	68–Pin Plastic Leaded Chip Carrier, 0 to +70	3.5 to 16	SOT188-3

NOTE:
1. OTP = One-Time Programmable EPROM.

BLOCK DIAGRAM



PCON	SCON	TMOD	TCON
PSW	TH0	TL0	TH1
TL1	DPH	DPL	AUXR
CSR	SBUF	IE	IP

CMOS single-chip 8-bit microcontroller 87C524

DESCRIPTION

The 87C524 single-chip 8-bit microcontroller is manufactured in an advanced CMOS process and is a derivative of the 80C51 microcontroller family. The 87C524 has the same instruction set as the 80C51.

This device provides architectural enhancements that make it applicable in a variety of applications in consumer, telecom and general control systems, especially in those systems which need large ROM and RAM capacity on-chip.

The 87C524 contains a 16k × 8 EPROM, a 512 × 8 RAM, four 8-bit I/O ports, two 16-bit timer/event counters (identical to the timers of the 80C51), a 16-bit timer (identical to the timer 2 of the 80C52), a watchdog timer with a separate oscillator, a multi-source, two-priority-level, nested interrupt structure, two serial interfaces (UART and I^{2}C-bus), and on-chip oscillator and timing circuits.

In addition, the 87C524 has two software selectable modes of power reduction—idle mode and power-down mode. 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.

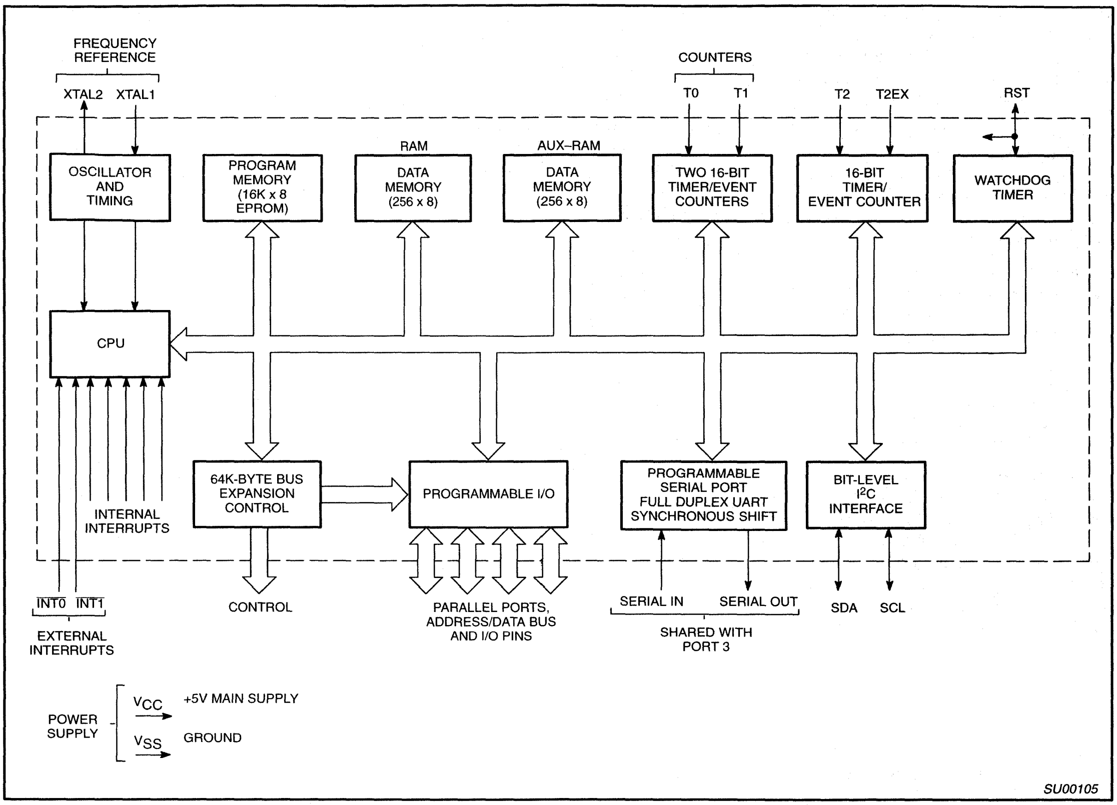
FEATURES

- 80C51 instruction set
  - 16k × 8 EPROM
  - 512 × 8 RAM
  - Memory addressing capability 64k ROM and 64k RAM
  - Three 16-bit counter/timers
  - On-chip watchdog timer with oscillator
  - Full duplex UART
  - I^{2}C serial interface
- Power control modes:
  - Idle mode
  - Power-down mode
  - Warm start from power-down
- CMOS and TTL compatible
- Two speed ranges at V_{CC} = 5V ±10%
  - 3.5 to 16MHz
- OTP package available
- EPROM code protection

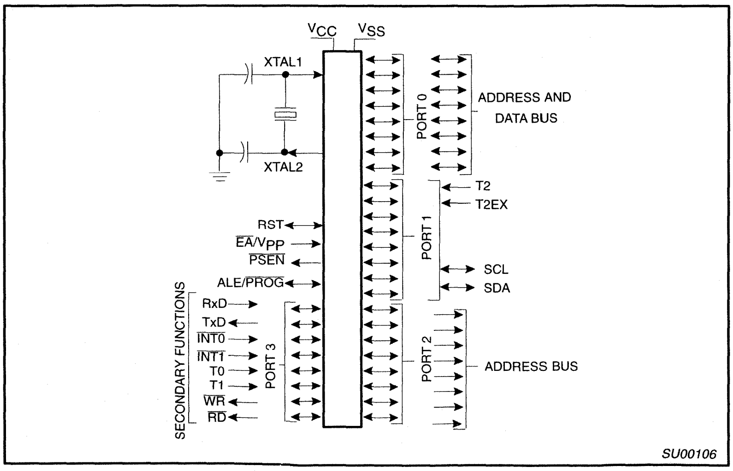
ORDERING INFORMATION

EPROM	TEMPERATURE RANGE °C AND PACKAGE	FREQUENCY	DRAWING NUMBER
P87C524EBA A	0 to +70, Plastic Leaded Chip Carrier	16MHz	SOT187-2
P87C524EBB B	0 to +70, Plastic Quad Flat Pack	16MHz	SOT307-2

BLOCK DIAGRAM



LOGIC SYMBOL



CMOS single-chip 8-bit microcontroller 87C528

DESCRIPTION

The 87C528 single-chip 8-bit microcontroller is manufactured in an advanced CMOS process and is a derivative of the 80C51 microcontroller family. The 87C528 has the same instruction set as the 80C51. Three versions of the derivative exist:

- 83C528—32k bytes mask programmable ROM
- 80C528—ROMless version of the 83C528
- 87C528—32k bytes EPROM

This device provides architectural enhancements that make it applicable in a variety of applications in consumer, telecom and general control systems, especially in those systems which need large ROM and RAM capacity on-chip.

The 87C528 contains a 32k × 8 EPROM, a 512 × 8 RAM, four 8-bit I/O ports, two 16-bit timer/event counters (identical to the timers of the 80C51), a 16-bit timer (identical to the timer 2 of the 80C52), a watchdog timer with a separate oscillator, a multi-source, two-priority-level, nested interrupt structure, two serial interfaces (UART and I^{2}C-bus), and on-chip oscillator and timing circuits.

In addition, the 87C528 has two software selectable modes of power reduction—idle mode and power-down mode. 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.

FEATURES

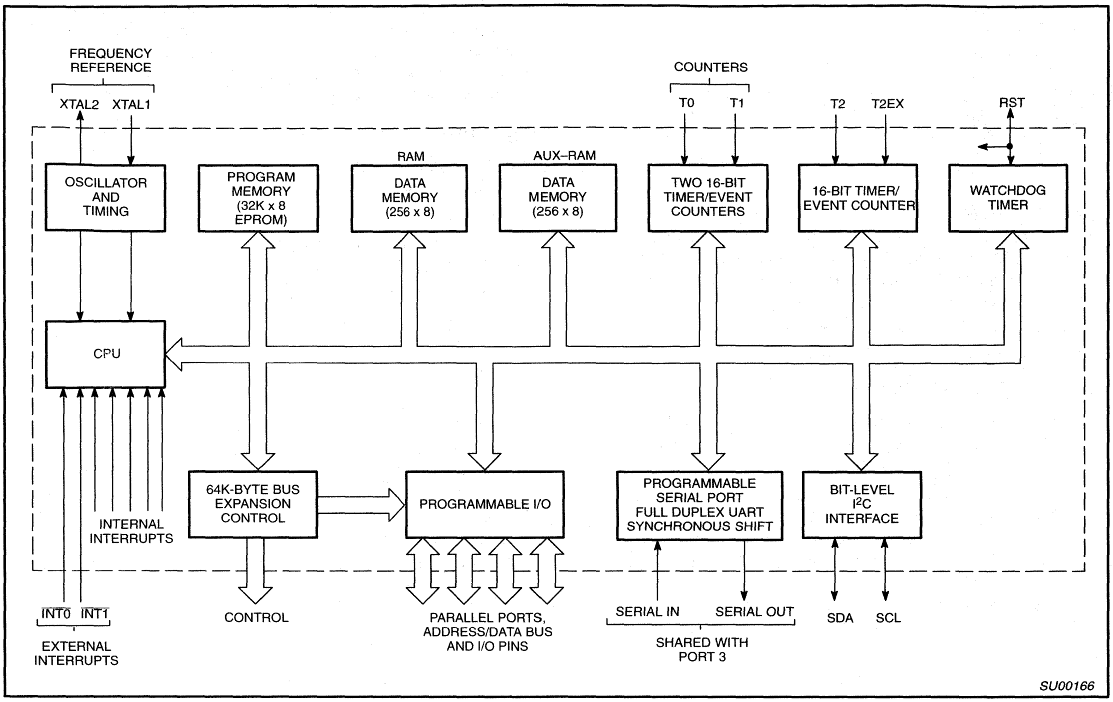
- 80C51 instruction set
  - 32k × 8 EPROM
  - 512 × 8 RAM
  - Memory addressing capability
    64k ROM and 64k RAM
  - Three 16-bit counter/timers
  - On-chip watchdog timer with oscillator
  - Full duplex UART
  - I^{2}C serial interface
- Power control modes:
  - Idle mode
  - Power-down mode
  - Warm start from power-down
- CMOS and TTL compatible
- Extended temperature ranges
- EPROM code protection
- OTP package available
- 16MHz speed at V_{CC} = 5V

ORDERING INFORMATION

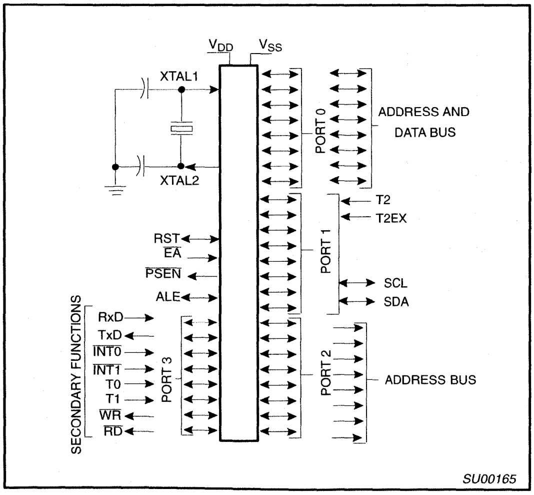
EPROM	Drawing Number	TEMPERATURE °C RANGE AND PACKAGE	FREQ MHz
P87C528EBP N	SOT129-1	0 to +70, Plastic Dual In-line Package	16
P87C528EBA A	SOT187-2	0 to +70, Plastic Leaded Chip Carrier	16
P87C528EBB B	SOT307-2[2]	0 to +70, Plastic Quad Flat Pack	16
P87C528EFP N	SOT129-1	–40 to +85, Plastic Dual In-line Package	16
P87C528EFB B	SOT307-2[2]	–40 to +85, Plastic Quad Flat Pack	16

1. For ROM & ROMless devices see data sheet P8X524/528
2. SOT311 replaced by SOT307-2.

BLOCK DIAGRAM



LOGIC SYMBOL



8-bit microcontrollers P83C524; P80C528; P83C528

1 FEATURES

- 80C51 CPU
- 32 kbytes on-chip ROM, expandable externally to 64 kbytes Program Memory address space
- P83C524:
  - 16 kbytes on-chip ROM, expandable externally from 32 kbytes to 64 kbytes Program Memory address space (address space 16 k to 32 k not usable)
- P80C528:
  - ROMless version of P83C528
- P83C528:
  - 32 kbytes on-chip ROM, expandable externally from 32 kbytes to 64 kbytes Program Memory address space
- EPROM versions are available: see separate data sheet P87C524 and P87C528
- 512 bytes on-chip RAM, expandable externally to 64 kbytes Data Memory address space
- Four 8-bit I/O ports
- Full-duplex UART compatible with the standard 80C51 and the 8052
- Two standard 16-bit timer/counters
- An additional 16-bit timer (functionally equivalent to the timer 2 of the 8052)
- On-chip Watchdog Timer (WDT) with an own oscillator
- Bit-level I^{2}C-bus hardware serial I/O Port
- 7-source and 7-vector interrupt structure with 2 priority levels
- Up to 3 external interrupt request inputs
- Two programmable power reduction modes (Idle and Power-down)
- Termination of Idle mode by any interrupt, external or WDT (watchdog) reset
- Wake-up from Power-down by external interrupt, external or WDT reset
- ROM code protection
- XTAL frequency range: 3.5 MHz to 16 MHz and 3.5 MHz to 24 MHz
- All packaging pin-outs fully compatible to the standard 8051/8052.

2 GENERAL DESCRIPTION

The P83C524 and P83C528 single-chip 8-bit microcontrollers are manufactured in an advanced CMOS process and are derivatives of the PCB80C51 microcontroller family. These devices provide architectural enhancements that make them applicable in a variety of applications in general control systems, especially in those systems which need a large ROM and RAM capacity on chip.

The P83C524 and P83C528 contain a non-volatile 16 k × 8 respectively 32 k × 8 read-only program memory, a volatile 512 bytes × 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 16-bit timer (identical to the timer 2 of the 8052), a multi-source, two-priority-level, nested interrupt structure, two serial interfaces (UART and bit-level I^{2}C-bus), a watchdog timer (WDT) with a separate oscillator, an on-chip oscillator and timing circuits. For systems that require extra capability, the P83C524 and P83C528 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 P83C524 and P83C528 have the same instruction set as the PCB80C51 which consists of over 100 instructions: 49 one-byte, 46 two-byte and 16 three-byte. With a 16 MHz crystal, 58% of the instructions are executed in 750 ns and 40% in 1.5 μs. Multiply and divide instructions require 3 μs.

3 QUICK REFERENCE DATA

SYMBOL	PARAMETER	CONDITION	MIN.	MAX.	UNIT
P83C524, P80C528, P83C528 (see characteristics tables for extended temperature range versions)					
V_{DD}	supply voltage range		4.5	5.5	V
I_{DD}	supply current: operating modes 16 MHz	V_{DD} = 5.5 V, f_{CLK} = 16 MHz	–	33	mA
I_{ID}	supply current: Idle mode 16 MHz	V_{DD} = 5.5 V, f_{CLK} = 16 MHz	–	6	mA
I_{PD}	supply current: Power-down mode	2V ≤ V_{PD} ≤ V_{DD} max.	–	100	µA
P_{tot}	total power dissipation		–	1	W
T_{stg}	storage temperature range		–65	+150	°C
T_{amb}	operating ambient temperature range		–40	+85	°C

4 ORDERING INFORMATION

EXTENDED TYPE NUMBER	PACKAGE			TEMPERATURE RANGE (°C)	FREQ. (MHZ)
	NAME	DESCRIPTION	VERSION		
ROMless					
P80C528EBP	DIP40	plastic dual in-line package; 40 leads (600 mil)	SOT129-1	0 to +70	3.5 to 16
P80C528EFP				–40 to +85	
P80C528IBP				0 to +70	3.5 to 24
P80C528IFP				–40 to +85	
P80C528EBA	PLCC44	plastic leaded chip carrier; 44 leads	SOT187-2	0 to +70	3.5 to 16
P80C528EFA				–40 to +85	
P80C528IBA				0 to +70	3.5 to 24
P80C528IFA				–40 to +85	
P80C528EBB	QFP44	plastic quad flat package; 44 leads (lead length 1.3 mm); body 10 × 10 × 1.75 mm	SOT307-2	0 to +70	3.5 to 16
P80C528EFB				–40 to +85	
P80C528IBB				0 to +70	3.5 to 24
P80C528IFB				–40 to +85	
ROM					
P83C524EBP	DIP40	plastic dual in-line package; 40 leads (600 mil)	SOT129-1	0 to +70	3.5 to 16
P83C524EFP				–40 to +85	
P83C524IBP				0 to +70	3.5 to 24
P83C524IFP				–40 to +85	
P83C524EBA	PLCC44	plastic leaded chip carrier; 44 leads	SOT187-2	0 to +70	3.5 to 16
P83C524EFA				–40 to +85	
P83C524IBA				0 to +70	3.5 to 24
P83C524IFA				–40 to +85	
P83C524EBB	QFP44	plastic quad flat package; 44 leads (lead length 1.3 mm); body 10 × 10 × 1.75 mm	SOT307-2	0 to +70	3.5 to 16
P83C524EFB				–40 to +85	
P83C524IBB				0 to +70	3.5 to 24
P83C524IFB				–40 to +85	

EXTENDED TYPE NUMBER	PACKAGE			TEMPERATURE RANGE (°C)	FREQ. (MHZ)
	NAME	DESCRIPTION	VERSION		
P83C528EBP	DIP40	plastic dual in-line package; 40 leads (600 mil)	SOT129-1	0 to +70	3.5 to 16
P83C528EFP				–40 to +85	
P83C528IBP				0 to +70	3.5 to 24
P83C528IFP				–40 to +85	
P83C528EBA	PLCC44	plastic leaded chip carrier; 44 leads	SOT187-2	0 to +70	3.5 to 16
P83C528EFA				–40 to +85	
P83C528IBA				0 to +70	3.5 to 24
P83C528IFA				–40 to +85	
P83C528EBB	QFP44	plastic quad flat package; 44 leads (lead length 1.3 mm); body 10 × 10 × 1.75 mm	SOT307-2	0 to +70	3.5 to 16
P83C528EFB				–40 to +85	
P83C528IBB				0 to +70	3.5 to 24
P83C528IFB				–40 to +85	

5 BLOCK DIAGRAM

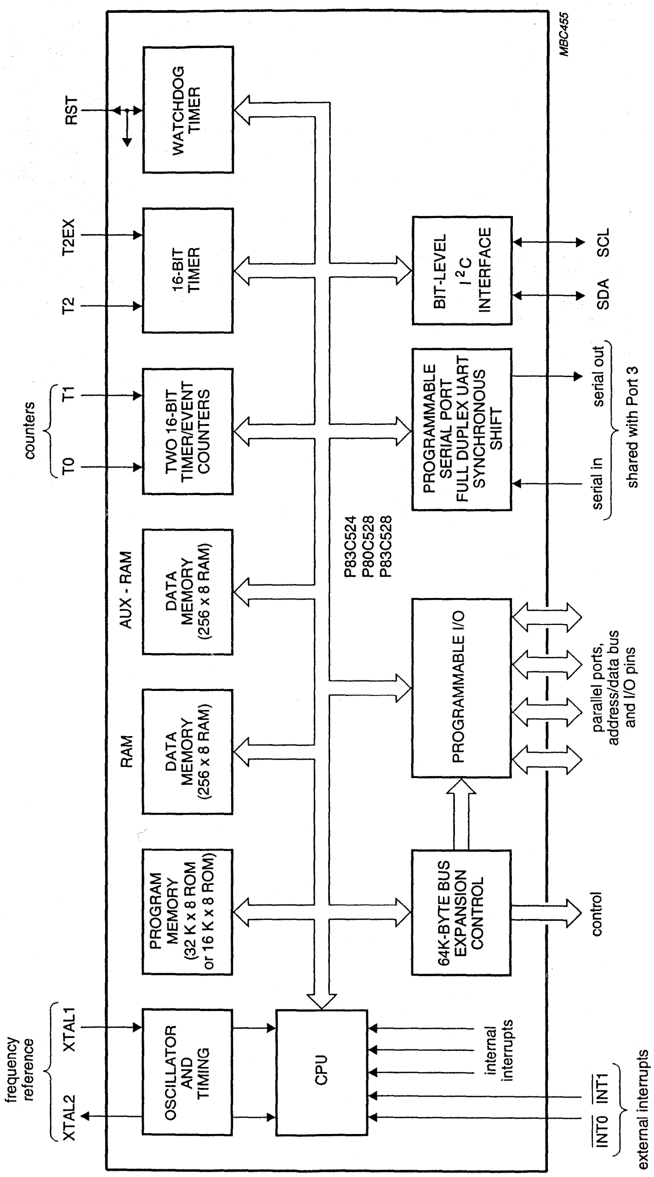


Fig.1 Block diagram.

CMOS single-chip 8-bit microcontroller with A/D and watchdog timer

80C550/83C550/87C550

DESCRIPTION

The Philips 8XC550 is a high-performance microcontroller fabricated with Philips high-density CMOS technology. This Philips CMOS technology combines the high speed and density characteristics of HMOS with the low power attributes of CMOS. Philips epitaxial substrate minimizes latch-up sensitivity. The CMOS 8XC550 has the same instruction set as the 80C51.

The 8XC550 contains a 4k × 8 EPROM (87C550)/ROM (83C550)/ROMless (80C550 has no program memory on-chip), a 128 × 8 RAM, 8 channels of 8-bit A/D, four 8-bit ports (port 1 is input only), a watchdog timer, two 16-bit counter/timers, a seven-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 an on-chip oscillator and clock circuits.

In addition, the 8XC550 has two software selectable modes of power reduction—idle mode and power-down mode. 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.

FEATURES

- 80C51 based architecture
  - 4k × 8 EPROM (87C550)/ROM (83C550)
  - 128 × 8 RAM
  - Eight channels of 8-bit A/D
  - Two 16-bit counter/timers
  - Watchdog timer
  - Full duplex serial channel
  - Boolean processor
- Memory addressing capability
  - 64k ROM and 64k RAM
- Power control modes:
  - Idle mode
  - Power-down mode
- CMOS and TTL compatible
- One speed range at V_{CC} = 5V ±10%
  - 3.5 to 16MHz
- Four package styles
- Extended temperature ranges
- OTP package available

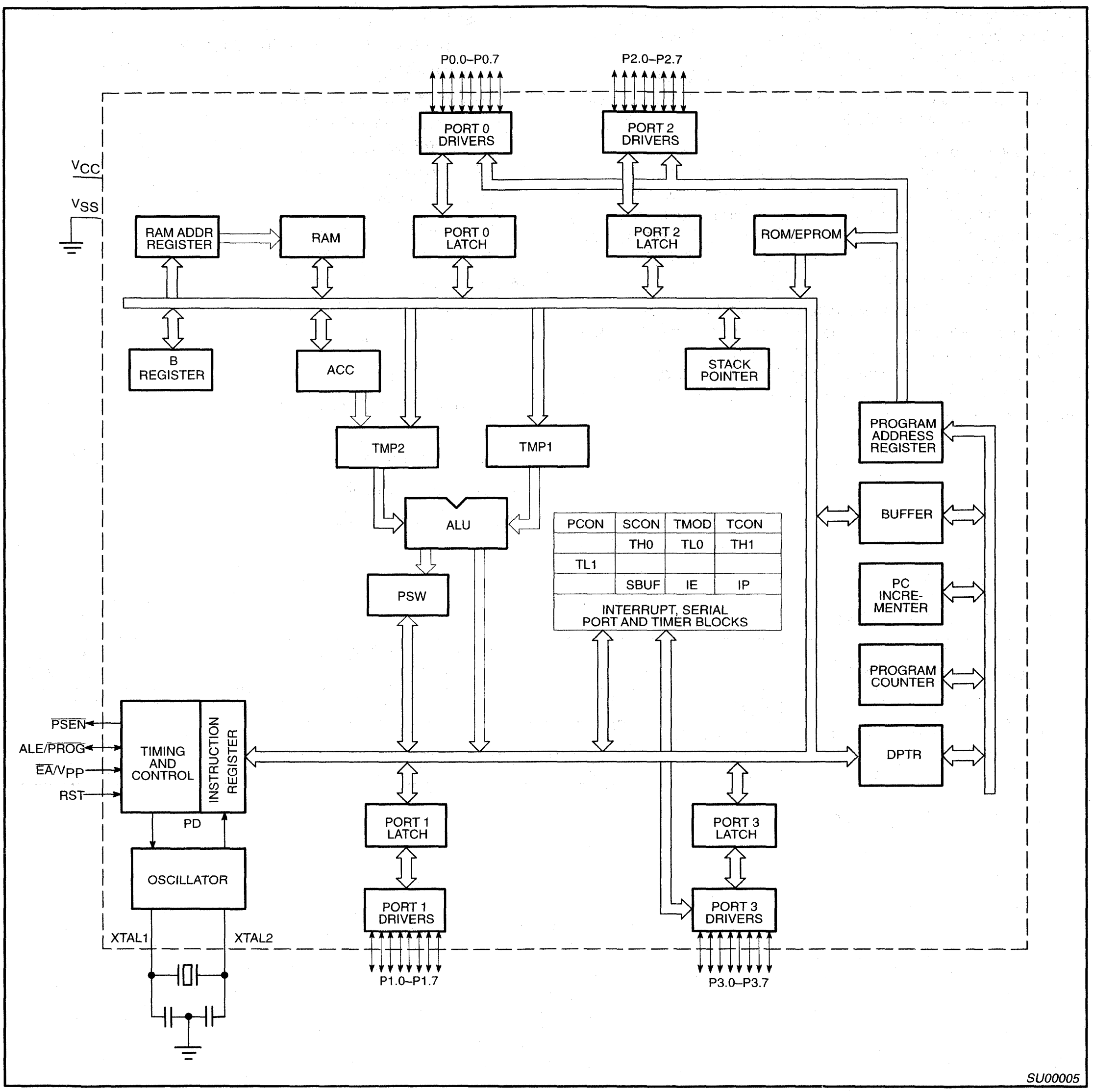
ORDERING INFORMATION

ROMless	ROM	EPROM		TEMPERATURE RANGE °C AND PACKAGE [1]	FREQ MHz	DRAWING NUMBER
P80C550EBP N	P83C550EBP N	P87C550EBP N	OTP	0 to +70, Plastic Dual In-Line Package	3.5 to 16	SOT129-1
P80C550EBA A	P83C550EBA A	P87C550EBA A	OTP	0 to +70, Plastic Leaded Chip Carrier	3.5 to 16	SOT187-2
P80C550EFA A	P83C550EFA A	P87C550EFA A	OTP	–40 to +85, Plastic Leaded Chip Carrier	3.5 to 16	SOT187-2

NOTES:

1. OTP = One Time Programmable EPROM.

BLOCK DIAGRAM



SU00005

Single-chip 8-bit microcontroller with 10-bit A/D, capture/compare timer, high-speed outputs, PWM

DESCRIPTION

The 80C552/83C552 (hereafter generically referred to as 8XC552) Single-Chip 8-Bit Microcontroller is manufactured in an advanced CMOS process and is a derivative of the 80C51 microcontroller family. The 8XC552 has the same instruction set as the 80C51. Three versions of the derivative exist:

- 83C552—8k bytes mask programmable ROM
- 80C552—ROMless version of the 83C552
- 87C552—8k bytes EPROM (described in a separate chapter)

The 8XC552 contains a non-volatile 8k × 8 read-only program memory (83C552), a volatile 256 × 8 read/write data memory, five 8-bit I/O ports, one 8-bit input port, 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 15-source, two-priority-level, nested interrupt structure, an 8-input ADC, a dual DAC pulse width modulated interface, two serial interfaces (UART and I^{2}C-bus), a "watchdog" timer and on-chip oscillator and timing circuits. For systems that require extra capability, the 8XC552 can be expanded using standard TTL compatible memories and logic.

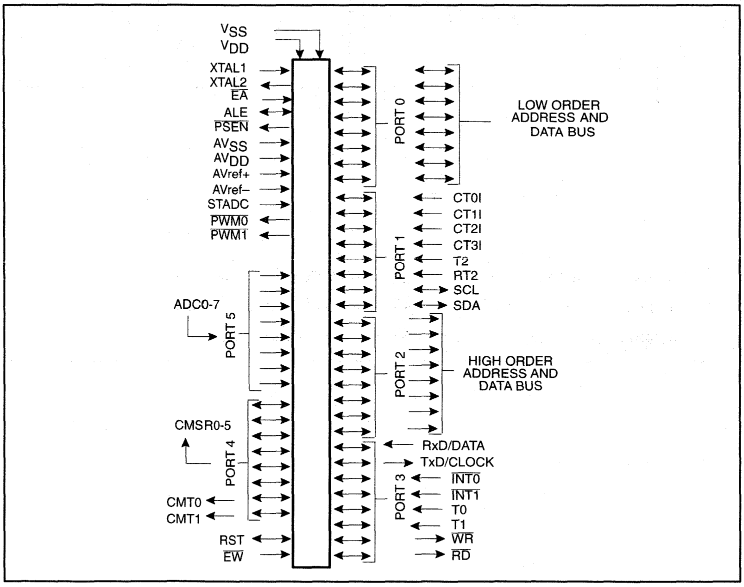
In addition, the 8XC552 has two software selectable modes of power reduction—idle mode and power-down mode. The idle mode freezes the CPU while allowing the RAM, timers, serial ports, and interrupt system to continue functioning. The power-down mode saves the RAM contents but freezes the oscillator, causing all other chip functions to be inoperative.

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: 49 one-byte, 45 two-byte, and 17 three-byte. With a 16MHz (24MHz) crystal, 58% of the instructions are executed in 0.75µs (0.5µs) and 40% in 1.5µs (1µs). Multiply and divide instructions require 3µs (2µs).

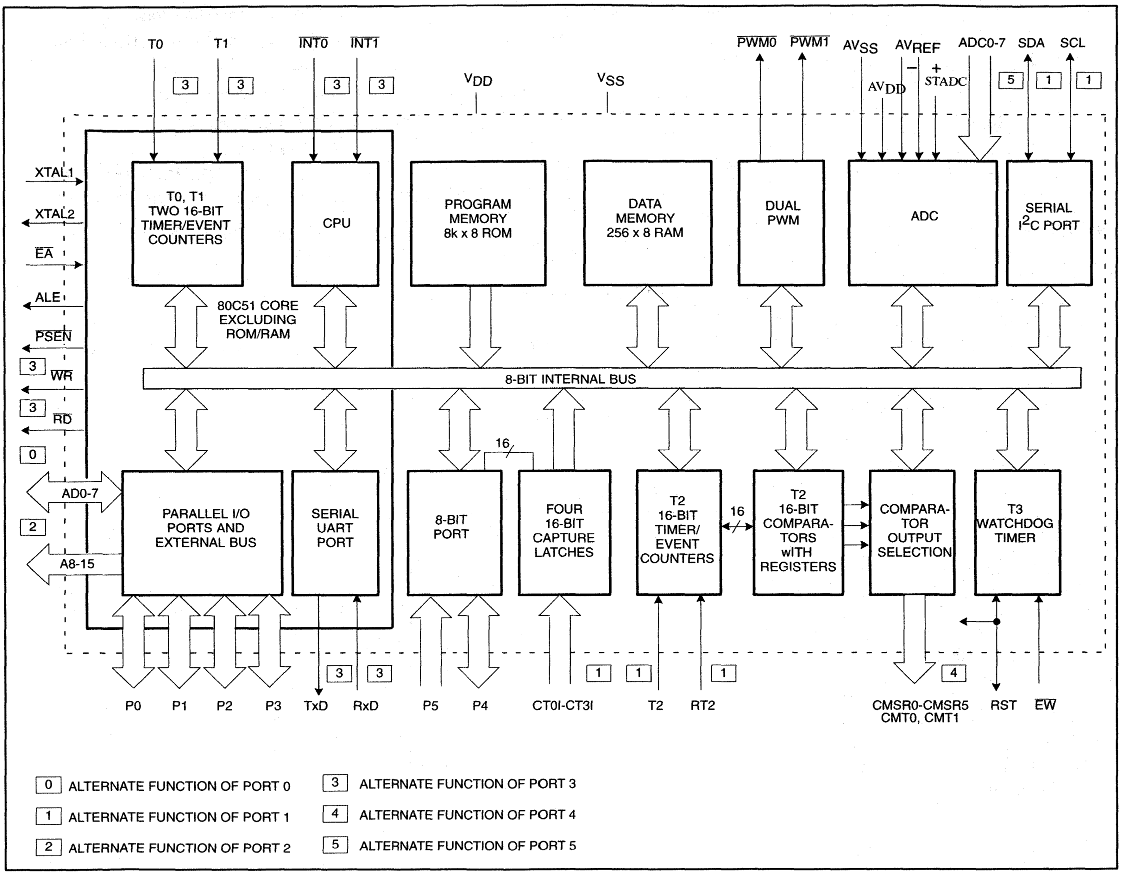
FEATURES

- 80C51 central processing unit
- 8k × 8 ROM expandable externally to 64k bytes
- An additional 16-bit timer/counter coupled to four capture registers and three compare registers
- Two standard 16-bit timer/counters
- 256 × 8 RAM, expandable externally to 64k bytes
- Capable of producing eight synchronized, timed outputs
- A 10-bit ADC with eight multiplexed analog inputs
- Two 8-bit resolution, pulse width modulation outputs
- Five 8-bit I/O ports plus one 8-bit input port shared with analog inputs
- I^{2}C-bus serial I/O port with byte oriented master and slave functions
- Full-duplex UART compatible with the standard 80C51
- On-chip watchdog timer
- Three speed ranges:
  - 3.5 to 16MHz
  - 3.5 to 24MHz (ROM, ROMless only)
  - 3.5 to 30MHz (ROM, ROMless only)
- Three operating ambient temperature ranges:
  - PCB83C552–5: 0°C to +70°C
  - PCF83C552–5: –40°C to +85°C (XTAL frequency max. 24 MHz)
  - PCA83C552–5: –40°C to +125°C (XTAL frequency max. 16 MHz)

LOGIC SYMBOL



BLOCK DIAGRAM



ORDERING INFORMATION

PHILIPS PART ORDER NUMBER PART MARKING		NORTH AMERICA PHILIPS PART ORDER NUMBER			DRAWING NUMBER	TEMPERATURE (°C) AND PACKAGE	FREQ (MHz)
ROMless	ROM[1]	ROMless	ROM	EPROM[2]			
PCB80C552-5-16WP	PCB83C552-5WP/xxx	S80C552-4A68	S83C552-4A68	S87C552-4A68	SOT188-3	0 to +70, Plastic Leaded Chip Carrier	16
PCB80C552-5-16H	PCB83C552-5H/xxx	S80C552-4B	S83C552-4B	S87C552-4BA	SOT318-2	0 to +70, Plastic Quad Flat Pack	16
PCF80C552-5-16WP	PCF83C552-5WP/xxx	S80C552-5A68	S83C552-5A68	S87C552-5A68	SOT188-3	–40 to +85, Plastic Leaded Chip Carrier	16
PCF80C552-5-16H	PCF83C552-5H/xxx	S80C552-5B	S83C552-5B		SOT318-2	–40 to +85, Plastic Quad Flat Pack	16
PCA80C552-5-16WP	PCA83C552-5WP/xxx	S80C552-6A68	S83C552-6A68		SOT188-3	–40 to +125, Plastic Leaded Chip Carrier	16
PCA80C552-5-16H	PCA83C552-5H/xxx	S80C552-6B	S83C552-6B		SOT318-2	–40 to +125, Plastic Quad Flat Pack	16
PCB80C552-5-24WP	PCB83C552-5WP/xxx	S80C552-AA68	S83C552-AA68		SOT188-3	0 to +70, Plastic Leaded Chip Carrier	24
PCB80C552-5-24H	PCB83C552-5H/xxx	S80C552-AB	S83C552-AB		SOT318-2	0 to +70, Plastic Quad Flat Pack	24
PCF80C552-5-24WP	PCF83C552-5WP/xxx	S80C552-BA68	S83C552-BA68		SOT188-3	–40 to +85, Plastic Leaded Chip Carrier	24
PCF80C552-5-24H	PCF83C552-5H/xxx	S80C552-BB	S83C552-BB		SOT318-2	–40 to +85, Plastic Quad Flat Pack	24
PCB80C552-5-30WP	PCB83C552-5WP/xxx	S80C552-CA68	S83C552-CA68		SOT188-3	0 to +70, Plastic Leaded Chip Carrier	30
PCB80C552-5-30H	PCB83C552-5H/xxx	S80C552-CB	S83C552-CB		SOT318-2	0 to +70, Plastic Quad Flat Pack	30

NOTE:

1. xxx denotes the ROM code number.
2. For EPROM device specification, refer to *87C552* datasheet.

Single-chip 8-bit microcontroller 87C552

Single-chip 8-bit microcontroller with 10-bit A/D, capture/compare timer, high-speed outputs, PWM

DESCRIPTION

The 87C552 Single-Chip 8-Bit Microcontroller is manufactured in an advanced CMOS process and is a derivative of the 80C51 microcontroller family. The 87C552 has the same instruction set as the 80C51. Three versions of the derivative exist:

- 83C552—8k bytes mask programmable ROM
- 80C552—ROMless version of the 83C552
- 87C552—8k bytes EPROM

The 87C552 contains a 8k × 8 a volatile 256 × 8 read/write data memory, five 8-bit I/O ports, one 8-bit input port, 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 15-source, two-priority-level, nested interrupt structure, an 8-input ADC, a dual DAC pulse width modulated interface, two serial interfaces (UART and I^{2}C-bus), a "watchdog" timer and on-chip oscillator and timing circuits. For systems that require extra capability, the 87C552 can be expanded using standard TTL compatible memories and logic.

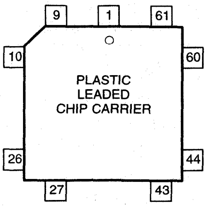
In addition, the 87C552 has two software selectable modes of power reduction—idle mode and power-down mode. The idle mode freezes the CPU while allowing the RAM, timers, serial ports, and interrupt system to continue functioning. The power-down mode saves the RAM contents but freezes the oscillator, causing all other chip functions to be inoperative.

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: 49 one-byte, 45 two-byte, and 17 three-byte. With a 16MHz crystal, 58% of the instructions are executed in 0.75µs and 40% in 1.5µs. Multiply and divide instructions require 3µs.

FEATURES

- 80C51 central processing unit
- 8k × 8 EPROM expandable externally to 64k bytes
- An additional 16-bit timer/counter coupled to four capture registers and three compare registers
- Two standard 16-bit timer/counters
- 256 × 8 RAM, expandable externally to 64k bytes
- Capable of producing eight synchronized, timed outputs
- A 10-bit ADC with eight multiplexed analog inputs
- Two 8-bit resolution, pulse width modulation outputs
- Five 8-bit I/O ports plus one 8-bit input port shared with analog inputs
- I^{2}C-bus serial I/O port with byte oriented master and slave functions
- Full-duplex UART compatible with the standard 80C51
- On-chip watchdog timer
- 16MHz speed
- Extended temperature ranges
- OTP package available

PIN CONFIGURATIONS



Pin	Function	Pin	Function
1	P5.0/ADC0	35	XTAL1
2	V_{DD}	36	V_{SS}
3	STADC	37	V_{SS}
4	$\overline{PWM0}$	38	NC
5	$\overline{PWM1}$	39	P2.0/A08
6	$\overline{EW}$	40	P2.1/A09
7	P4.0/CMSR0	41	P2.2/A10
8	P4.1/CMSR1	42	P2.3/A11
9	P4.2/CMSR2	43	P2.4/A12
10	P4.3/CMSR3	44	P2.5/A13
11	P4.4/CMSR4	45	P2.6/A14
12	P4.5/CMSR5	46	P2.7/A15
13	P4.6/CMT0	47	$\overline{PSEN}$
14	P4.7/CMT1	48	ALE/$\overline{PROG}$
15	RST	49	$\overline{EA}$/V_{PP}
16	P1.0/CT0I	50	P0.7/AD7
17	P1.1/CT1I	51	P0.6/AD6
18	P1.2/CT2I	52	P0.5/AD5
19	P1.3/CT3I	53	P0.4/AD4
20	P1.4/T2	54	P0.3/AD3
21	P1.5/RT2	55	P0.2/AD2
22	P1.6/SCL	56	P0.1/AD1
23	P1.7/SDA	57	P0.0/AD0
24	P3.0/RxD	58	AVref–
25	P3.1/TxD	59	AVref+
26	P3.2/$\overline{INT0}$	60	AV_{SS}
27	P3.3/$\overline{INT1}$	61	AV_{DD}
28	P3.4/T0	62	P5.7/ADC7
29	P3.5/T1	63	P5.6/ADC6
30	P3.6/$\overline{WR}$	64	P5.5/ADC5
31	P3.7/$\overline{RD}$	65	P5.4/ADC4
32	NC	66	P5.3/ADC3
33	NC	67	P5.2/ADC2
34	XTAL2	68	P5.1/ADC1

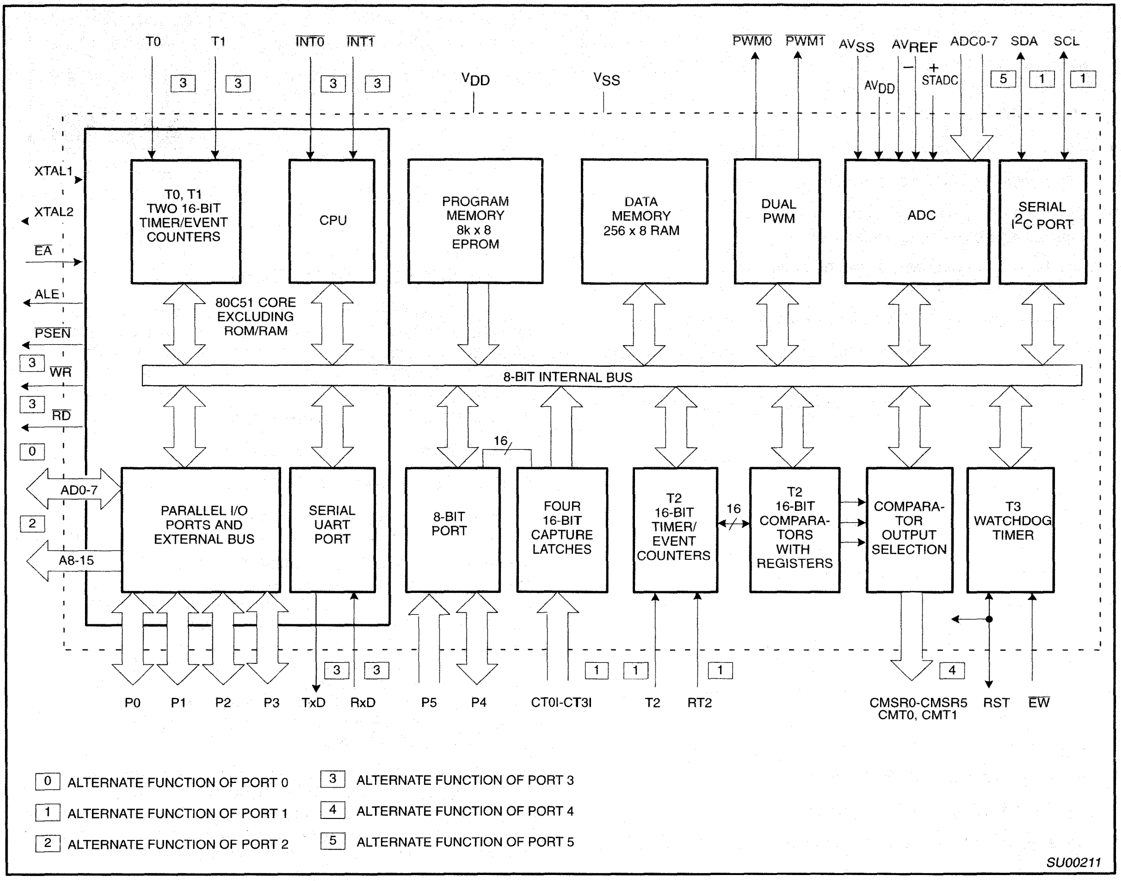
SU00208

EPROM	DRAWING NUMBER	TEMPERATURE °C AND PACKAGE	FREQ MHz
S87C552-4A68	SOT188-3	0 to +70, Plastic Leaded Chip Carrier	16
S87C552-4BA	SOT318-2	0 to +70, Plastic Quad Flat Pack	16
S87C552-5A68	SOT188-3	–40 to +85, Plastic Leaded Chip Carrier	16

NOTES:

1. For ROM and ROMless see datasheet 80C552/83C552

BLOCK DIAGRAM


T0
T1
INT0
INT1
3
VDD
VSS
PWM0
PWM1
AVSS
AVREF
ADC0-7
SDA
SCL
AVDD
–
+
STADC
5
1
XTAL1
XTAL2
EA
ALE
PSEN
WR
RD
0
2
AD0-7
A8-15
T0, T1 TWO 16-BIT TIMER/EVENT COUNTERS
CPU
PROGRAM MEMORY 8k x 8 EPROM
DATA MEMORY 256 x 8 RAM
DUAL PWM
ADC
SERIAL I2C PORT
80C51 CORE EXCLUDING ROM/RAM
8-BIT INTERNAL BUS
16
PARALLEL I/O PORTS AND EXTERNAL BUS
SERIAL UART PORT
8-BIT PORT
FOUR 16-BIT CAPTURE LATCHES
T2 16-BIT TIMER/ EVENT COUNTERS
T2 16-BIT COMPARA-TORS WITH REGISTERS
COMPARA-TOR OUTPUT SELECTION
T3 WATCHDOG TIMER
4
P0
P1
P2
P3
TxD
RxD
P5
P4
CT0I-CT3I
T2
RT2
CMSR0-CMSR5
CMT0, CMT1
RST
EW
0 ALTERNATE FUNCTION OF PORT 0
1 ALTERNATE FUNCTION OF PORT 1
2 ALTERNATE FUNCTION OF PORT 2
3 ALTERNATE FUNCTION OF PORT 3
4 ALTERNATE FUNCTION OF PORT 4
5 ALTERNATE FUNCTION OF PORT 5
SU00211

1. FEATURES

- 80C51 central processing unit
- 32 K × 8 ROM respectively FEEPROM (Flash-EEPROM), expandable externally to 64 Kbytes
- ROM/FEEPROM Code protection
- 1024 × 8 RAM, expandable externally to 64 Kbytes
- Two standard 16-bit timer/counters
- An additional 16-bit timer/counter coupled to four capture registers and three compare registers
- A 10-bit ADC with eight multiplexed analog inputs and programmable autoscan
- Two 8-bit resolution, pulse width modulation outputs
- Five 8-bit I/O ports plus one 8-bit input port shared with analog inputs
- I^{2}C-bus serial I/O port with byte oriented master and slave functions
- Full-duplex UART compatible with the standard 80C51
- On-chip watchdog timer
- 15 interrupt sources with 2 priority levels (2 to 6 external sources possible)
- Extended temperature range (–40 to +85°C)
- 4.5 to 5.5 V supply voltage range
- Frequency range for 80C51-family standard oscillator: 3.5 MHz to 16 MHz
- PLL oscillator with 32 kHz reference and software-selectable system clock frequency
- Seconds Timer
- Software enable/disable of ALE output pulse
- Electromagnetic compatibility improvements
- Wake-up from Power-down by external or seconds interrupt

2. GENERAL DESCRIPTION

The P80CE558/P83CE558/P89CE558 (hereafter generically referred to as P8xCE558) single-chip 8-bit microcontroller is manufactured in an advanced CMOS process and is a derivative of the 80C51 microcontroller family. The P8xCE558 has the same instruction set as the 80C51. Three versions of the derivative exist:

- P83CE558 — 32 Kbytes mask programmable ROM
- P80CE558 — ROMless version of the P83CE558
- P89CE558 — 32 Kbytes FEEPROM (Flash-EEPROM)

The P8xCE558 contains a non-volatile 32 Kbytes mask programmable ROM (P83CE558) or electrically erasable FEEPROM respectively (P89CE558), a volatile 1024 × 8 read/write data memory, five 8-bit I/O ports, one 8-bit input port, 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 15-source, two-priority-level, nested interrupt structure, an 8-input ADC, a dual DAC pulse width modulated interface, two serial interfaces (UART and I^{2}C-bus), a "watchdog" timer, an on-chip oscillator and timing circuits. For systems that require extra capability the P8xCE558 can be expanded using standard TTL compatible memories and logic.

In addition, the P8xCE558 has two software selectable modes of power reduction — Idle Mode and power-down mode. 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.

The device also functions as an arithmetic processor having facilities for both binary and BCD arithmetic as well as bit-handling capabilities. The instruction set consists of over 100 instructions: 49 one-byte, 45 two-byte, and 17 three- byte. With a 16 MHz system clock, 58% of the instructions are executed in 0.75 µs and 40% in 1.5 µs. Multiply and divide instructions require 3 µs.

3. ORDERING INFORMATION

EXTENDED TYPE NUMBER	PACKAGE			FREQUENCY RANGE (MHz)	TEMPERATURE RANGE (°C)
	NAME	DESCRIPTION	CODE		
ROMless					
P80CE558EBB	QFP80	Plastic Quad Flat Pack; 80 leads	SOT318-1	3.5 to 16	0 to +70
P80CE558EFB	QFP80	Plastic Quad Flat Pack; 80 leads	SOT318-1	3.5 to 16	–40 to +85
ROM coded					
P83CE558EBB/YYY[1]	QFP80	Plastic Quad Flat Pack; 80 leads	SOT318-1	3.5 to 16	0 to +70
P83CE558EFB/YYY[1]	QFP80	Plastic Quad Flat Pack; 80 leads	SOT318-1	3.5 to 16	–40 to +85
FEEPROM					
P89CE558EBB	QFP80	Plastic Quad Flat Pack; 80 leads	SOT318-1	3.5 to 16	0 to +70
P89CE558EFB	QFP80	Plastic Quad Flat Pack; 80 leads	SOT318-1	3.5 to 16	–40 to +85

NOTE:

1. YYY denotes the ROM code number

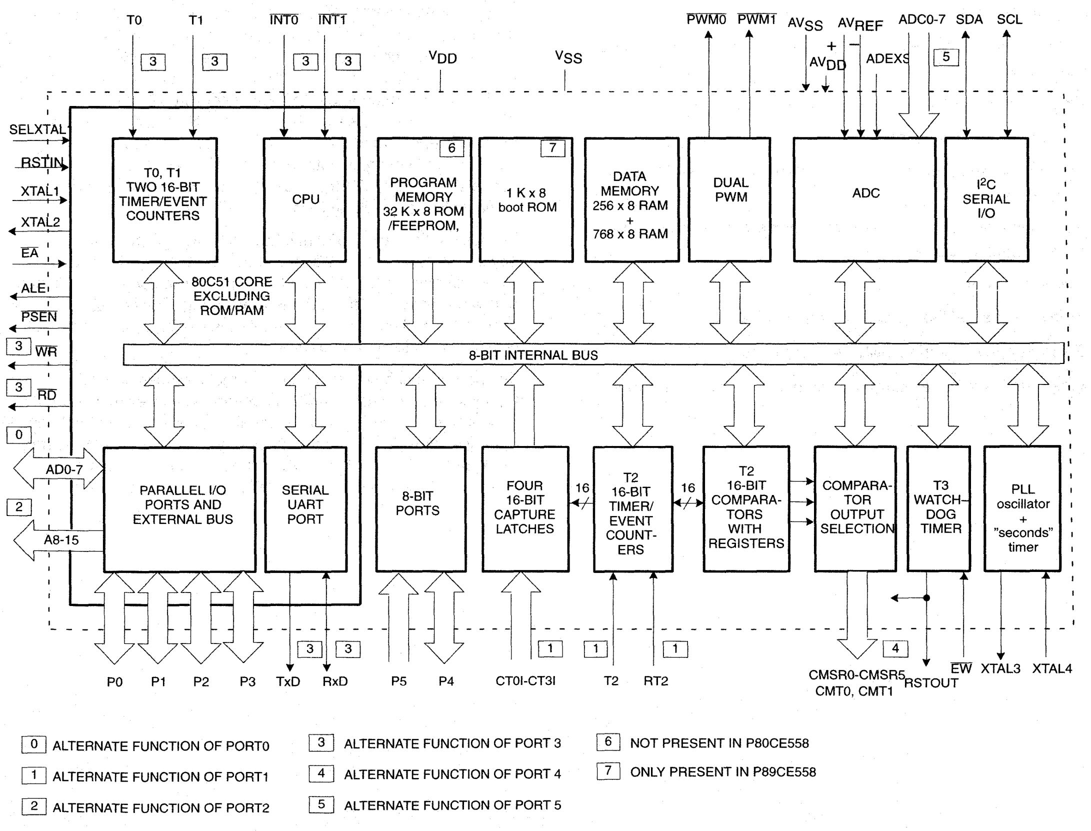

T0
T1
INT0
INT1
3
VDD
VSS
PWM0
PWM1
AVSS
AVREF
ADC0-7
SDA
SCL
+
–
AVDD
ADEXS
5
SELXTAL1
RSTIN
XTAL1
XTAL2
EA
ALE
PSEN
WR
RD
0
2
AD0-7
A8-15
T0, T1 TWO 16-BIT TIMER/EVENT COUNTERS
CPU
6
PROGRAM MEMORY 32 K x 8 ROM /FEEPROM,
7
1 K x 8 boot ROM
DATA MEMORY 256 x 8 RAM + 768 x 8 RAM
DUAL PWM
ADC
I²C SERIAL I/O
80C51 CORE EXCLUDING ROM/RAM
8-BIT INTERNAL BUS
PARALLEL I/O PORTS AND EXTERNAL BUS
SERIAL UART PORT
8-BIT PORTS
FOUR 16-BIT CAPTURE LATCHES
16
T2 16-BIT TIMER/ EVENT COUNT-ERS
T2 16-BIT COMPARA-TORS WITH REGISTERS
COMPARA-TOR OUTPUT SELECTION
T3 WATCH-DOG TIMER
PLL oscillator + "seconds" timer
1
4
P0
P1
P2
P3
TxD
RxD
P5
P4
CT0I-CT3I
T2
RT2
CMSR0-CMSR5 CMT0, CMT1
RSTOUT
EW
XTAL3
XTAL4
0 ALTERNATE FUNCTION OF PORT0
1 ALTERNATE FUNCTION OF PORT1
2 ALTERNATE FUNCTION OF PORT2
3 ALTERNATE FUNCTION OF PORT 3
4 ALTERNATE FUNCTION OF PORT 4
5 ALTERNATE FUNCTION OF PORT 5
6 NOT PRESENT IN P80CE558
7 ONLY PRESENT IN P89CE558

Figure 1. Block diagram P8xCE558

1. FEATURES

- 80C51 central processing unit
- 48 K × 8 ROM, expandable externally to 64 Kbytes
- ROM Code protection
- 1536 × 8 RAM, expandable externally to 64 Kbytes
- Two standard 16-bit timer/counters
- An additional 16-bit timer/counter coupled to four capture registers and three compare registers
- A 10-bit ADC with eight multiplexed analog inputs and programmable autoscan
- Two 8-bit resolution, pulse width modulation outputs
- Five 8-bit I/O ports plus one 8-bit input port shared with analog inputs
- I^2C-bus serial I/O port with byte oriented master and slave functions
- Full-duplex UART compatible with the standard 80C51
- On-chip watchdog timer
- 15 interrupt sources with 2 priority levels (2 to 6 external sources possible)
- Extended temperature range (–40 to +85 °C)
- 4.5 to 5.5 V supply voltage range
- Frequency range for 80C51–family standard oscillator: 3.5 MHz to 16 MHz
- PLL oscillator with 32 kHz reference and software–selectable system clock frequency
- Seconds Timer
- Software enable/disable of ALE output pulse
- Electromagnetic compatibility improvements
- Wake–up from Power–down by external or seconds interrupt

2. GENERAL DESCRIPTION

The P80CE559/P83CE559 (hereafter generically referred to as P8xCE559) single-chip 8-bit microcontroller is manufactured in an advanced CMOS process and is a derivative of the 80C51 microcontroller family. The P8xCE559 has the same instruction set as the 80C51. Three versions of the derivative exist:

- P83CE559 — 48 Kbytes mask programmable ROM
- P80CE559 — ROMless version of the P83CE559
- P89CE559 — not planned any longer

The P8xCE559 contains a non-volatile 48 Kbytes mask programmable ROM (P83CE559), a volatile 1536 × 8 read/write data memory, five 8-bit I/O ports, one 8-bit input port, 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 15-source, two-priority-level, nested interrupt structure, an 8-input ADC, a dual DAC pulse width modulated interface, two serial interfaces (UART and I^2C-bus), a “watchdog” timer, an on-chip oscillator and timing circuits. For systems that require extra capability the P8xCE559 can be expanded using standard TTL compatible memories and logic.

In addition, the P8xCE559 has two software selectable modes of power reduction — Idle Mode and power-down mode. 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.

The device also functions as an arithmetic processor having facilities for both binary and BCD arithmetic as well as bit-handling capabilities. The instruction set consists of over 100 instructions: 49 one-byte, 45 two-byte, and 17 three- byte. With a 16 MHz system clock, 58% of the instructions are executed in 0.75 μs and 40% in 1.5 μs. Multiply and divide instructions require 3 μs.

3. ORDERING INFORMATION

EXTENDED TYPE NUMBER	PACKAGE			FREQUENCY RANGE (MHz)	TEMPERATURE RANGE (°C)
	NAME	DESCRIPTION	CODE		
ROMless					
P80CE559EBB	QFP80	Plastic Quad Flat Pack; 80 leads	SOT318-1	3.5 to 16	0 to +70
P80CE559EFB	QFP80	Plastic Quad Flat Pack; 80 leads	SOT318-1	3.5 to 16	–40 to +85
ROM coded					
P83CE559EBB/YYY[1]	QFP80	Plastic Quad Flat Pack; 80 leads	SOT318-1	3.5 to 16	0 to +70
P83CE559EFB/YYY[1]	QFP80	Plastic Quad Flat Pack; 80 leads	SOT318-1	3.5 to 16	–40 to +85

NOTE:

1. YYY denotes the ROM code number.

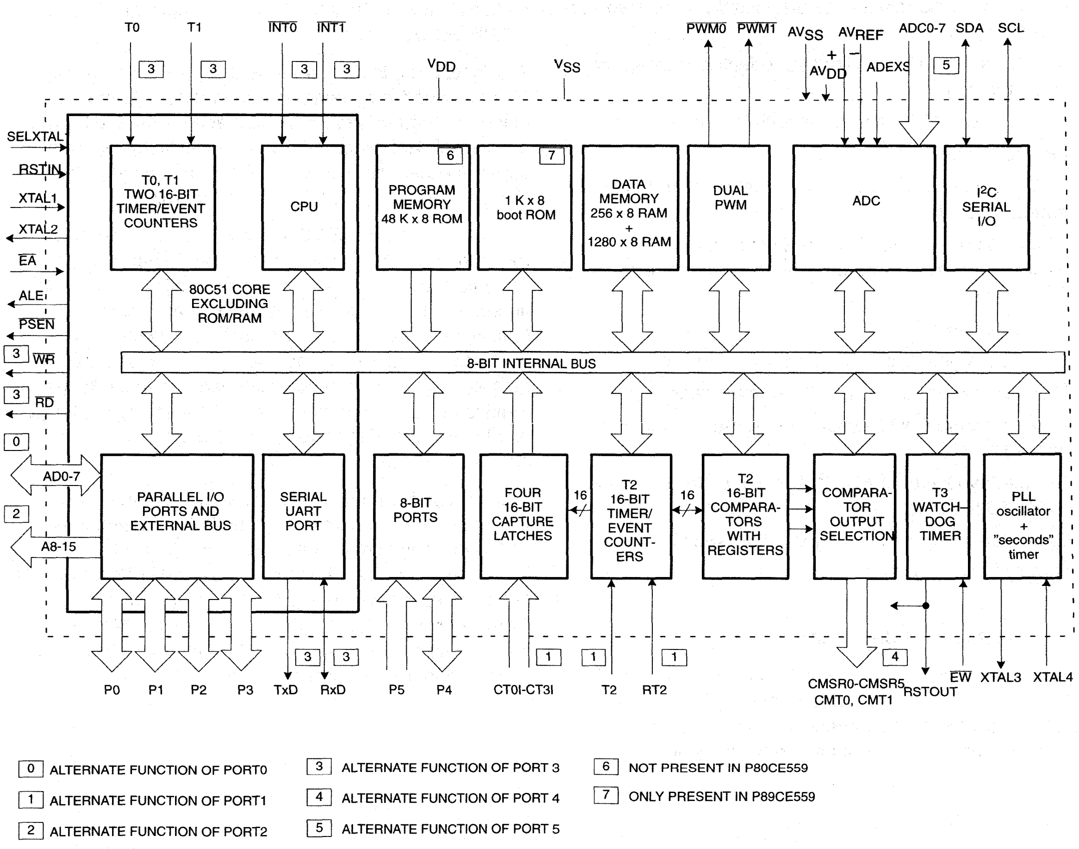

T0
T1
$\overline{INT0}$
$\overline{INT1}$
3
V_{DD}
V_{SS}
$\overline{PWM0}$
$\overline{PWM1}$
AV_{SS}
AV_{REF}
+
−
AV_{DD}
ADEXS
ADC0-7
SDA
SCL
5
SELXTAL
$\overline{RSTIN}$
XTAL1
XTAL2
$\overline{EA}$
ALE
$\overline{PSEN}$
$\overline{WR}$
$\overline{RD}$
0
AD0-7
2
A8-15
T0, T1 TWO 16-BIT TIMER/EVENT COUNTERS
CPU
6
PROGRAM MEMORY 48 K x 8 ROM
7
1 K x 8 boot ROM
DATA MEMORY 256 x 8 RAM + 1280 x 8 RAM
DUAL PWM
ADC
I²C SERIAL I/O
80C51 CORE EXCLUDING ROM/RAM
8-BIT INTERNAL BUS
PARALLEL I/O PORTS AND EXTERNAL BUS
SERIAL UART PORT
8-BIT PORTS
FOUR 16-BIT CAPTURE LATCHES
16
T2 16-BIT TIMER/ EVENT COUNT-ERS
T2 16-BIT COMPARA-TORS WITH REGISTERS
COMPARA-TOR OUTPUT SELECTION
T3 WATCH-DOG TIMER
PLL oscillator + "seconds" timer
1
4
P0
P1
P2
P3
TxD
RxD
P5
P4
CT0I-CT3I
T2
RT2
CMSR0-CMSR5
CMT0, CMT1
RSTOUT
$\overline{EW}$
XTAL3
XTAL4
0 ALTERNATE FUNCTION OF PORT0
1 ALTERNATE FUNCTION OF PORT1
2 ALTERNATE FUNCTION OF PORT2
3 ALTERNATE FUNCTION OF PORT 3
4 ALTERNATE FUNCTION OF PORT 4
5 ALTERNATE FUNCTION OF PORT 5
6 NOT PRESENT IN P80CE559
7 ONLY PRESENT IN P89CE559

Figure 1. Block diagram P8xCE559.

1 FEATURES

- 80C51 Central Processing Unit (CPU)
- 64 kbytes ROM (only P83CE560)
- 64 kbytes EPROM (only P87CE560)
- ROM/EPROM Code protection
- 2048 bytes RAM, expandable externally to 64 kbytes
- Two standard 16-bit timers/counters
- An additional 16-bit timer/counter coupled to four capture registers and three compare registers
- A 10-bit Analog-to-Digital Converter (ADC) with eight multiplexed analog inputs and programmable autoscan
- Two 8-bit resolution, Pulse Width Modulation outputs
- Five 8-bit I/O ports plus one 8-bit input port shared with analog inputs
- I^{2}C-bus serial I/O port with byte oriented master and slave functions
- Full-duplex UART compatible with the standard 80C51
- On-chip Watchdog Timer
- 15 interrupt sources with 2 priority levels (2 to 6 external sources possible)
- Phase-Locked Loop (PLL) oscillator with 32 kHz reference and software-selectable system clock frequency
- Seconds timer
- Software enable/disable of ALE output pulse
- Electromagnetic compatibility improvements
- Wake-up from Power-down by external or seconds interrupt
- Frequency range for 80C51-family standard oscillator: 3.5 to 16 MHz
- Extended temperature range: –40 to +85 C
- Supply voltage: 4.5 to 5.5 V.

2 GENERAL DESCRIPTION

The 8-bit microcontrollers P80CE560, P83CE560 and P87CE560 - hereafter referred to as P8xCE560 - are manufactured in an advanced CMOS process and are derivatives of the 80C51 microcontroller family.

The P8xCE560 contains a volatile 2048 bytes read/write Data Memory, five 8-bit I/O ports, one 8-bit input port, 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 15-source, two-priority-level, nested interrupt structure, an 8-input ADC, a dual Digital-to-Analog Convertor (DAC), Pulse Width Modulated interface, two serial interfaces (UART and I^{2}C-bus), a Watchdog Timer, an on-chip oscillator and timing circuits.

The P8xCE560 is available in 3 versions:

- P80CE560: ROMless version
- P83CE560: containing a non-volatile 64 kbytes mask programmable ROM
- P87CE560: containing 64 kbytes programmable EPROM/OTP.

The P8xCE560 is a control-oriented CPU with on-chip Program and Data Memory; it cannot be extended with external Program Memory. It can access up to 64 kbytes of external Data Memory. For systems requiring extra capability, the P8xCE560 can be expanded using standard TTL compatible memories and peripherals.

In addition, the P8xCE560 has two software selectable reduced power modes: Idle mode and Power-down mode. 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.The Power-down mode can be terminated by an external reset, by the seconds interrupt and by any one of the two external interrupts; see Section 15.3.

The device also functions as an arithmetic processor having facilities for both binary and BCD arithmetic as well as bit-handling capabilities. The instruction set of the P8xCE560 is the same as the 80C51 and consists of over 100 instructions: 49 one-byte, 45 two-byte, and 17 three-byte. With a 16 MHz system clock, 58% of the instructions are executed in 0.75 μs and 40% in 1.5 μs. Multiply and divide instructions require 3 μs.

2.1 Electromagnetic Compatibility (EMC)

Primary attention is paid to the reduction of electromagnetic emission of the microcontroller P8xCE560. The following features reduce the electromagnetic emission and additionally improve the electromagnetic susceptibility:

- Four digital part supply voltage pins (V_{DD1} to V_{DD4}) and four digital ground pins (V_{SS1} to V_{SS4}) are placed as pairs of V_{DDn} and V_{SSn} at two adjacent pins, at each side of the package.
- Separated V_{DD} pins for the internal logic and the port buffers.
- Internal decoupling capacitance improves the EMC radiation behaviour and the EMC immunity.
- External capacitors should be connected across associated V_{DDn} and V_{SSn} pins (i.e. V_{DD1} and V_{SS1}). Lead length should be as short as possible. Ceramic chip capacitors are recommended (100 nF).

2.2 Recommendation on ALE

For applications that require no external memory or temporarily no external memory: the ALE output signal (pulses at a frequency of $\frac{1}{6} \times f_{OSC}$) can be disabled under software control (bit RFI; SFR: PCON.5); if disabled, no ALE pulse will occur. ALE pin will be pulled down internally, switching an external address latch to a quiet state. The MOVX instruction will still toggle ALE (external Data Memory is accessed). ALE will retain its normal HIGH value during Idle mode and a LOW value during Power-down mode while in the '**RFI** reduction mode'.

Additionally during internal access ($\overline{EA}$ = 1) ALE will toggle normally when the address exceeds the internal Program Memory size. During external access ($\overline{EA}$ = 0) ALE will always toggle normally, whether the flag 'RFI' is set or not.

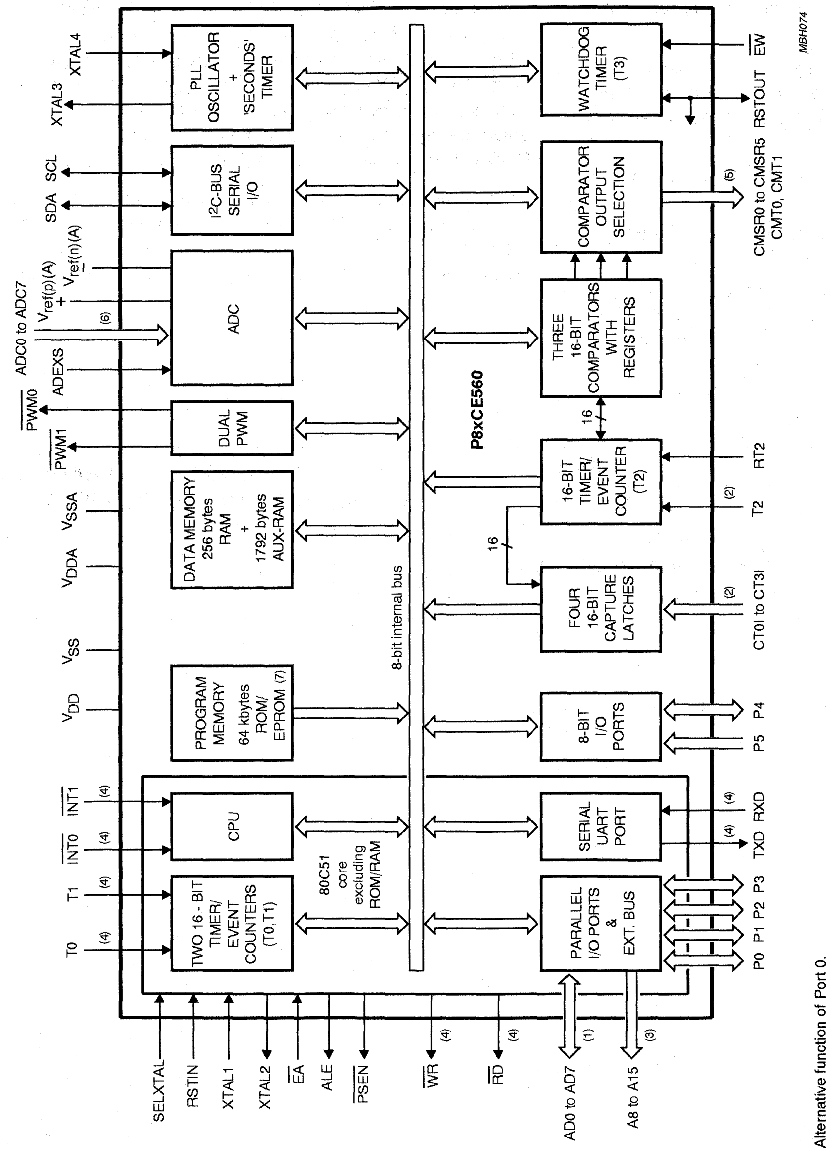
3 ORDERING INFORMATION

TYPE NUMBER	PACKAGE			FREQUENCY RANGE (MHZ)	TEMPERATURE RANGE (°C)
	NAME	DESCRIPTION	VERSION		
P80CE560EFB[1]	QFP80	plastic quad flat package; 80 leads (lead length 1.95 mm); body 14 × 20 × 2.8 mm	SOT318-2	3.5 to 16	−40 to +85
P83CE560EFB/nnn[2]					
P87CE560EFB[3]					

Notes

1. ROMless type.
2. ROM coded type; 'nnn' denotes the ROM code number.
3. EPROM/OTP type.

4 BLOCK DIAGRAM



(1) Alternative function of Port 0.
(2) Alternative function of Port 1.
(3) Alternative function of Port 2.
(4) Alternative function of Port 3.
(5) Alternative function of Port 5.
(6) Alternative function of Port 6.
(7) Not present in P80CE560.

Fig.1 Block diagram P8xCE560.

8-bit microcontroller P83C562; P80C562

1 FEATURES

- 80C51 Central Processing Unit
- 8 kbytes ROM, expandable externally to 64 kbytes
- 256 bytes RAM, expandable externally to 64 kbytes
- Two standard 16-bit timer/counters
- An additional 16-bit timer/counter coupled to four capture registers and three compare registers
- An 8-bit ADC with 8 multiplexed analog inputs
- Two 8-bit resolution, Pulse Width Modulated outputs
- Five 8-bit I/O ports plus one 8-bit input port shared with analog inputs
- Full-duplex UART compatible with the standard 80C51
- On-chip Watchdog Timer
- Oscillator frequency: 3.5 to 16 MHz.

2 GENERAL DESCRIPTION

The P80C562/P83C562 (hereafter generally referred to as P8xC562) single-chip 8-bit microcontroller is manufactured in an advanced CMOS process and is a derivative of the 80C51 microcontroller family.
The P8xC562 has the same instruction set as the 80C51.
Two versions of the derivative exist:

- With 8 kbytes mask-programmable ROM
- ROMless version of the P8xC562.

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 P8xC562 contains a non-volatile 8 kbyte read only program memory, a volatile 256 byte 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 fourteen-source, two-priority-level, nested interrupt structure, an 8-input ADC, a dual DAC with pulse width modulated outputs, a serial interface (UART), a Watchdog Timer and on-chip oscillator and timing circuits. For systems that require extra capability, the P8xC562 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: 49 one-byte, 45 two-byte and 17 three-byte. With a 16 MHz crystal, 58% of the instructions are executed in 0.75 μs and 40% in 1.5 μs. Multiply and divide instructions require 3 μs.

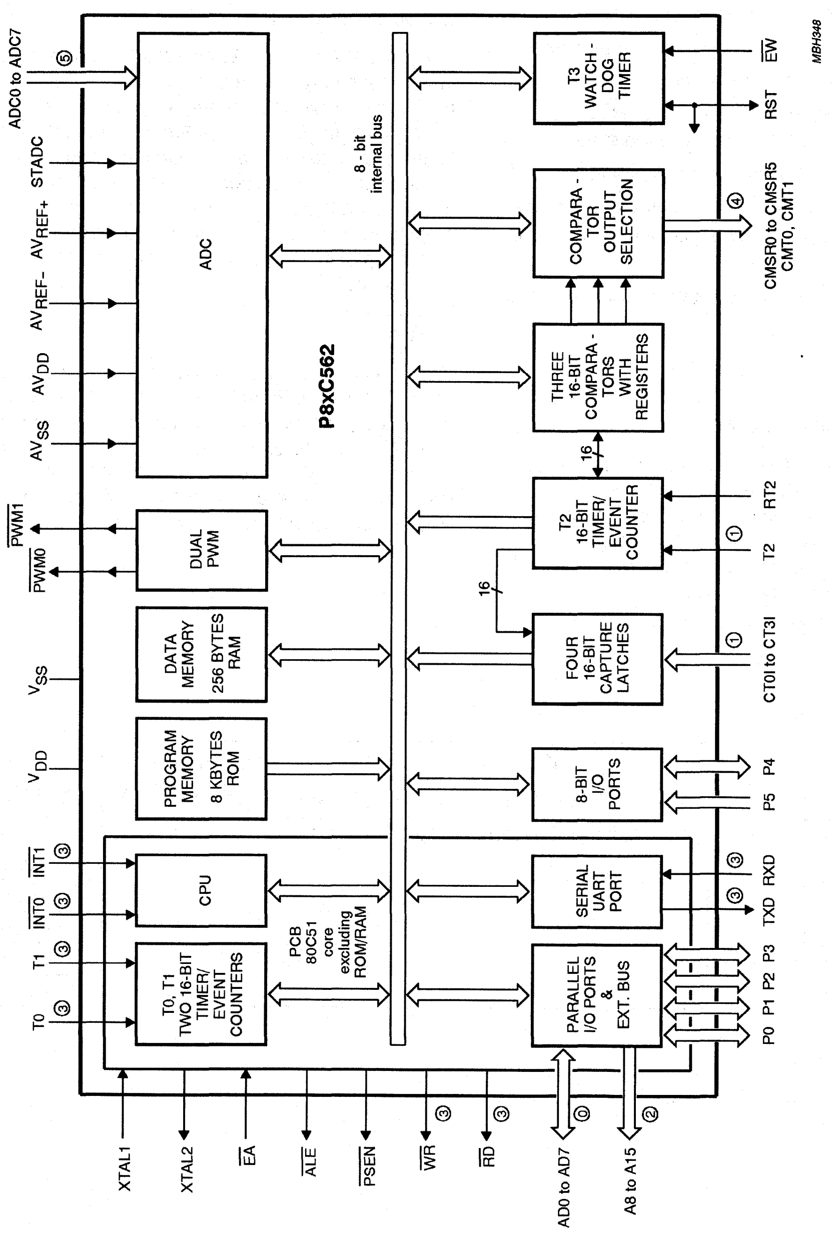
3 ORDERING INFORMATION

TYPE NUMBER	PACKAGE			FREQUENCY RANGE (MHz)	TEMPERATURE RANGE (°C)
	NAME	DESCRIPTION	VERSION		
P80CE562EHA[1]	PLCC68	plastic leaded chip carrier; 68 leads	SOT188-2	3.5 to 16	−40 to +125
P80C562EBA[1]					0 to +70
P80C562EFA[1]					−40 to +85
P83C562EHA/nnn[2]					−40 to +125
P83C562EBA/nnn[2]					0 to +70
P83C562EFA/nnn[2]					−40 to +85

Notes

1. ROMless type.
2. ROM coded type; nnn denotes the ROM code number.

4 BLOCK DIAGRAM


ADC0 to ADC7
⑤
STADC
AVREF+
AVREF−
AVDD
AVSS
ADC
PWM1
PWM0
DUAL PWM
VSS
VDD
DATA MEMORY 256 BYTES RAM
PROGRAM MEMORY 8 KBYTES ROM
INT1 ③
INT0 ③
T1 ③
T0 ③
CPU
T0, T1 TWO 16-BIT TIMER/ EVENT COUNTERS
PCB 80C51 core excluding ROM/RAM
8 - bit internal bus
P8xC562
XTAL1
XTAL2
EA
ALE
PSEN
WR ③
RD ③
AD0 to AD7 ⓪
A8 to A15 ②
T3 WATCH - DOG TIMER
EW
RST
COMPARA - TOR OUTPUT SELECTION
④
CMSR0 to CMSR5 CMT0, CMT1
THREE 16-BIT COMPARA - TORS WITH REGISTERS
16
T2 16-BIT TIMER/ EVENT COUNTER
RT2
T2 ①
16
FOUR 16-BIT CAPTURE LATCHES
CT0I to CT3I ①
8-BIT I/O PORTS
P4
P5
SERIAL UART PORT
RXD ③
TXD ③
PARALLEL I/O PORTS & EXT. BUS
P0 P1 P2 P3
MBH348

⓪ alternative function of port 0
① alternative functions 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

Fig.1 Block diagram.

CMOS single-chip 8-bit microcontrollers 80C575/83C575/87C575

DESCRIPTION

The Philips 80C575/83C575/87C575 is a high-performance microcontroller fabricated with Philips high-density CMOS technology. The Philips CMOS technology combines the high speed and density characteristics of HMOS with the low power attributes of CMOS. Philips epitaxial substrate minimizes latch-up sensitivity.

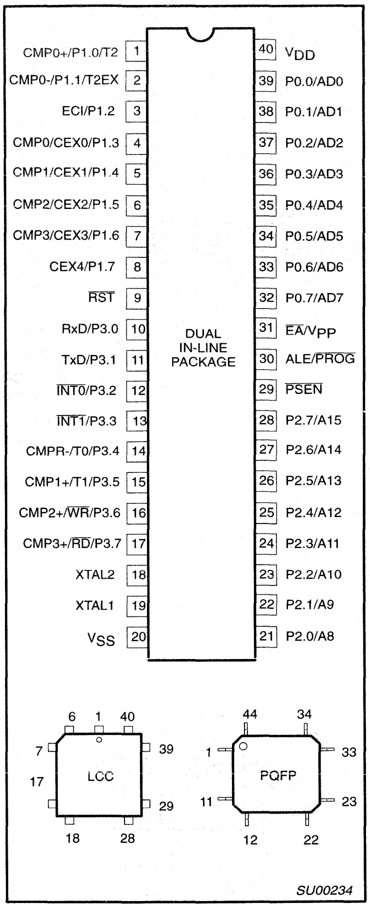
The 8XC575 contains an 8k × 8 ROM (83C575) EPROM (87C575), a 256 × 8 RAM, 32 I/O lines, three 16-bit counter/timers, a Programmable Counter Array (PCA), a seven-source, two-priority level nested interrupt structure, an enhanced UART, four analog comparators, power-fail detect and oscillator fail detect circuits, and on-chip oscillator and clock circuits.

In addition, the 8XC575 has a low active reset, and the port pins are reset to a low level. There is also a fully configurable watchdog timer, and internal power on clear circuit. The part includes idle mode and power-down mode states for reduced power consumption.

FEATURES

- 80C51 based architecture
  - 8k × 8 ROM (83C575)
  - 8k × 8 EPROM (87C575)
  - ROMless (80C575)
  - 256 × 8 RAM
  - Three 16-bit counter/timers
  - Programmable Counter Array
  - Enhanced UART
  - Boolean processor
  - Oscillator fail detect
  - Low active reset
  - Asynchronous low port reset
  - Schmitt trigger inputs
  - 4 analog comparators
  - Watchdog timer
  - Low V_{CC} detect
- Memory addressing capability
  - 64k ROM and 64k RAM
- Power control modes:
  - Idle mode
  - Power-down mode
- CMOS and TTL compatible
- 4.0 to 16MHz
- Extended temperature ranges
- OTP package available

PIN CONFIGURATIONS

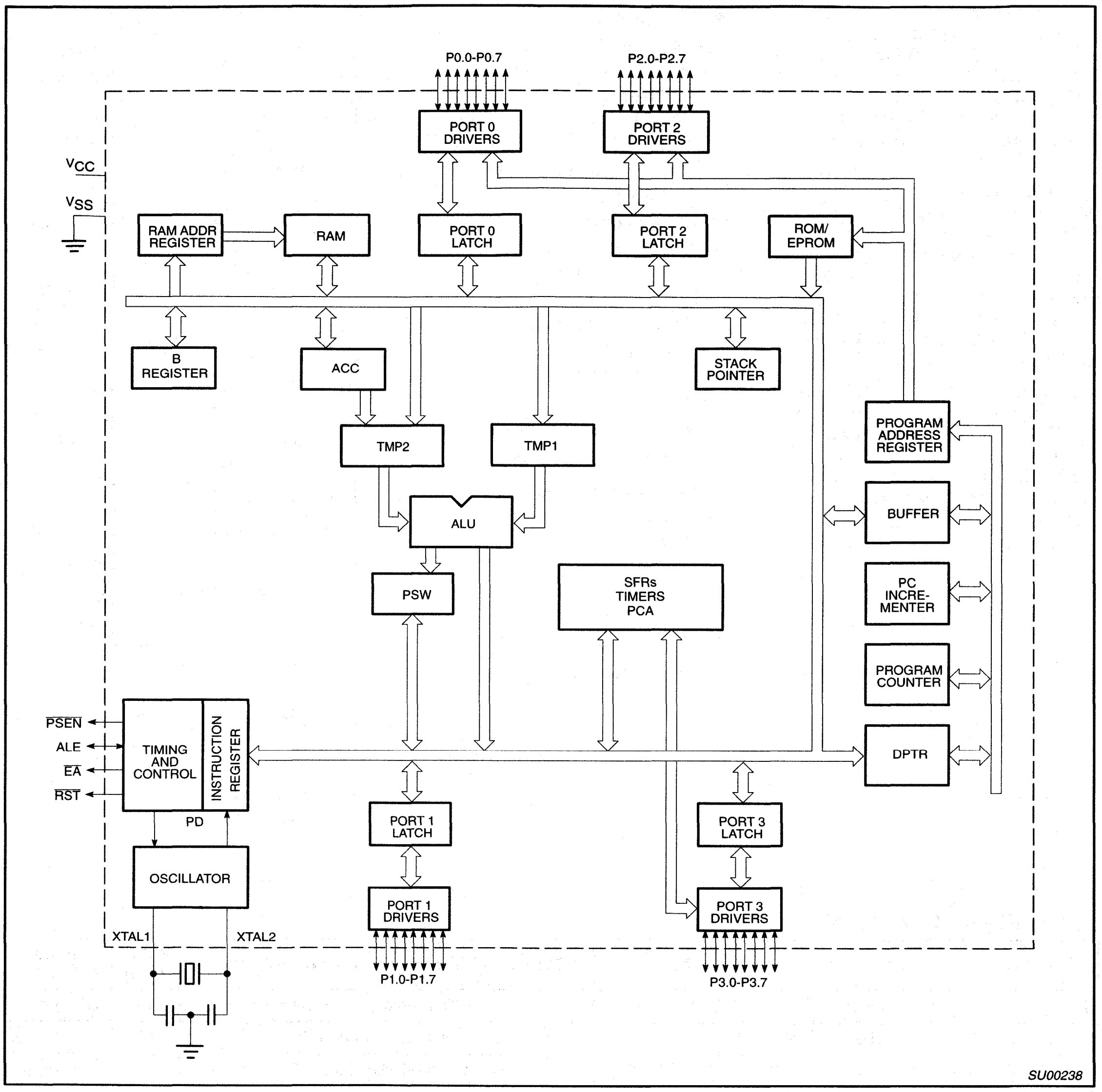


ORDERING INFORMATION

ROMless	ROM	EPROM[1]		TEMPERATURE RANGE °C AND PACKAGE	FREQ (MHz)	DRAWING NUMBER
P80C575EBPN	P83C575EBPN	P87C575EBPN	OTP	0 to +70, 40-Pin Plastic Dual In-line Package	16	SOT129-1
P80C575EBAA	P83C575EBAA	P87C575EBAA	OTP	0 to +70, 44-Pin Plastic Leaded Chip Carrier	16	SOT187-2
P80C575EHAA	P83C575EHAA	P87C575EHAA	OTP	–40 to +125, 44-Pin Plastic Leaded Chip Carrier	16	SOT187-2
P80C575EBBB	P83C575EBBB	P87C575EBBB	OTP	0 to +70, 44-Pin Plastic Quad Flat Pack	16	SOT307-2

NOTE:
1. OTP - One Time Programmable EPROM. UV - Erasable EPROM

BLOCK DIAGRAM



FEATURES

- 80C51 based architecture
  - 8k × 8 ROM (83C576)
  - 8k × 8 EPROM (87C576)
  - 256 × 8 RAM
  - 10-bit, 6 channel A/D
  - Three 16-bit counter/timers
  - 2 PWM outputs
  - Programmable Counter Array
  - Universal Peripheral Interface
  - Enhanced UART
  - Oscillator fail detect
  - Low active reset
  - 4 analog comparators
  - Watchdog timer
  - Low V_{CC} detect
  - Power-on detect
- Memory addressing capability
  - 64k ROM and 64k RAM
- Power control modes:
  - Idle mode
  - Power-down mode
- CMOS and TTL compatible
- 6 to 16MHz
- Extended temperature ranges
- OTP versions available
- That can be programmed in circuit
- Software Reset
- 15 source, 2 level interrupt structure
- Lower EMI noise
- Programmable I/O pins
- Serial on-board programming
- Schmitt trigger inputs on Port 1

DESCRIPTION

The Philips 83C576/87C576 is a high-performance microcontroller fabricated with Philips high-density CMOS technology. The Philips CMOS technology combines the high speed and density characteristics of HMOS with the low power attributes of CMOS. Philips epitaxial substrate minimizes latch-up sensitivity.

The 8XC576 contains an 8k × 8 ROM (83C576) EPROM (87C576), a 256 × 8 RAM, 32 I/O lines, three 16-bit counter/timers, a Programmable Counter Array (PCA), a 10-bit, 6 channel A/D, 2 PWM outputs, an 8-bit UPI interface, a fifteen-source, two-priority level nested interrupt structure, an enhanced UART, four analog comparators, power-fail detect and oscillator fail detect circuits, and on-chip oscillator and clock circuits.

In addition, the 8XC576 has a low active reset, and a software reset. There is also a fully configurable watchdog timer, and internal power on clear circuit. The part includes idle mode and power-down mode states for reduced power consumption.

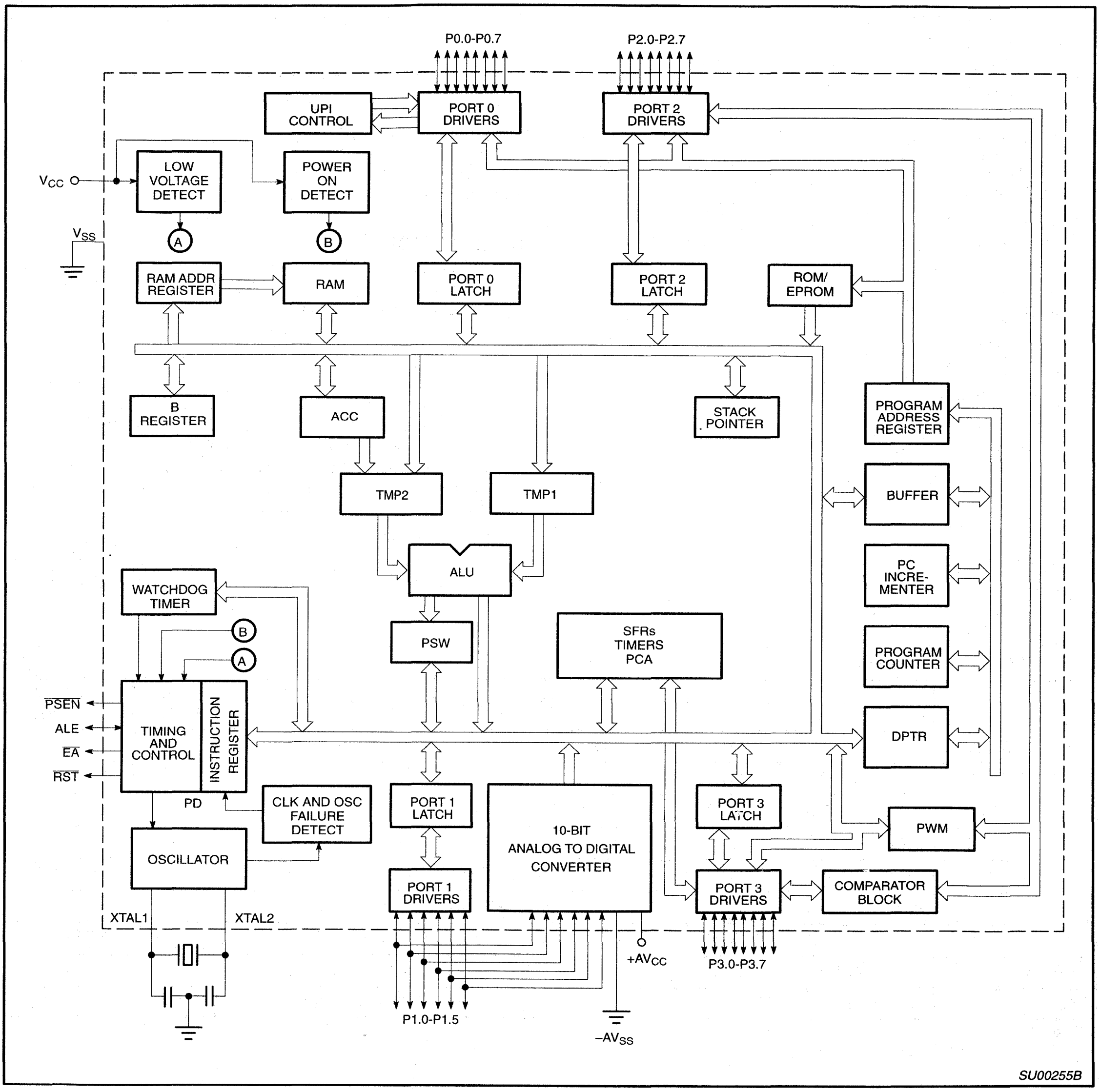
ORDERING INFORMATION

ROM	EPROM[1]		TEMPERATURE RANGE °C AND PACKAGE	FREQ (MHz)	DRAWING NUMBER
P83C576EBPN	P87C576EBPN	OTP	0 to +70, 40-Pin Plastic Dual In-line Package	16	SOT129-1
P83C576EBAA	P87C576EBAA	OTP	0 to +70, 44-Pin Plastic Leaded Chip Carrier	16	SOT187-2
P83C576EBBB	P87C576EBBB	OTP	0 to +70, 44-Pin Plastic Quad Flat Pack	16	SOT307-2
P83C576EFPN	P87C576EBPN	OTP	–40 to +85, 40-Pin Plastic Dual In-line Package	16	SOT129-1
P83C576EFAA	P87C576EFAA	OTP	–40 to +85, 44-Pin Plastic Leaded Chip Carrier	16	SOT187-2
P83C576EFBB	P87C576EFBB	OTP	–40 to +85, 44-Pin Plastic Quad Flat Pack	16	SOT307-2
P83C576EHPN	P87C576EHPN	OTP	–40 to +125, 40-Pin Plastic Dual In-line Package	16	SOT129-1
P83C576EHAA	P87C576EHAA	OTP	–40 to +125, 44-Pin Plastic Leaded Chip Carrier	16	SOT187-2
P83C576EHBB	P87C576EHBB	OTP	–40 to +125, 44-Pin Plastic Quad Flat Pack	16	SOT307-2

NOTE:

1. OTP - One Time Programmable EPROM. UV - Erasable EPROM

BLOCK DIAGRAM



Low voltage 8-bit microcontrollers with UART, I²C-bus and ADC

P80CL580; P83CL580

1 FEATURES

- Full static 80C51 Central Processing Unit
- 8-bit CPU, ROM, RAM, I/O in a 56-lead VSO or 64-lead QFP package
- 256 bytes on-chip RAM Data Memory
- 6 kbytes on-chip ROM Program Memory for P83CL580
- External memory expandable up to 128 kbytes: RAM up to 64 kbytes and ROM up to 64 kbytes
- Five 8-bit ports; 40 I/O lines
- Three 16-bit Timers/Event counters
- On-chip oscillator suitable for RC, LC, quartz crystal or ceramic resonator
- Fifteen source, fifteen vector, nested interrupt structure with two priority levels
- Full duplex serial port (UART)
- I²C-bus interface for serial transfer on two lines
- Analog-to-digital converter (ADC) with Power-down mode; 4 input channels and 8-bit ADC
- Pulse Width Modulated (PWM) output (8-bit resolution)
- Watchdog Timer
- Enhanced architecture with:
  - non-page oriented instructions
  - direct addressing
  - four 8-byte RAM register banks
  - stack depth limited only by available internal RAM (maximum 256 bytes)
  - multiply, divide, subtract and compare instructions
- Reduced power consumption through Power-down and Idle modes
- Wake-up via external interrupts at Port 1
- Frequency range: 0 to 12 MHz. For ADC operation minimum 250 kHz at 2.7 V
- Supply voltage: 2.5 to 6.0 V
- Very low current consumption: typically 4.5 mA at 2.5 V and 8 MHz
- Operating ambient temperature range: −40 to +85 °C.

2 GENERAL DESCRIPTION

The P80CL580; P83CL580 (hereafter generally referred to as P8xCL580) is manufactured in an advanced CMOS technology. The P8xCL580 has the same instruction set as the 80C51, consisting of over 100 instructions:
49 one-byte, 46 two-byte, and 16 three-byte. The device operates over a wide range of supply voltages and has low power consumption; there are two software selectable modes for power reduction: Idle and Power-down.
For emulation purposes, the P85CL580 (piggy-back version) with 256 bytes of RAM is recommended.

This data sheet details the specific properties of the P80CL580; P83CL580. For details of the 80C51 core and the I²C-bus see *"Data Handbook IC20"*.

2.1 ROMless version: P80CL580

The P80CL580 is the ROMless version of the P83CL580. The mask options on the P80CL580 are fixed as follows:

- All ports have option '1S' (standard port, HIGH after reset), except ports P1.6 and P1.7 which have option '2S' (open-drain, HIGH after reset)
- Oscillator option: Oscillator 3
- Power-on-reset option: off.

3 APPLICATIONS

The P8xCL580 is an 8-bit general purpose microcontroller especially suited for cordless telephone and mobile communication applications. The P8xCL580 also functions as an arithmetic processor having facilities for both binary and BCD arithmetic plus bit-handling capabilities.

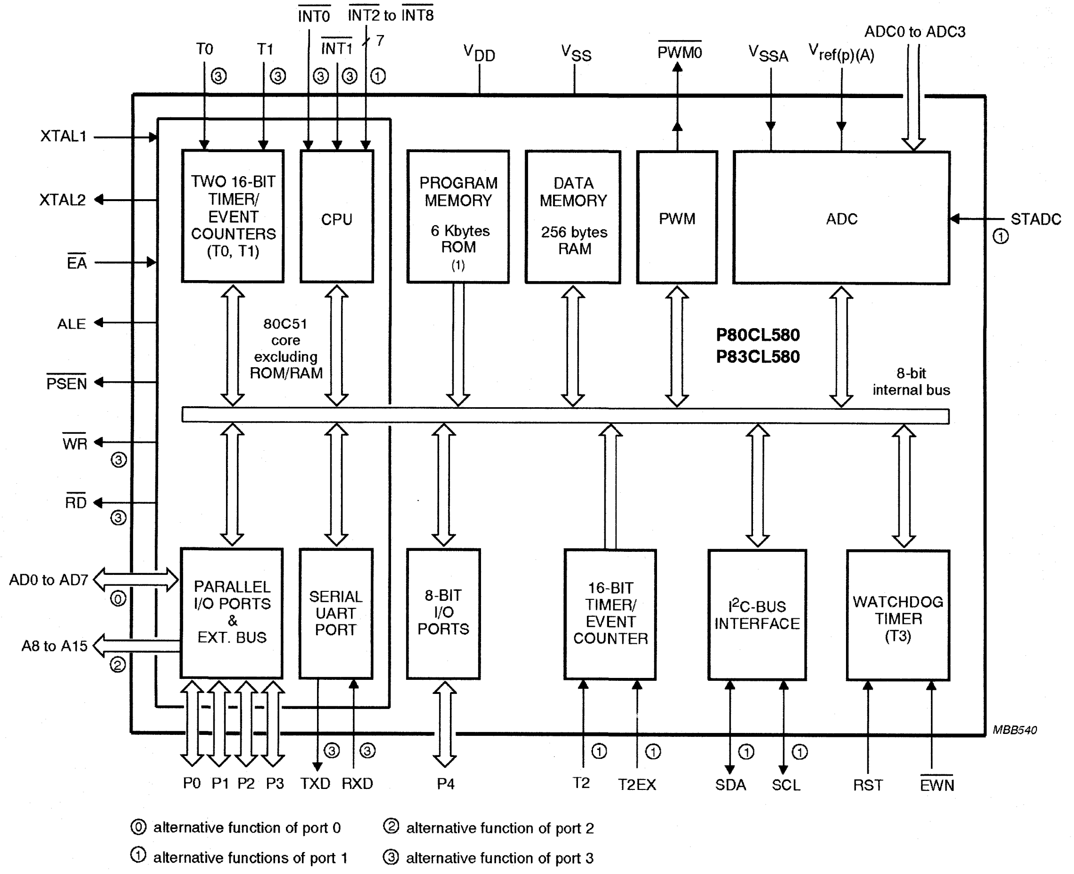
4 ORDERING INFORMATION

TYPE NUMBER[(1)]	PACKAGE		
	NAME	DESCRIPTION	VERSION
P8xCL580HFT	VSO56	plastic very small outline package; 56 leads	SOT190-1
P8xCL580HFH	QFP64	plastic quad flat package; 64 leads (lead length 1.95 mm); body 14 x 20 x 2.8 mm	SOT319-2

Note

1. 'x' = 0 or 3. Refer to the Order Entry Form (OEF) for this device for the full type number, including options/program.

5 BLOCK DIAGRAM


INT0
INT2 to INT8
7
T0
T1
INT1
③
①
V_DD
V_SS
PWM0
V_SSA
V_ref(p)(A)
ADC0 to ADC3
XTAL1
XTAL2
EA
ALE
PSEN
WR
RD
AD0 to AD7
⓪
A8 to A15
②
TWO 16-BIT TIMER/ EVENT COUNTERS (T0, T1)
CPU
PROGRAM MEMORY 6 Kbytes ROM (1)
DATA MEMORY 256 bytes RAM
PWM
ADC
STADC
80C51 core excluding ROM/RAM
P80CL580
P83CL580
8-bit internal bus
PARALLEL I/O PORTS & EXT. BUS
SERIAL UART PORT
8-BIT I/O PORTS
16-BIT TIMER/ EVENT COUNTER
I²C-BUS INTERFACE
WATCHDOG TIMER (T3)
MBB540
P0 P1 P2 P3
TXD RXD
P4
T2 T2EX
SDA SCL
RST EWN
⓪ alternative function of port 0
① alternative functions of port 1
② alternative function of port 2
③ alternative function of port 3

(1) Not available in the **P80CL580**.

Fig.1 Block diagram.

CMOS single-chip 8-bit microcontrollers 80C652/83C652

DESCRIPTION

The P80C652/83C652 Single-Chip 8-Bit Microcontroller is manufactured in an advanced CMOS process and is a derivative of the 80C51 microcontroller family. The 80C652/83C652 has the same instruction set as the 80C51. Three versions of the derivative exist:

83C652 — 8k bytes mask programmable ROM

80C652 — ROMless version

87C652 — EPROM version (described in a separate chapter)

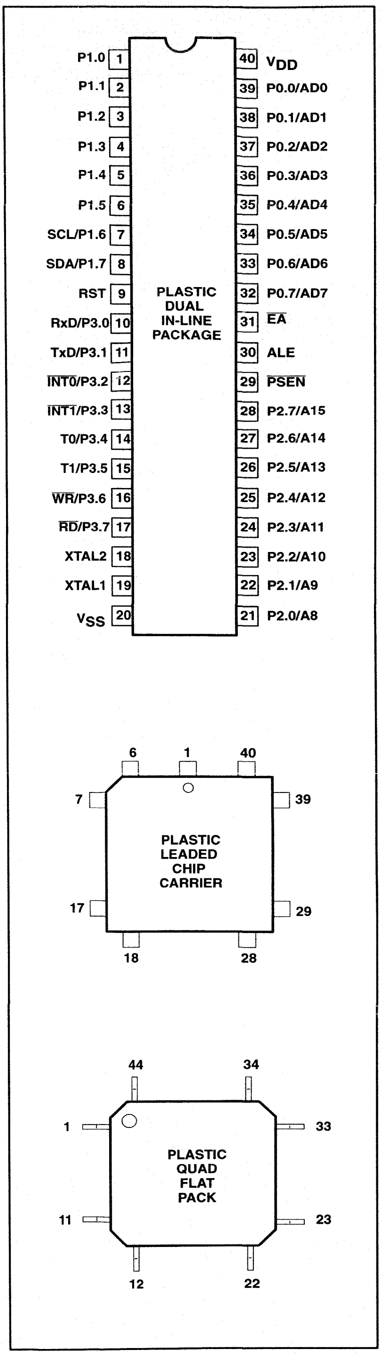
This device provides architectural enhancements that make it applicable in a variety of applications for general control systems. The 8XC652 contains a non-volatile 8k × 8 read-only program memory, a volatile 256 × 8 read/write data memory, four 8-bit I/O ports, two 16-bit timer/event counters (identical to the timers of the 80C51), a multi-source, two-priority-level, nested interrupt structure, an I^{2}C interface, UART and on-chip oscillator and timing circuits. For systems that require extra capability, the 8XC652 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: 49 one-byte, 45 two-byte and 17 three-byte. With a 16(24)MHz crystal, 58% of the instructions are executed in 0.75(0.5)µs and 40% in 1.5(1)µs. Multiply and divide instructions require 3(2)µs.

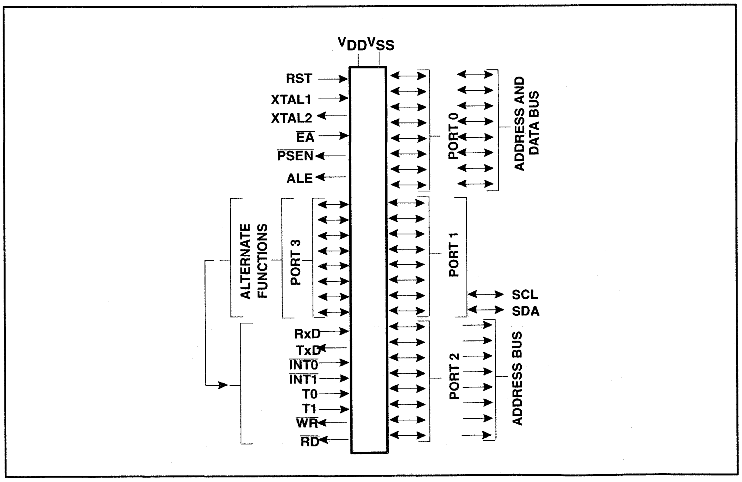
FEATURES

- 80C51 central processing unit
- 8k × 8 ROM expandable externally to 64k bytes
- 256 × 8 RAM, expandable externally to 64k bytes
- Two standard 16-bit timer/counters
- Four 8-bit I/O ports
- I^{2}C-bus serial I/O port with byte oriented master and slave functions
- Full-duplex UART facilities
- Power control modes
  - Idle mode
  - Power-down mode
- ROM code protection
- Extended frequency range: 3.5 to 24 MHz
- Three operating ambient temperature ranges:
  - 0 to +70°C
  - –40 to +85°C
  - –40 to +125°C

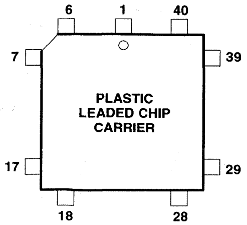
PIN CONFIGURATIONS



LOGIC SYMBOL



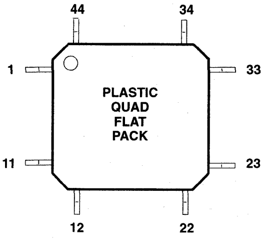
PLASTIC LEADED CHIP CARRIER PIN FUNCTIONS



Pin	Function	Pin	Function
1	NC*	23	NC*
2	P1.0	24	P2.0/A8
3	P1.1	25	P2.1/A9
4	P1.2	26	P2.2/A10
5	P1.3	27	P2.3/A11
6	P1.4	28	P2.4/A12
7	P1.5	29	P2.5/A13
8	P1.6/SCL	30	P2.6/A14
9	P1.7/SDA	31	P2.7/A15
10	RST	32	$\overline{\text{PSEN}}$
11	P3.0/RxD	33	ALE
12	NC*	34	NC*
13	P3.1/TxD	35	$\overline{\text{EA}}$
14	P3.2/$\overline{\text{INT0}}$	36	P0.7/AD7
15	P3.3/$\overline{\text{INT1}}$	37	P0.6/AD6
16	P3.4/T0	38	P0.5/AD5
17	P3.5/T1	39	P0.4/AD4
18	P3.6/$\overline{\text{WR}}$	40	P0.3/AD3
19	P3.7/$\overline{\text{RD}}$	41	P0.2/AD2
20	XTAL2	42	P0.1/AD1
21	XTAL1	43	P0.0/AD0
22	V_{SS}	44	V_{DD}

*DO NOT CONNECT

PLASTIC QUAD FLAT PACK PIN FUNCTIONS



Pin	Function	Pin	Function
1	P1.5	23	P2.5/A13
2	P1.6/SCL	24	P2.6/A14
3	P1.7/SDA	25	P2.7/A15
4	RST	26	$\overline{\text{PSEN}}$
5	P3.0/RxD	27	ALE
6	V_{SS4}	28	V_{SS2}
7	P3.1/TxD	29	$\overline{\text{EA}}$/V_{PP}
8	P3.2/$\overline{\text{INT0}}$	30	P0.7/AD7
9	P3.3/$\overline{\text{INT1}}$	31	P0.6/AD6
10	P3.4/T0	32	P0.5/AD5
11	P3.5/T1	33	P0.4/AD4
12	P3.6/$\overline{\text{WR}}$	34	P0.3/AD3
13	P3.7/$\overline{\text{RD}}$	35	P0.2/AD2
14	XTAL2	36	P0.1/AD1
15	XTAL1	37	P0.0/AD0
16	V_{SS1}	38	V_{DD}
17	NC*	39	V_{SS3}
18	P2.0/A8	40	P1.0
19	P2.1/A9	41	P1.1
20	P2.2/A10	42	P1.2
21	P2.3/A11	43	P1.3
22	P2.4/A12	44	P1.4

*DO NOT CONNECT

NOTES TO QFP ONLY:

1. Due to EMC improvements, all V_{SS} pins (6, 16, 28, 39) must be connected to V_{SS} on the 80C652/83C652.

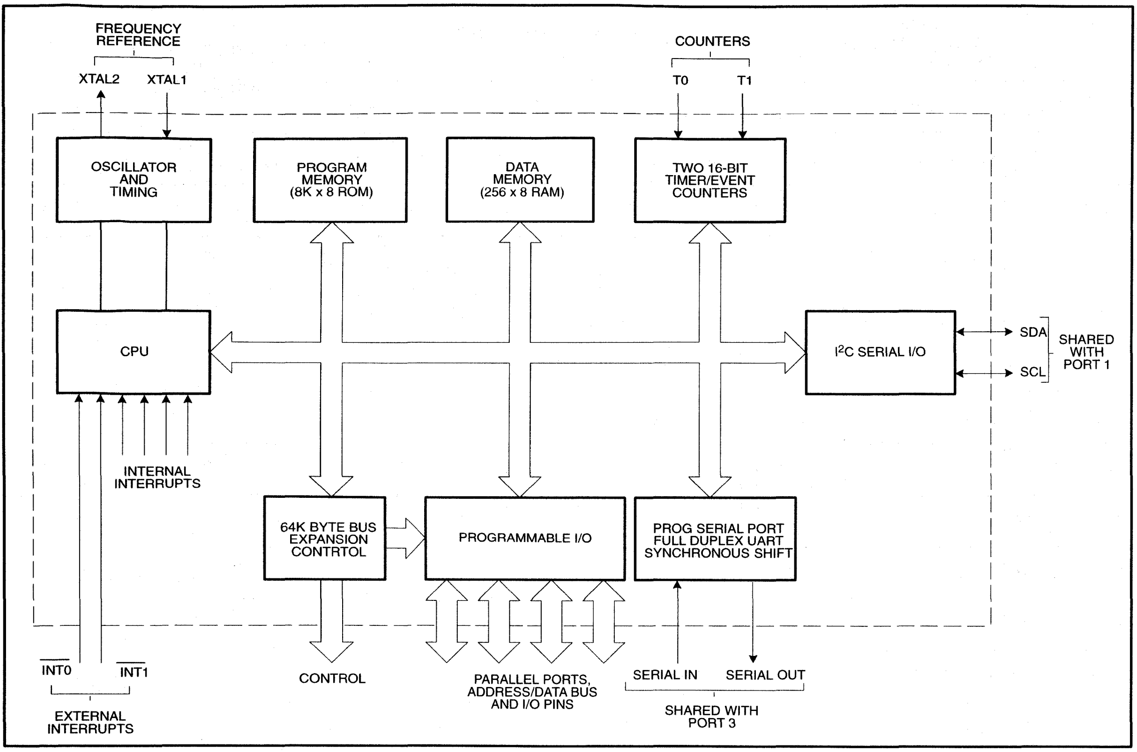
ORDER INFORMATION

PHILIPS PART ORDER NUMBER PART MARKING			PHILIPS NORTH AMERICA PART ORDER NUMBER			TEMPERATURE RANGE (°C) AND PACKAGE	FREQ
ROMless	ROM[3]	Drawing Number	ROMless	ROM	EPROM[2]		MHz[1,2]
P80C652EBP	P83C652EBP/xxx	SOT129-1	P80C652EBPN	P83C652EBPN	S87C652-4N40	0 to +70, Plastic Dual In-line Package	16
P80C652EBA	P83C652EBA/xxx	SOT187-2	P80C652EBAA	P83C652EBAA	S87C652-4A44	0 to +70, Plastic Leaded Chip Carrier	16
P80C652EBB	P83C652EBB/xxx	SOT307-2	P80C652EBBB	P83C652EBBB	S87C652-4B44	0 to +70, Plastic Quad Flat Pack	16
P80C652EFP	P83C652EFP/xxx	SOT129-1	P80C652EFPN	P83C652EFPN	S87C652-5N40	–40 to +85, Plastic Dual In-line Package	16
P80C652EFA	P83C652EFA/xxx	SOT187-2	P80C652EFAA	P83C652EFAA	S87C652-5A44	–40 to +85, Plastic Leaded Chip Carrier	16
P80C652EFB	P83C652EFB/xxx	SOT307-2	P80C652EFBB	P83C652EFBB	S87C652-5B44	–40 to +85, Plastic Quad Flat Pack	16
P80C652EHP	P83C652EHP/xxx	SOT129-1	P80C652EHPN	P83C652EHPN		–40 to +125, Plastic Dual In-line Package	16
P80C652EHA	P83C652EHA/xxx	SOT187-2	P80C652EHAA	P83C652EHAA		–40 to +125, Plastic Leaded Chip Carrier	16
P80C652EHB	P83C652EHB/xxx	SOT307-2	P80C652EHBB	P83C652EHBB		–40 to +125, Plastic Quad Flat Pack	16
P80C652IBP	P83C652IBP/xxx	SOT129-1	P80C652IBPN	P83C652IBPN		0 to +70, Plastic Dual In-line Package	24
P80C652IBA	P83C652IBA/xxx	SOT187-2	P80C652IBAA	P83C652IBAA		0 to +70, Plastic Leaded Chip Carrier	24
P80C652IBB	P83C652IBB/xxx	SOT307-2	P80C652IBBB	P83C652IBBB		0 to +70, Plastic Quad Flat Pack	24
P80C652IFP	P83C652IFP/xxx	SOT129-1	P80C652IFPN	P83C652IFPN		–40 to +85, Plastic Dual In-line Package	24
P80C652IFA	P83C652IFA/xxx	SOT187-2	P80C652IFAA	P83C652IFAA		–40 to +85, Plastic Leaded Chip Carrier	24
P80C652IFB	P83C652IFB/xxx	SOT307-2	P80C652IFBB	P83C652IFBB		–40 to +85, Plastic Quad Flat Pack	24

NOTES:

1. 80C652 and 83C652 frequency range is 3.5MHz–16MHz or 3.5MHz–24MHz.
2. For specification of the EPROM version, see the 87C652 data sheet.
3. xxx denotes the ROM code number.

BLOCK DIAGRAM


FREQUENCY REFERENCE
XTAL2
XTAL1
COUNTERS
T0
T1
OSCILLATOR AND TIMING
PROGRAM MEMORY (8K x 8 ROM)
DATA MEMORY (256 x 8 RAM)
TWO 16-BIT TIMER/EVENT COUNTERS
CPU
I²C SERIAL I/O
SDA
SCL
SHARED WITH PORT 1
INTERNAL INTERRUPTS
64K BYTE BUS EXPANSION CONTRTOL
PROGRAMMABLE I/O
PROG SERIAL PORT FULL DUPLEX UART SYNCHRONOUS SHIFT
INT0
INT1
EXTERNAL INTERRUPTS
CONTROL
PARALLEL PORTS, ADDRESS/DATA BUS AND I/O PINS
SERIAL IN
SERIAL OUT
SHARED WITH PORT 3

CMOS single-chip 8-bit microcontroller | 87C652

DESCRIPTION

The 87C652 Single-Chip 8-Bit Microcontroller is manufactured in an advanced CMOS process and is a derivative of the 80C51 microcontroller family. The 87C652 has the same instruction set as the 80C51.

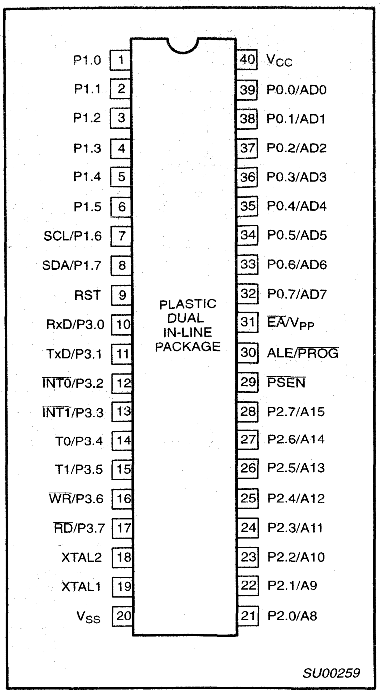
This device provides architectural enhancements that make it applicable in a variety of applications for general control systems. The 87C652 contains a non-volatile 8k × 8 EPROM, a volatile 256 × 8 read/write data memory, four 8-bit I/O ports, two 16-bit timer/event counters (identical to the timers of the 80C51), a multi-source, two-priority-level, nested interrupt structure, an I^{2}C interface, UART and on-chip oscillator and timing circuits. For systems that require extra capability, the 87C652 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: 49 one-byte, 45 two-byte and 17 three-byte. With a 16MHz crystal, 58% of the instructions are executed in 0.75µs and 40% in 1.5µs. Multiply and divide instructions require 3µs.

FEATURES

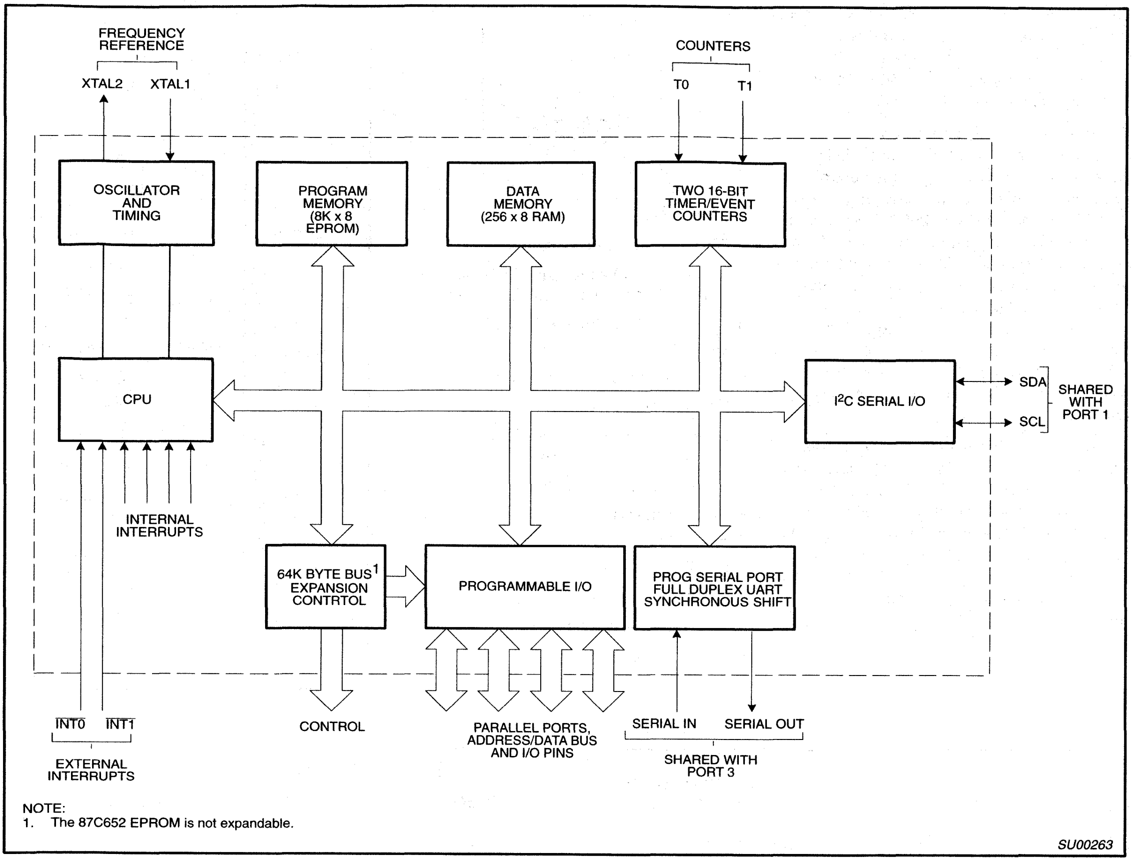
- 80C51 central processing unit
- 8k × 8 EPROM expandable externally to 64k bytes (EPROM is not expandable)
- 256 × 8 RAM, expandable externally to 64k bytes
- Two standard 16-bit timer/counters
- Four 8-bit I/O ports
- I^{2}C-bus serial I/O port with byte oriented master and slave functions
- Full-duplex UART facilities
- Power control modes
  - Idle mode
  - Power-down mode
- Extended temperature range
- OTP package available
- 16MHz speed range

PIN CONFIGURATION



EPROM	Drawing Number	TEMPERATURE RANGE (°C) AND PACKAGE	FREQ MHz
S87C652-4N40	SOT129-1	0 to +70, Plastic Dual In-line Package	16
S87C652-4A44	SOT187-2	0 to +70, Plastic Leaded Chip Carrier	16
S87C652-4B44	SOT307-2	0 to +70, Plastic Quad Flat Pack	16
S87C652-5N40	SOT129-1	–40 to +85, Plastic Dual In-line Package	16
S87C652-5A44	SOT187-2	–40 to +85, Plastic Leaded Chip Carrier	16
S87C652-5B44	SOT307-2	–40 to +85, Plastic Quad Flat Pack	16

BLOCK DIAGRAM


FREQUENCY REFERENCE
XTAL2
XTAL1
COUNTERS
T0
T1
OSCILLATOR AND TIMING
PROGRAM MEMORY (8K x 8 EPROM)
DATA MEMORY (256 x 8 RAM)
TWO 16-BIT TIMER/EVENT COUNTERS
CPU
I²C SERIAL I/O
SDA
SCL
SHARED WITH PORT 1
INTERNAL INTERRUPTS
64K BYTE BUS¹ EXPANSION CONTRTOL
PROGRAMMABLE I/O
PROG SERIAL PORT FULL DUPLEX UART SYNCHRONOUS SHIFT
INT0
INT1
EXTERNAL INTERRUPTS
CONTROL
PARALLEL PORTS, ADDRESS/DATA BUS AND I/O PINS
SERIAL IN
SERIAL OUT
SHARED WITH PORT 3

NOTE:
1. The 87C652 EPROM is not expandable.

SU00263

DESCRIPTION

The P83C654 Single-Chip 8-Bit Microcontroller is manufactured in an advanced CMOS process and is a derivative of the 80C51 microcontroller family. The 83C654 has the same instruction set as the 80C51. Two versions of the derivative exist:

83C654 — 16k bytes mask programmable ROM

87C654 — EPROM version (described in a separate data sheet)

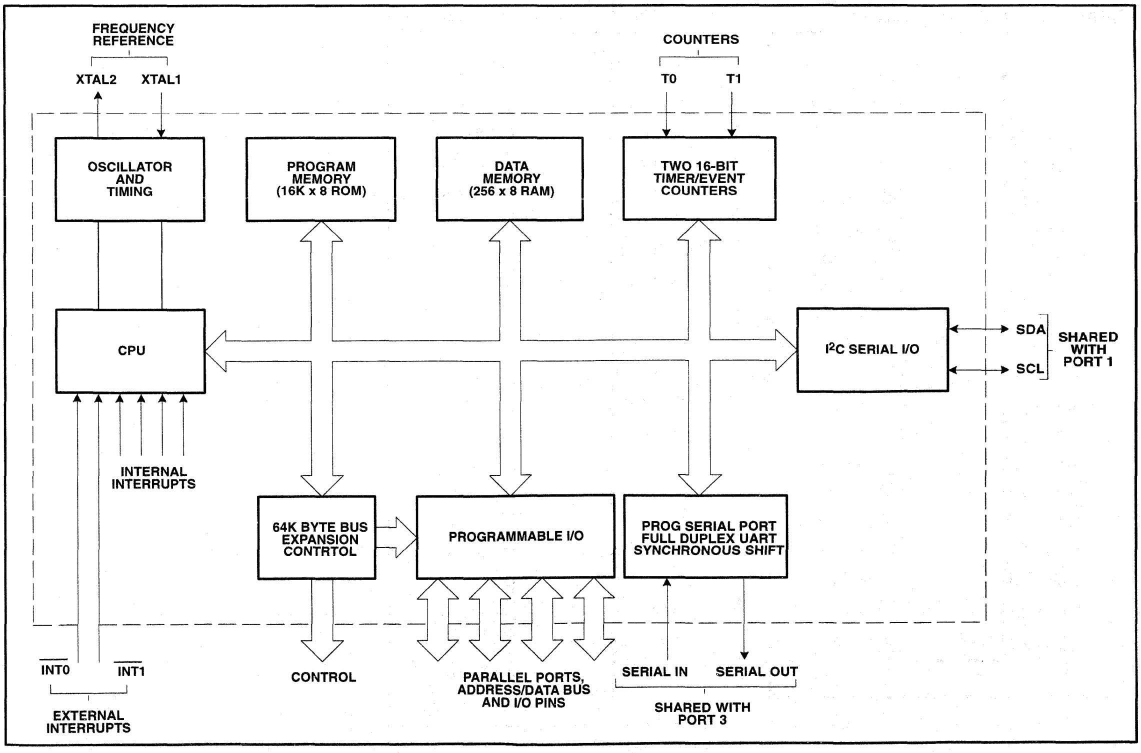
This device provides architectural enhancements that make it applicable in a variety of applications for general control systems. The 83C654 contains a non-volatile 16k × 8 read-only program memory, a volatile 256 × 8 read/write data memory, four 8-bit I/O ports, two 16-bit timer/event counters (identical to the timers of the 80C51), a multi-source, two-priority-level, nested interrupt structure, an I^{2}C interface, UART and on-chip oscillator and timing circuits. For systems that require extra capability, the 8XC654 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: 49 one-byte, 45 two-byte and 17 three-byte. With a 16(24)MHz crystal, 58% of the instructions are executed in 0.75(0.5)µs and 40% in 1.5(1)µs. Multiply and divide instructions require 3(2)µs.

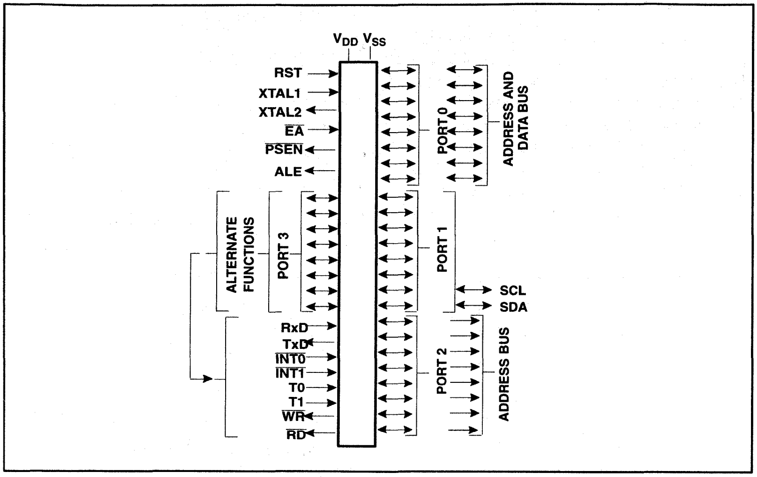
FEATURES

- 80C51 central processing unit
- 16k × 8 ROM expandable externally to 64k bytes
- 256 × 8 RAM, expandable externally to 64k bytes
- Two standard 16-bit timer/counters
- Four 8-bit I/O ports
- I^{2}C-bus serial I/O port with byte oriented master and slave functions
- Full-duplex UART facilities
- Power control modes
  - Idle mode
  - Power-down mode
- ROM code protection
- Extended frequency range: 3.5 to 24 MHz
- Three operating ambient temperature ranges:
  0 to +70°C
  –40 to +85°C
  –40 to +125°C

BLOCK DIAGRAM

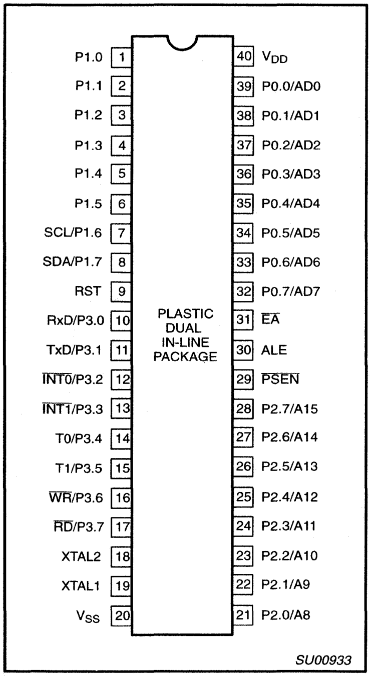

FREQUENCY REFERENCE
XTAL2
XTAL1
COUNTERS
T0
T1
OSCILLATOR AND TIMING
PROGRAM MEMORY (16K x 8 ROM)
DATA MEMORY (256 x 8 RAM)
TWO 16-BIT TIMER/EVENT COUNTERS
CPU
I²C SERIAL I/O
SDA
SCL
SHARED WITH PORT 1
INTERNAL INTERRUPTS
64K BYTE BUS EXPANSION CONTRTOL
PROGRAMMABLE I/O
PROG SERIAL PORT FULL DUPLEX UART SYNCHRONOUS SHIFT
INT0
INT1
EXTERNAL INTERRUPTS
CONTROL
PARALLEL PORTS, ADDRESS/DATA BUS AND I/O PINS
SERIAL IN
SERIAL OUT
SHARED WITH PORT 3

LOGIC SYMBOL

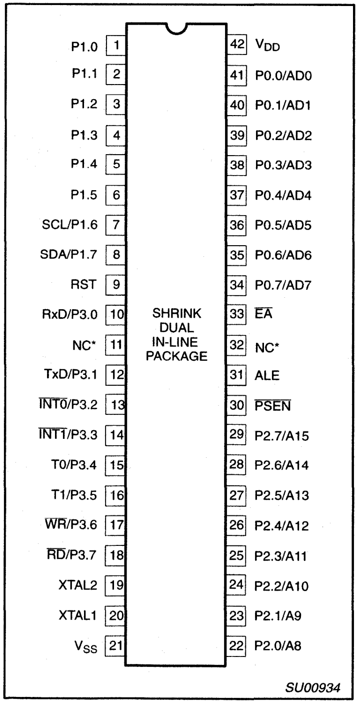


PIN CONFIGURATIONS

Plastic dual in-line package

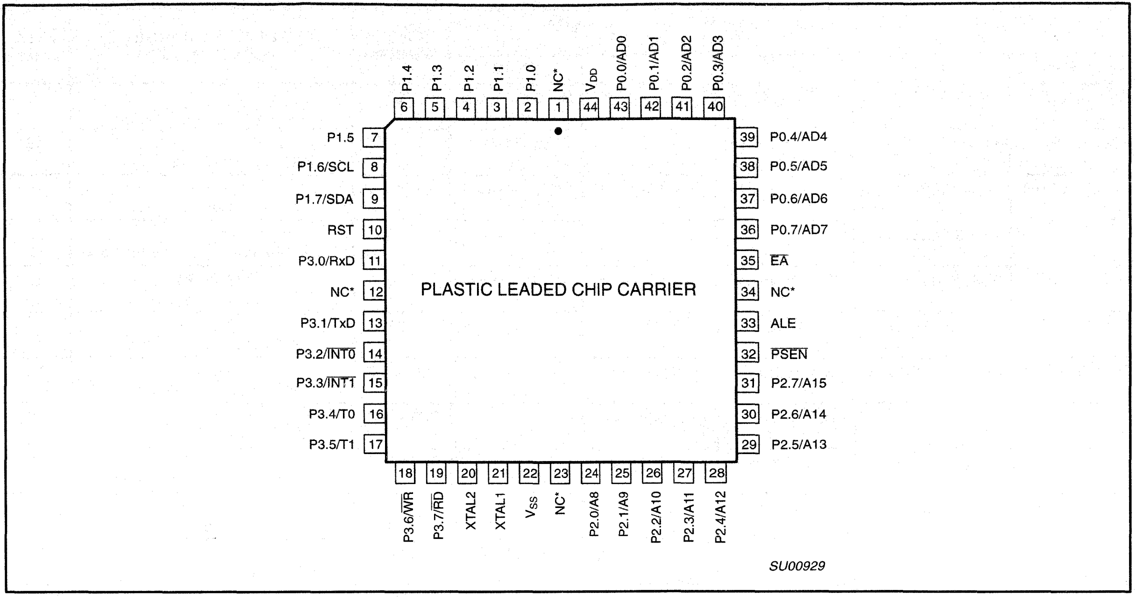


Plastic shrink dual in-line package



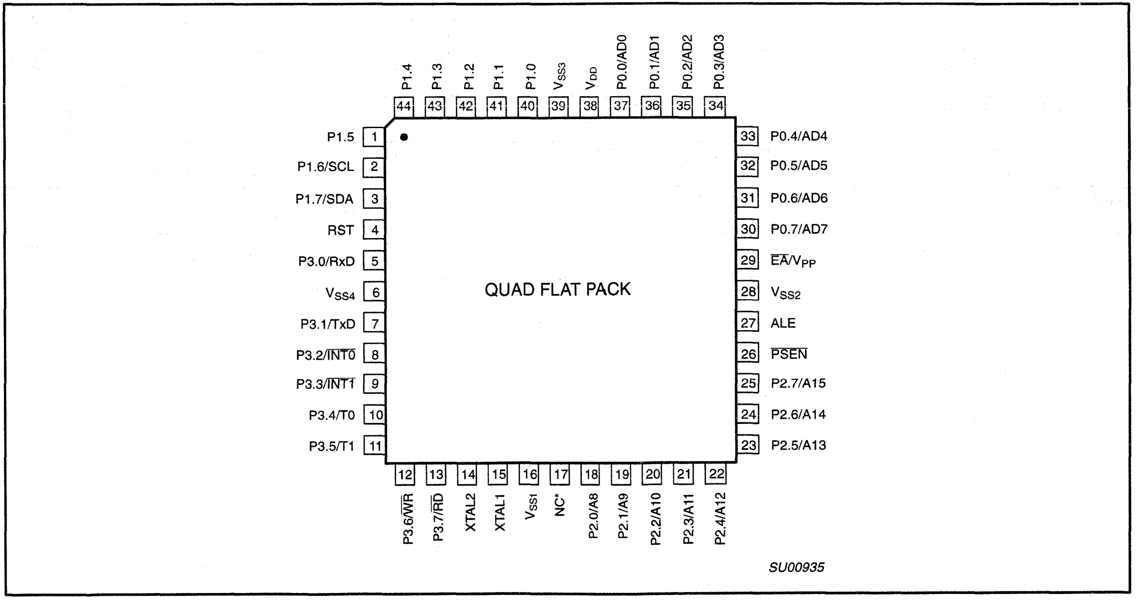
* Do not connect.

Plastic leaded chip carrier



* Do not connect.

Plastic quad flat pack



* Do not connect.

(QFP only): Due to EMC improvements, all V_{SS} pins (6, 16, 28, 39) must be connected to V_{SS} on the 80C652/83C654.

ORDERING INFORMATION

PHILIPS PART ORDER NUMBER PART MARKING		PHILIPS NORTH AMERICA PART ORDER NUMBER			DRAWING NUMBER	TEMPERATURE RANGE (°C) AND PACKAGE	FREQ MHz[2,3]
ROMless[1]	ROM	ROMless[1]	ROM	EPROM[3]			
P80C652EBP	P83C654EBP/xxx	P80C652EBPN	P83C654EBPN	S87C654-4N40	SOT129-1	0 to +70, Plastic Dual In-line Package	16
P80C652EBA	P83C654EBA/xxx	P80C652EBAA	P83C654EBAA	S87C654-4A44	SOT187-2	0 to +70, Plastic Leaded Chip Carrier	16
P80C652EBB	P83C654EBB/xxx	P80C652EBBB	P83C654EBBB	S87C654-4B44	SOT307-2	0 to +70, Plastic Quad Flat Pack	16
	P83C654EBR/xxx				SOT270-1	0 to +70, Plastic Shrink Dual In-Line Package	16
P80C652EFP	P83C654EFP/xxx	P80C652EFPN	P83C654EFPN	S87C654-5N40	SOT129-1	–40 to +85, Plastic Dual In-line Package	16
P80C652EFA	P83C654EFA/xxx	P80C652EFAA	P83C654EFAA	S87C654-5A44	SOT187-2	–40 to +85, Plastic Leaded Chip Carrier	16
P80C652EFB	P83C654EFB/xxx	P80C652EFBB	P83C654EFBB	S87C654-5B44	SOT307-2	–40 to +85, Plastic Quad Flat Pack	16
P80C652EHP	P83C654EHP/xxx	P80C652EHPN	P83C654EHPN		SOT129-1	–40 to +125, Plastic Dual In-line Package	16
P80C652EHA	P83C654EHA/xxx	P80C652EHAA	P83C654EHAA		SOT187-2	–40 to +125, Plastic Leaded Chip Carrier	16
P80C652EHB	P83C654EHB/xxx	P80C652EHBB	P83C654EHBB		SOT307-2	–40 to +125, Plastic Quad Flat Pack	16
				S87C654-7N40	SOT129-1	0 to +70, Plastic Dual In-line Package	20
				S87C654-7A44	SOT187-2	0 to +70, Plastic Leaded Chip Carrier	20
				S87C654-8N40	SOT129-1	–40 to +85, Plastic Dual In-line Package	20
				S87C654-8A44	SOT187-2	–40 to +85, Plastic Leaded Chip Carrier	20
P80C652IBP	P83C654IBP/xxx	P80C652IBPN	P83C654IBPN		SOT129-1	0 to +70, Plastic Dual In-line Package	24
P80C652IBA	P83C654IBA/xxx	P80C652IBAA	P83C654IBAA		SOT187-2	0 to +70, Plastic Leaded Chip Carrier	24
P80C652IBB	P83C654IBB/xxx	P80C652IBBB	P83C654IBBB		SOT307-2	0 to +70, Plastic Quad Flat Pack	24
P80C652IFP	P83C654IFP/xxx	P80C652IFPN	P83C654IFPN		SOT129-1	–40 to +85, Plastic Dual In-line Package	24
P80C652IFA	P83C654IFA/xxx	P80C652IFAA	P83C654IFAA		SOT187-2	–40 to +85, Plastic Leaded Chip Carrier	24
P80C652IFB	P83C654IFB/xxx	P80C652IFBB	P83C654IFBB		SOT307-2	–40 to +85, Plastic Quad Flat Pack	24

NOTES:

1. For full specification, see the 80C652/83C652 data sheet.
2. 83C654 frequency range is 3.5MHz–16MHz or 3.5MHz–24MHz.
3. For specification of the EPROM version, see the 87C654 data sheet.
4. xxx denotes the ROM code number.

CMOS single-chip 8-bit microcontroller | 87C654

DESCRIPTION

The 87C654 Single-Chip 8-Bit Microcontroller is manufactured in an advanced CMOS process and is a derivative of the 80C51 microcontroller family. The 87C654 has the same instruction set as the 80C51. Two versions of the derivative exist:

83C654—16k bytes mask programmable ROM

87C654—EPROM version

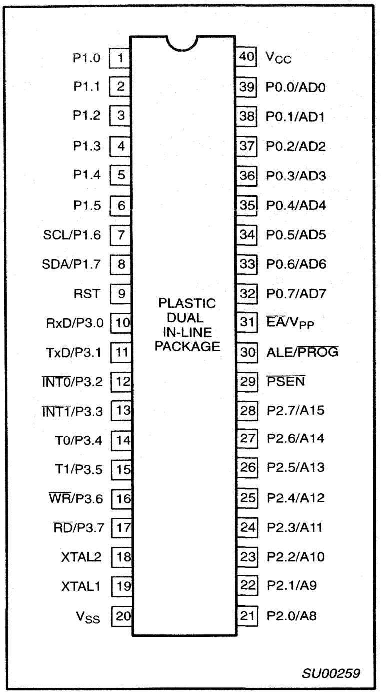
This device provides architectural enhancements that make it applicable in a variety of applications for general control systems. The 87C654 contains a non-volatile 16k × 8 EPROM, a volatile 256 × 8 read/write data memory, four 8-bit I/O ports, two 16-bit timer/event counters (identical to the timers of the 80C51), a multi-source, two-priority-level, nested interrupt structure, an I^{2}C interface, UART and on-chip oscillator and timing circuits. For systems that require extra capability, the 87C654 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: 49 one-byte, 45 two-byte and 17 three-byte. With a 16MHz crystal, 58% of the instructions are executed in 0.75µs and 40% in 1.5µs. Multiply and divide instructions require 3µs.

FEATURES

- 80C51 central processing unit
- 16k × 8 EPROM expandable externally to 64k bytes
- 256 × 8 RAM, expandable externally to 64k bytes
- Two standard 16-bit timer/counters
- Four 8-bit I/O ports
- I^{2}C-bus serial I/O port with byte oriented master and slave functions
- Full-duplex UART facilities
- Power control modes
  - Idle mode
  - Power-down mode
- Extended temperature range
- OTP package available
- Two speed ranges
  - 16MHz
  - 20MHz

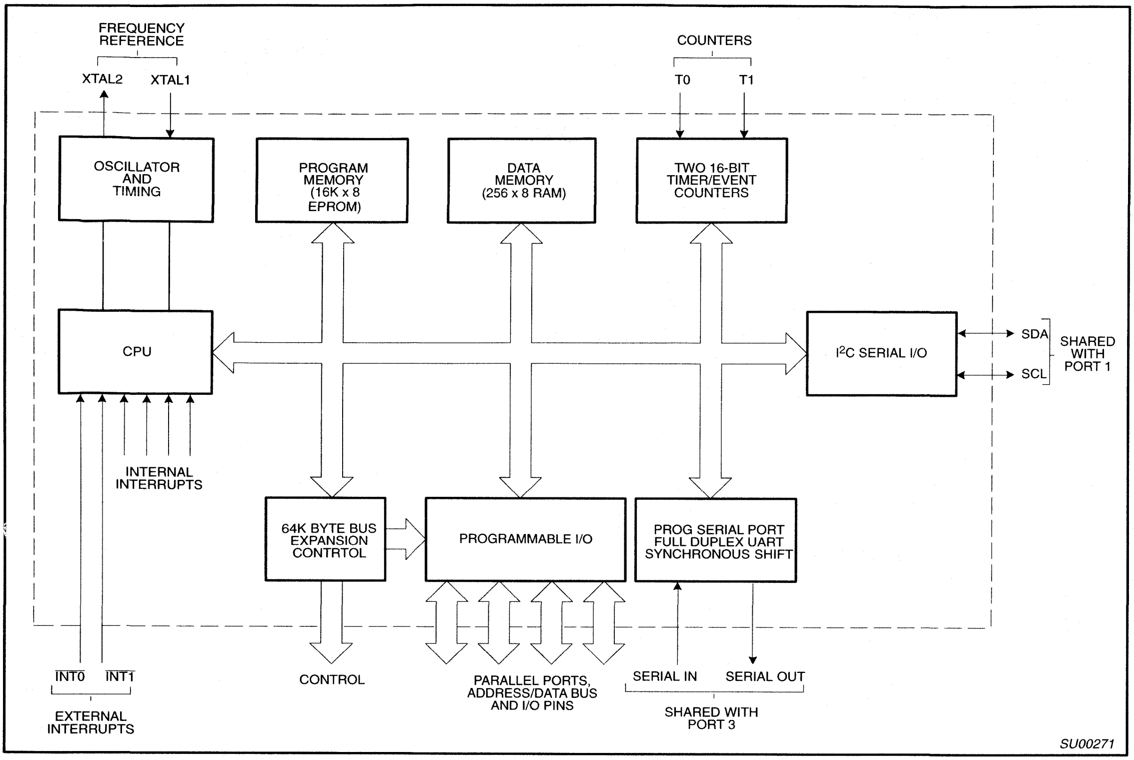
PIN CONFIGURATIONS



EPROM	Drawing Number	TEMPERATURE RANGE °C AND PACKAGE	FREQ MHz
S87C654-4N40	SOT129-1	0 to +70, Plastic Dual In-line Package	16
S87C654-4A44	SOT187-2	0 to +70, Plastic Leaded Chip Carrier	16
S87C654–4B44	SOT307-2	0 to +70, Plastic Quad Flat Pack	16
S87C654-5N40	SOT129-1	–40 to +85, Plastic Dual In-line Package	16
S87C654-5A44	SOT187-2	–40 to +85, Plastic Leaded Chip Carrier	16
S87C654-5B44	SOT307-2	–40 to +85, Plastic Quad Flat Pack	16
S87C654–7N40	SOT129-1	0 to +70, Plastic Dual In-line Package	20
S87C654–7A44	SOT187-2	0 to +70, Plastic Leaded Chip Carrier	20
S87C654–8N40	SOT129-1	–40 to +85, Plastic Dual In-line Package	20
S87C654–8A44	SOT187-2	–40 to +85, Plastic Leaded Chip Carrier	20

NOTES:
1. For ROM see data sheet 83C654
2. For full specification, see the 87C652 data sheet.

BLOCK DIAGRAM



8-bit microcontrollers P89C738; P89C739

1 FEATURES

- 80C51 CPU
- 64 kbytes on-chip Multiple Programming ROM (MTP-ROM), expandable externally to 64 kbytes program memory address space
- 512 bytes on-chip RAM, expandable externally to 64 kbytes data memory address space
- P89C738 pin outs fully compatible to the standard 8051/8052
- 8-bit I/O ports for P89C738: 4 and P89C739: 6
- Full-duplex UART compatible with the standard 80C51 and the 8052
- Two standard 16-bit timers/event counters
- An additional 16-bit timer (functionally equivalent to the Timer 2 of the 8052)
- On-chip Watchdog Timer (T3)
- 6-source and 6-vector interrupt structure with 2 priority levels
- Up to 3 external interrupt request inputs
- Two programmable power reduction modes: Idle and Power-down
- Termination of Idle mode by any interrupt, external or Watchdog Timer reset
- Wake-up from Power-down by external interrupt, external or Watchdog Timer reset
- Packages,
  - P89C738: DIP40, PLCC44 and QFP44
  - P89C739: PLCC68 and QFP64
- Improved Electromagnetic Compatibility (EMC)
- Frequency range: 3.5 to 40 MHz
- ROM code protection

2 GENERAL DESCRIPTION

The P89C738 and P89C739 (hereafter generally referred to as P89C738 unless the P89C739 is specifically mentioned) are 8-bit microcontrollers manufactured in an advanced CMOS process and is a derivative of the PCB80C51 microcontroller family. This device provides architectural enhancements that make it applicable in a variety of applications in general control systems, especially in those systems which need a large on-chip ROM and RAM capacity.

The P89C738 contains a non-volatile 64 kbytes Multiple Programming ROM (MTP-ROM) program memory, a volatile 512 bytes read/write data memory, four 8-bit I/O ports (six for the P89C739), two 16-bit timer/event counters (identical to the timers of the 80C51), a 16-bit timer (identical to the Timer 2 of the 8052), a multi-source two-priority-level nested interrupt structure, one serial interface (UART), a Watchdog Timer (T3), an on-chip oscillator and timing circuits. For systems that require extra capability, the P89C738 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 P89C738 has the same instruction set as the PCB80C51 which consists of over 100 instructions: 49 one-byte, 46 two-byte and 16 three-byte. With a 16 MHz crystal, 58% of the instructions are executed in 750 ns and 40% in 1.5 μs. Multiply and divide instructions require 3 μs.

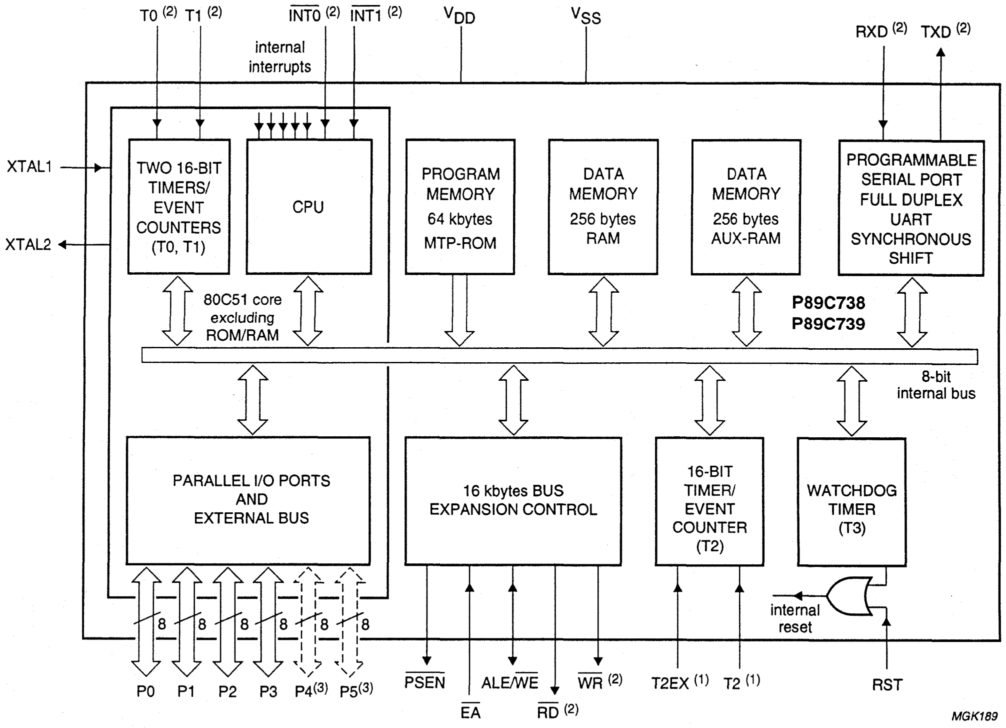
3 ORDERING INFORMATION

TYPE NUMBER[1]	PACKAGE		
	NAME	DESCRIPTION	VERSION
P89C738ABA	PLCC44	plastic leaded chip carrier; 44 leads	note 2
P89C738ABP	DIP40	plastic dual in-line package; 40 leads (600 mil)	SOT129-1
P89C738BBB	QFP44	plastic quad flat package; 44 leads	note 2
P89C739ABA	PLCC68	plastic leaded chip carrier; 68 leads	note 2
P89C739ABB	QFP64	plastic quad flat package; 64 leads (lead length 1.95 mm); body 14 × 20 × 2.7 mm; high stand-off height	SOT319-1

Note

1. Temperature and frequency range for all types: 0 to 70 °C and 3.5 to 40 MHz.
2. For more information on the package outline of this version, please contact the Philips Semiconductors Sales office.

4 BLOCK DIAGRAM


T0 (2)
T1 (2)
INT0 (2)
INT1 (2)
internal interrupts
VDD
VSS
RXD (2)
TXD (2)
XTAL1
XTAL2
TWO 16-BIT TIMERS/ EVENT COUNTERS (T0, T1)
CPU
PROGRAM MEMORY 64 kbytes MTP-ROM
DATA MEMORY 256 bytes RAM
DATA MEMORY 256 bytes AUX-RAM
PROGRAMMABLE SERIAL PORT FULL DUPLEX UART SYNCHRONOUS SHIFT
80C51 core excluding ROM/RAM
P89C738
P89C739
8-bit internal bus
PARALLEL I/O PORTS AND EXTERNAL BUS
16 kbytes BUS EXPANSION CONTROL
16-BIT TIMER/ EVENT COUNTER (T2)
WATCHDOG TIMER (T3)
internal reset
8
P0
P1
P2
P3
P4(3)
P5(3)
PSEN
EA
ALE/WE
RD (2)
WR (2)
T2EX (1)
T2 (1)
RST
MGK189

(1) Alternative function for Port 1.

(2) Alternative function for Port 3.

(3) P4 and P5 are only available on the P89C738ABA and P89C739ABB (PLCC68 and QFP64).

Fig.1 Block diagram.

CMOS single-chip 8-bit microcontrollers 83C748/87C748

DESCRIPTION

The Philips 83C748/87C748 offers the advantages of the 80C51 architecture in a small package and at low cost.

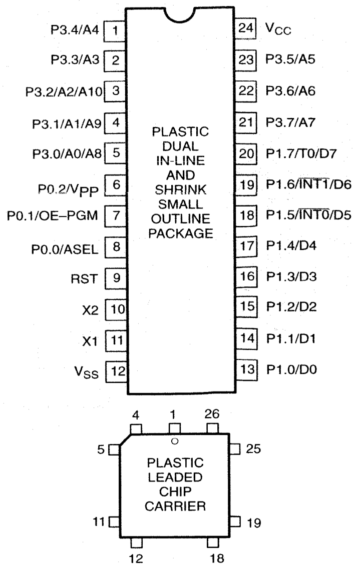
The 8XC748 Microcontroller is fabricated with Philips high-density CMOS technology. Philips epitaxial substrate minimizes CMOS latch-up sensitivity.

The 8XC748 contains a 2k × 8 ROM (83C748) EPROM (87C748), a 64 × 8 RAM, 19 I/O lines, a 16-bit auto-reload counter/timer, a four-source, fixed-priority level interrupt structure, and an on-chip oscillator.

FEATURES

- 80C51 based architecture
- Small package sizes
  - 24-pin DIP (300 mil "skinny DIP")
  - 24-pin Shrink Small Outline Package (SSOP)
  - 28-pin PLCC
- 87C748 available in one-time programmable plastic packages
- Wide oscillator frequency range: –3.5 to 16MHz
- Low power consumption:
  - Normal operation: less than 11mA @ 5V, 12MHz
  - Idle mode
  - Power-down mode
- 2k × 8 ROM (83C748)
  2k × 8 EPROM (87C748)
- 64 × 8 RAM
- 16-bit auto reloadable counter/timer
- 10-bit fixed-rate timer
- Boolean processor
- CMOS and TTL compatible
- Well suited for logic replacement, consumer and industrial applications
- LED drive outputs

PIN CONFIGURATIONS



Pin	Function	Pin	Function
1	P3.4/A4	15	P1.0/D0
2	P3.3/A3	16	P1.1/D1
3	P3.2/A2/A10	17	P1.2/D2
4	P3.1/A1/A9	18	P1.3/D3
5	NC*	19	P1.4/D4
6	P3.0/A0/A8	20	P1.5/$\overline{\text{INT0}}$/D5
7	P0.2/V_{PP}	21	NC*
8	P0.1/OE-PGM	22	NC*
9	P0.0/ASEL	23	P1.6/$\overline{\text{INT1}}$/D6
10	NC*	24	P1.7/T0/D7
11	RST	25	P3.7/A7
12	X2	26	P3.6/A6
13	X1	27	P3.5/A5
14	V_{SS}	28	V_{CC}

* NO INTERNAL CONNECTION

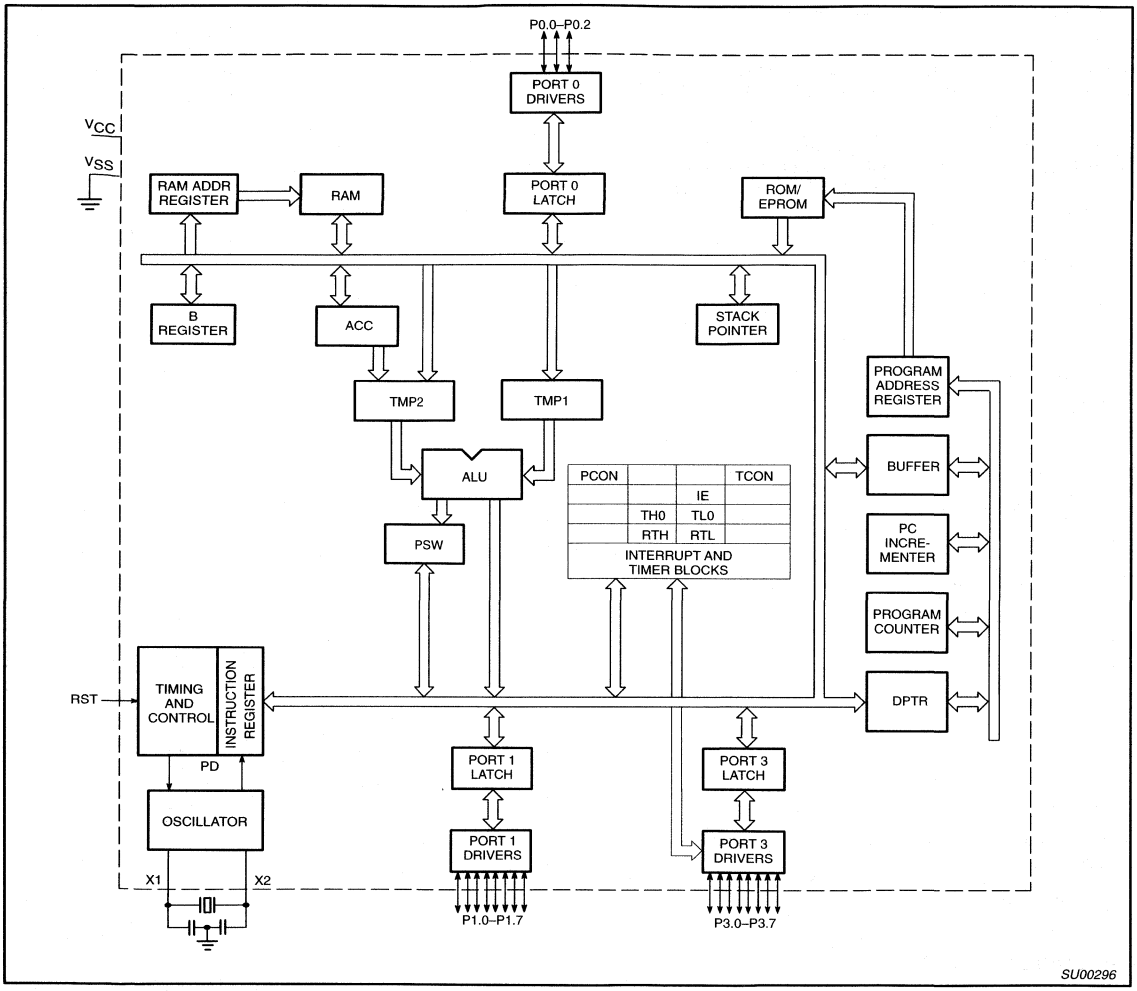
SU00295A

ORDERING INFORMATION

ROM	EPROM[1]		TEMPERATURE RANGE °C AND PACKAGE	FREQUENCY MHz	DRAWING NUMBER
P83C748EBP N	P87C748EBP N	OTP	0 to +70, Plastic Dual In-line Package	3.5 to 16	SOT222-1
P83C748EBA A	P87C748EBA A	OTP	0 to +70, Plastic Leaded Chip Carrier	3.5 to 16	SOT261-3
P83C748EBD DB	P87C748EBD DB	OTP	0 to +70, Shrink Small Outline Package	3.5 to 16	SOT340-1

NOTE:
1. OTP = One Time Programmable EPROM.

BLOCK DIAGRAM



CMOS single-chip 8-bit microcontrollers 83C749/87C749

DESCRIPTION

The Philips 83C749/87C749 offers many of the advantages of the 80C51 architecture in a small package and at low cost.

The 8XC749 Microcontroller is fabricated with Philips high-density CMOS technology. Philips epitaxial substrate minimizes CMOS latch-up sensitivity.

The 8XC749 contains a 2k × 8 ROM (83C749) EPROM (87C749), a 64 × 8 RAM, 21 I/O lines, a 16-bit auto-reload counter/timer, a fixed-priority level interrupt structure, an on-chip oscillator, a five channel multiplexed 8-bit A/D converter, and an 8-bit PWM output.

The EPROM version of this device, the 87C749, is available in plastic one-time programmable (OTP) packages. Once the array has been programmed, it is functionally equivalent to the masked ROM 83C749. Thus, unless explicitly stated otherwise, all references made to the 83C749 apply equally to the 87C749.

The 83C749 supports two power reduction modes of operation referred to as the idle mode and the power-down mode.

FEATURES

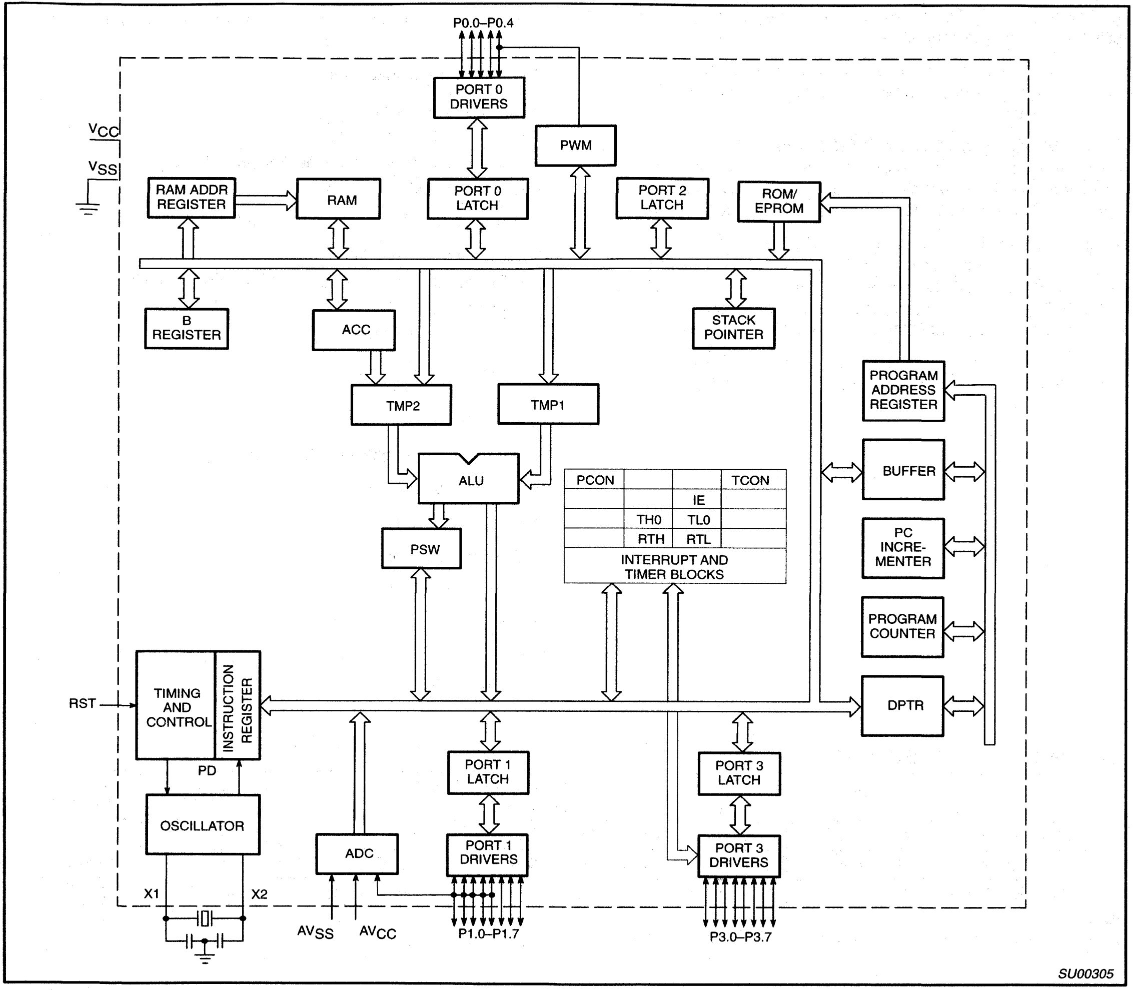
- Available in erasable quartz lid or One-Time Programmable plastic packages
- 80C51 based architecture
- Small package sizes
  - 28-pin DIP
  - 28-pin Shrink Small Outline Package (SSOP)
  - 28-pin PLCC
- Wide oscillator frequency range: 3.5MHz to 16MHz
- Low power consumption:
  - Normal operation: less than 11mA @ 5V, 12MHz
  - Idle mode
  - Power-down mode
- 2k × 8 ROM (83C749)
  EPROM (87C749)
- 64 × 8 RAM
- 16-bit auto reloadable counter/timer
- 5-channel 8-bit A/D converter
- 8-bit PWM output/timer
- 10-bit fixed-rate timer
- Boolean processor
- CMOS and TTL compatible
- Well suited for logic replacement, consumer and industrial applications

PART NUMBER SELECTION

ROM	EPROM[1]		TEMPERATURE RANGE °C AND PACKAGE	FREQUENCY	DRAWING NUMBER
P83C749EBP N	P87C749EBP N	OTP	0 to +70, 28-pin Plastic Dual In-line Package	3.5 to 16MHz	SOT117-2
P83C749EBA A	P87C749EBA A	OTP	0 to +70, 28-pin Plastic Leaded Chip Carrier	3.5 to 16MHz	SOT261-3
P83C749EBD DB	P87C749EBD DB	OTP	0 to +70, 28-pin Shrink Small Outline Package	3.5 to 16MHz	SOT341-1

NOTE:
1. OTP = One Time Programmable EPROM. UV = UV Erasable EPROM.

BLOCK DIAGRAM



Microcontroller with TrackPoint™ microcode from IBM TPM749

DESCRIPTION

The Philips Semiconductors TPM749 is a small package, low cost, ROM-coded 80C51 with IBM®'s TrackPoint™ pointing algorithms and control code. TrackPoint is the result of years of human factors research and innovation at IBM. The result is a "velocity sensitive" pointing solution more efficient and easier to use than "position sensitive" devices such as the mouse, the trackball, or the touchpad.

IBM has licensed Philips Semiconductors to sell microcontrollers with TrackPoint code. By purchasing a TPM from Philips, the purchaser becomes a sub-licensee of Philips. The selling price of Philips' TPM includes the royalties for IBM's intellectual property, which Philips in turn pays to IBM. Customers for TPMs do not need to sign any licensing agreement with either IBM or Philips. This code is the intellectual property of IBM, which is covered by numerous patents, and must be treated accordingly.

The TPM is fabricated with Philips high-density CMOS technology. Philips epitaxial substrate minimizes CMOS latch-up sensitivity.

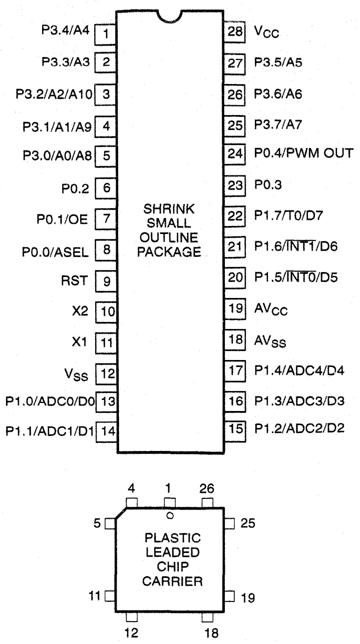
The TPM contains a 2k × 8 ROM, a 64 × 8 RAM, 21 I/O lines, a 16-bit auto-reload counter/timer, a fixed-priority level interrupt structure, an on-chip oscillator, a five channel multiplexed 8-bit A/D converter, and an 8-bit PWM output.

The TPM supports two power reduction modes of operation referred to as the idle mode and the power-down mode.

FEATURES

- 80C51 based architecture
- Small package sizes
  - 28-pin Shrink Small Outline Package (SSOP)
  - 28-pin PLCC
- Low power consumption:
  - Normal operation: less than 11mA @ 5V, 12MHz
  - Idle mode
  - Power-down mode
- 2k × 8 ROM
- 64 × 8 RAM
- 16-bit auto reloadable counter/timer
- 5-channel 8-bit A/D converter
- 8-bit PWM output/timer
- 10-bit fixed-rate timer
- CMOS and TTL compatible

PIN CONFIGURATION



Pin	Function	Pin	Function
1	P3.4/A4	15	P1.2/ADC2/D2
2	P3.3/A3	16	P1.3/ADC3/D3
3	P3.2/A2/A10	17	P1.4/ADC4/D4
4	P3.1/A1/A9	18	AV_{SS}
5	P3.0/A0/A8	19	AV_{CC}
6	P0.2	20	P1.5/$\overline{\text{INT0}}$/D5
7	P0.1/OE	21	P1.6/$\overline{\text{INT1}}$/D6
8	P0.0/ASEL	22	P1.7/T0/D7
9	RST	23	P0.3
10	X2	24	P0.4/PWM OUT
11	X1	25	P3.7/A7
12	V_{SS}	26	P3.6/A6
13	P1.0/ADC0/D0	27	P3.5/A5
14	P1.1/ADC1/D1	28	V_{CC}

SU00692A

ORDERING INFORMATION

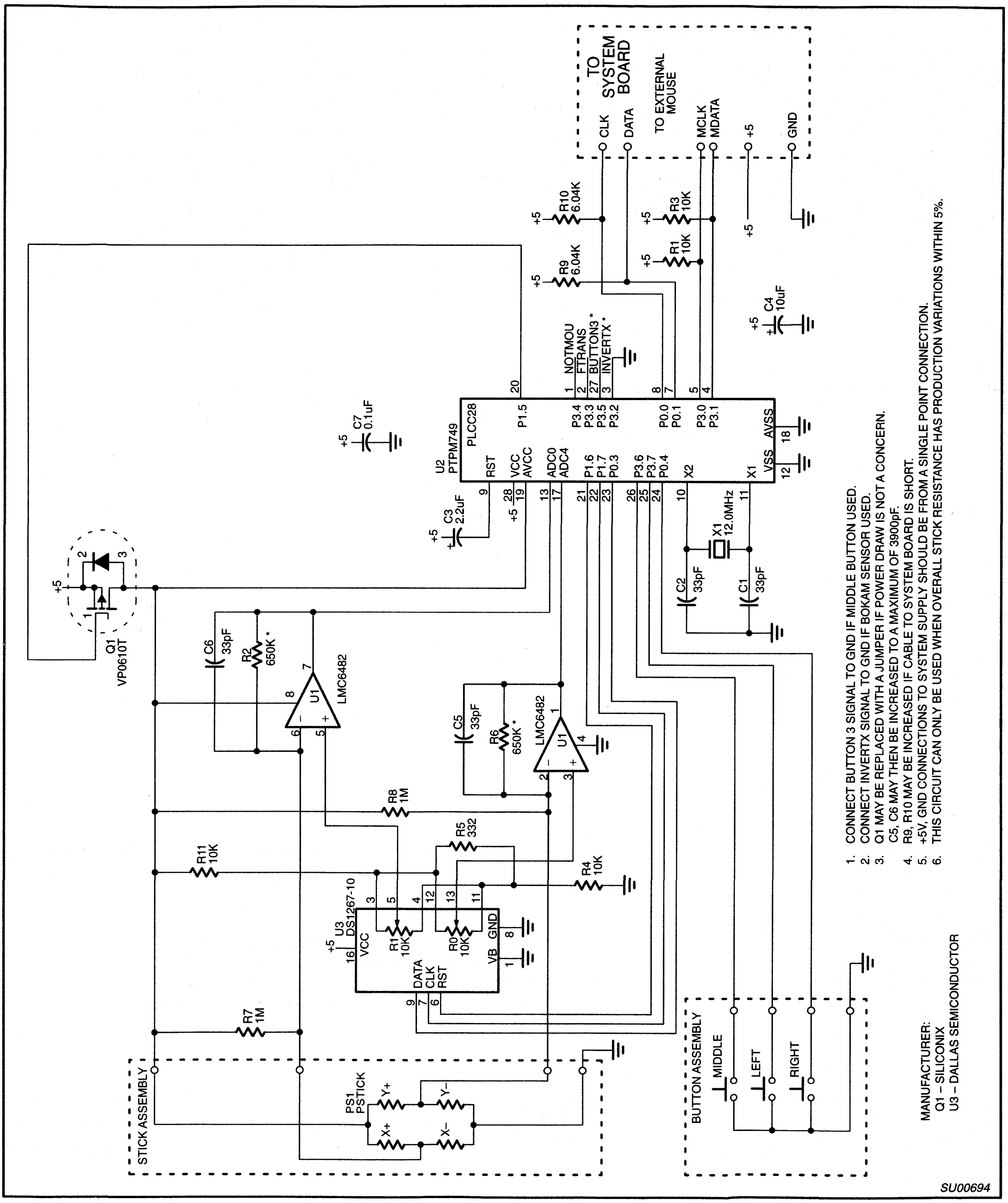
ORDERING CODE	TEMPERATURE RANGE AND PACKAGE	DRAWING NUMBER
PTPM749 A	0 to +70°C, Plastic Leaded Chip Carrier	SOT261-3
PTPM749 DB	0 to +70°C, Shrink Small Ouline Package	SOT341-1

For compatible pointing device, contact:

COMPANY	CONTACT	TELEPHONE
Bokam Engineering	Ms. Jane Kamenster	(714) 513–2200
CTS Corporation	Mr. Dave Poole	(219) 589–7169

IBM is a registered trademark, and TrackPoint is a trademark of IBM Corporation.

SCHEMATIC OF TrackPoint SYSTEM WITH PHILIPS TPM749



1. CONNECT BUTTON 3 SIGNAL TO GND IF MIDDLE BUTTON USED.
2. CONNECT INVERTX SIGNAL TO GND IF BOKAM SENSOR USED.
3. Q1 MAY BE REPLACED WITH A JUMPER IF POWER DRAW IS NOT A CONCERN. C5, C6 MAY THEN BE INCREASED TO A MAXIMUM OF 3900pF.
4. R9, R10 MAY BE INCREASED IF CABLE TO SYSTEM BOARD IS SHORT.
5. +5V, GND CONNECTIONS TO SYSTEM SUPPLY SHOULD BE FROM A SINGLE POINT CONNECTION.
6. THIS CIRCUIT CAN ONLY BE USED WHEN OVERALL STICK RESISTANCE HAS PRODUCTION VARIATIONS WITHIN 5%.

MANUFACTURER:
Q1 – SILICONIX
U3 – DALLAS SEMICONDUCTOR

SU00694

CMOS single-chip 8-bit microcontrollers 83C750/87C750

DESCRIPTION

The Philips 8XC750 offers the advantages of the 80C51 architecture in a small package and at low cost.

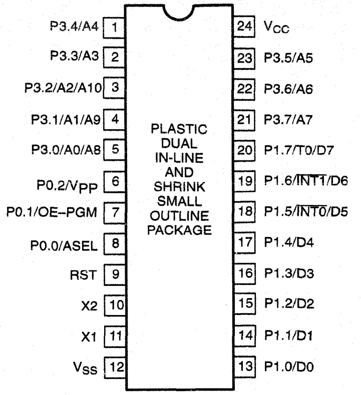
The 8XC750 Microcontroller is fabricated with Philips high-density CMOS technology. Philips epitaxial substrate minimizes CMOS latch-up sensitivity.

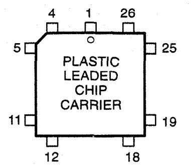
The 87C750 contains a 1k × 8 EPROM, a 64 × 8 RAM, 19 I/O lines, a 16-bit auto-reload counter/timer, a five-source, fixed-priority level interrupt structure and an on-chip oscillator.

FEATURES

- 80C51 based architecture
- Oscillator frequency range—up to 16MHz
- Small package sizes
  - 24-pin DIP (300 mil “skinny DIP”)
  - 24-pin Shrink Small Outline Package
  - 28-pin PLCC
- 87C750 available in one-time programmable plastic packages
- Low power consumption:
  - Normal operation: less than 11mA @ 5V, 12MHz
  - Idle mode
  - Power-down mode
- 1k × 8 EPROM (87C750)
- 64 × 8 RAM
- 16-bit auto reloadable counter/timer
- Boolean processor
- CMOS and TTL compatible
- Well suited for logic replacement, consumer and industrial applications
- LED drive outputs

PIN CONFIGURATIONS





Pin	Function	Pin	Function
1	P3.4/A4	15	P1.0/D0
2	P3.3/A3	16	P1.1/D1
3	P3.2/A2/A10	17	P1.2/D2
4	P3.1/A1/A9	18	P1.3/D3
5	NC*	19	P1.4/D4
6	P3.0/A0/A8	20	P1.5/$\overline{INT0}$/D5
7	P0.2/V_{PP}	21	NC*
8	P0.1/OE-PGM	22	NC*
9	P0.0/ASEL	23	P1.6/$\overline{INT1}$/D6
10	NC*	24	P1.7/T0/D7
11	RST	25	P3.7/A7
12	X2	26	P3.6/A6
13	X1	27	P3.5/A5
14	V_{SS}	28	V_{CC}

* NO INTERNAL CONNECTION

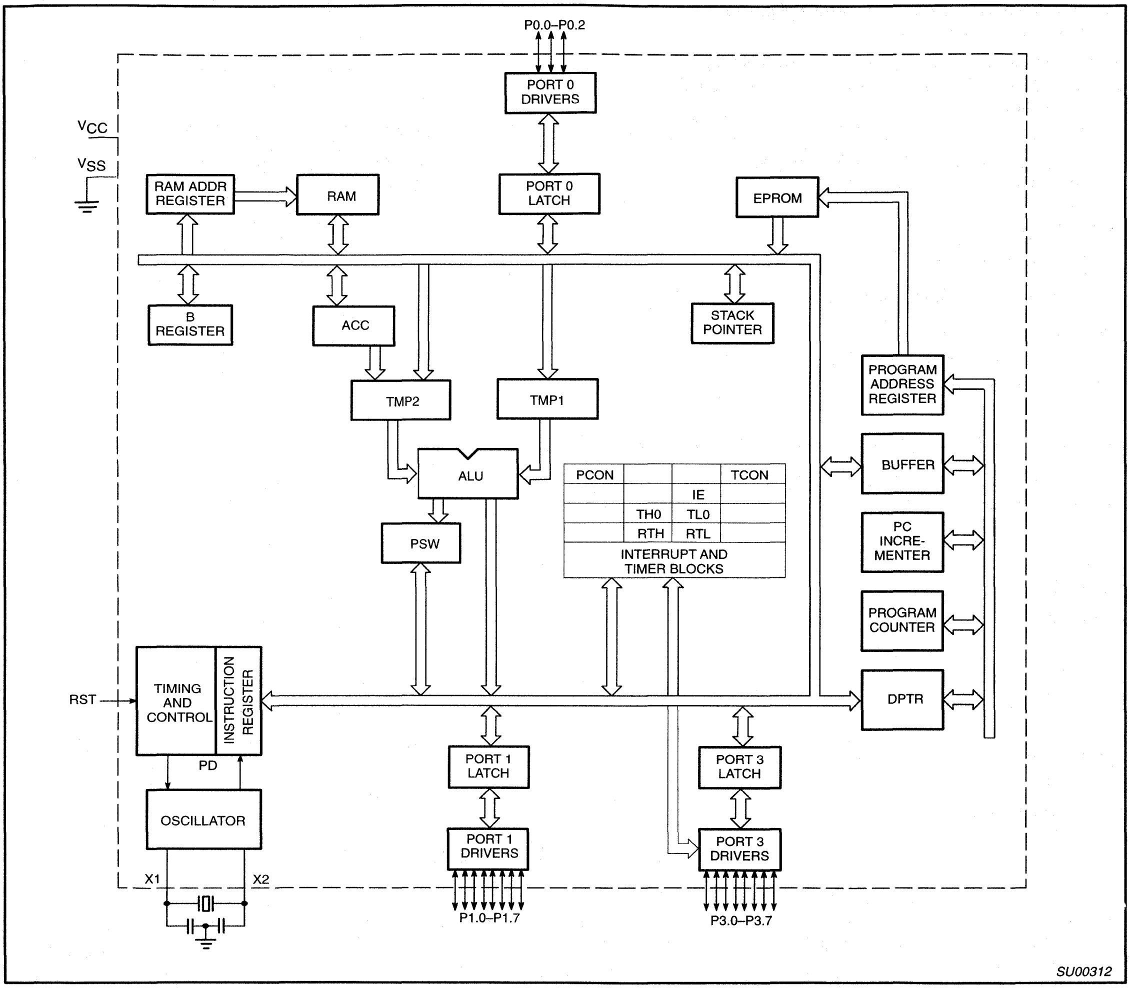
SU00295A

ORDERING INFORMATION

ROM	EPROM[1]		TEMPERATURE RANGE °C AND PACKAGE	FREQUENCY	DRAWING NUMBER
P83C750EBP N	P87C750EBP N	OTP	0 to +70, Plastic Dual In-line Package	3.5 to 16MHz	SOT222-1
P83C750EFP N	P87C750EFP N	OTP	–40 to +85, Plastic Dual In-line Package	3.5 to 16MHz	SOT222-1
P83C750EBA A	P87C750EBA A	OTP	0 to +70, Plastic Lead Chip Carrier	3.5 to 16MHz	SOT261-3
P83C750EFA A	P87C750EFA A	OTP	–40 to +85, Plastic Lead Chip Carrier	3.5 to 16MHz	SOT261-3
P83C750EBD DB	P87C750EBD DB	OTP	0 to +70, Shrink Small Outline Package	3.5 to 16MHz	SOT340-1

NOTE:
1. OTP = One Time Programmable EPROM.

BLOCK DIAGRAM



DESCRIPTION

The Philips 83C751/87C751 offers the advantages of the 80C51 architecture in a small package and at low cost.

The 8XC751 Microcontroller is fabricated with Philips high-density CMOS technology. Philips epitaxial substrate minimizes CMOS latch-up sensitivity.

The 8XC751 contains a 2k × 8 ROM (83C751) EPROM (87C751), a 64 × 8 RAM, 19 I/O lines, a 16-bit auto-reload counter/timer, a five-source, fixed-priority level interrupt structure, a bidirectional inter-integrated circuit (I^{2}C) serial bus interface, and an on-chip oscillator.

The on-board inter-integrated circuit (I^{2}C) bus interface allows the 8XC751 to operate as a master or slave device on the I^{2}C small area network. This capability facilitates I/O and RAM expansion, access to EEPROM, processor-to-processor communication, and efficient interface to a wide variety of dedicated I^{2}C peripherals.

FEATURES

- 80C51 based architecture
- Inter-Integrated Circuit (I^{2}C) serial bus interface
- Small package sizes
  - 24-pin DIP (300 mil "skinny DIP")
  - 24-pin Shrink Small Outline Package
  - 28-pin PLCC
- 87C751 available in one-time programmable plastic packages
- Wide oscillator frequency range
- Low power consumption:
  - Normal operation: less than 11mA @ 5V, 12MHz
  - Idle mode
  - Power-down mode
- 2k × 8 ROM (83C751)
  2k × 8 EPROM (87C751)
- 64 × 8 RAM
- 16-bit auto reloadable counter/timer
- Fixed-rate timer
- Boolean processor
- CMOS and TTL compatible
- Well suited for logic replacement, consumer and industrial applications
- LED drive outputs

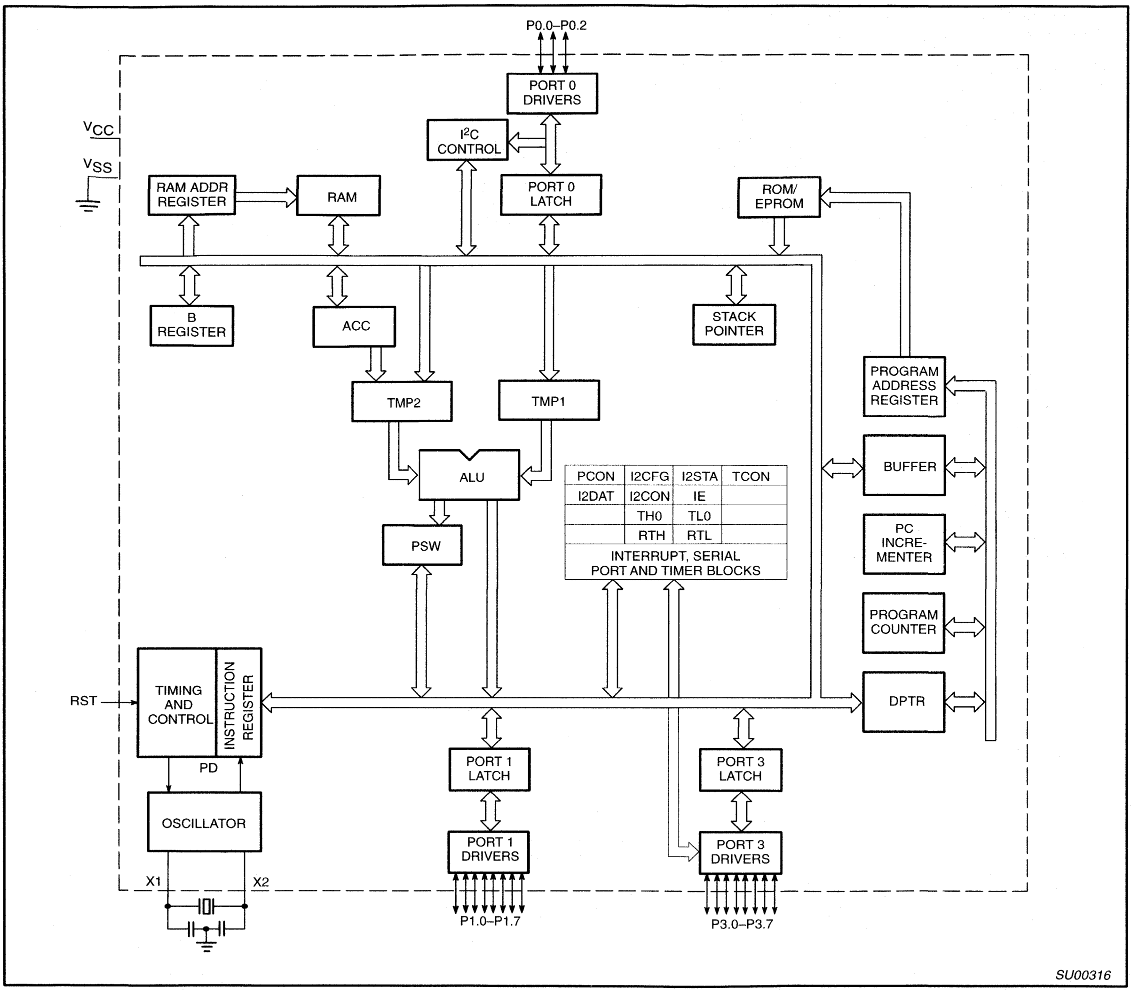
ORDERING INFORMATION

ROM	EPROM[1]		TEMPERATURE RANGE °C AND PACKAGE	FREQUENCY	DRAWING NUMBER
S83C751–1N24	S87C751–1N24	OTP	0 to +70, Plastic Dual In-line Package	3.5 to 12MHz	SOT222-1
S83C751–2N24	S87C751–2N24	OTP	–40 to +85, Plastic Dual In-line Package	3.5 to 12MHz	SOT222-1
S83C751–4N24	S87C751–4N24	OTP	0 to +70, Plastic Dual In-line Package	3.5 to 16MHz	SOT222-1
S83C751–5N24	S87C751–5N24	OTP	–40 to +85, Plastic Dual In-line Package	3.5 to 16MHz	SOT222-1
S83C751–1A28	S87C751–1A28	OTP	0 to +70, Plastic Leaded Chip Carrier	3.5 to 12MHz	SOT261-3
S83C751–2A28	S87C751–2A28	OTP	–40 to +85, Plastic Leaded Chip Carrier	3.5 to 12MHz	SOT261-3
S83C751–4A28	S87C751–4A28	OTP	0 to +70, Plastic Leaded Chip Carrier	3.5 to 16MHz	SOT261-3
S83C751–5A28	S87C751–5A28	OTP	–40 to +85, Plastic Leaded Chip Carrier	3.5 to 16MHz	SOT261-3
S83C751–1DB	S87C751–1DB	OTP	0 to +70, Shrink Small Outline Package	3.5 to 12MHz	SOT340-1
S83C751–4DB	S87C751–4DB	OTP	0 to +70, Shrink Small Outline Package	3.5 to 16MHz	SOT340-1

NOTE:

[1] OTP = One Time Programmable EPROM. UV = UV Erasable EPROM.

BLOCK DIAGRAM



CMOS single-chip 8-bit microcontroller with A/D, PWM

DESCRIPTION

The Philips 83C752/87C752 offers many of the advantages of the 80C51 architecture in a small package and at low cost.

The 8XC752 Microcontroller is fabricated with Philips high-density CMOS technology. Philips epitaxial substrate minimizes CMOS latch-up sensitivity.

The 8XC752 contains a 2k × 8 ROM (83C752) EPROM (87C752), a 64 × 8 RAM, 21 I/O lines, a 16-bit auto-reload counter/timer, a fixed-priority level interrupt structure, a bidirectional inter-integrated circuit (I^2C) serial bus interface, an on-chip oscillator, a five channel multiplexed 8-bit A/D converter, and an 8-bit PWM output.

The onboard inter-integrated circuit (I^2C) bus interface allows the 8XC752 to operate as a master or slave device on the I^2C small area network. This capability facilitates I/O and RAM expansion, access to EEPROM, processor-to-processor communication, and efficient interface to a wide variety of dedicated I^2C peripherals.

The EPROM version of this device, the 87C752, is also available in both quartz-lid erasable and plastic one-time programmable (OTP) packages. Once the array has been programmed, it is functionally equivalent to the masked ROM 83C752. Thus, unless explicitly stated otherwise, all references made to the 83C752 apply equally to the 87C752.

The 83C752 supports two power reduction modes of operation referred to as the idle mode and the power-down mode.

FEATURES

- Available in erasable quartz lid or One-Time Programmable plastic packages
- 80C51 based architecture
- Inter-integrated Circuit (I^2C) serial bus interface
- Small package sizes
  - 28-pin DIP
  - 28-pin PLCC
  - 28-pin SSOP
- Wide oscillator frequency range
- Low power consumption:
  - Normal operation: less than 11mA @ 5V, 12MHz
  - Idle mode
  - Power-down mode
- 2k × 8 ROM (83C752)
  EPROM (87C752)
- 64 × 8 RAM
- 16-bit auto reloadable counter/timer
- 5-channel 8-bit A/D converter
- 8-bit PWM output/timer
- Fixed-rate timer
- Boolean processor
- CMOS and TTL compatible
- Well suited for logic replacement, consumer and industrial applications

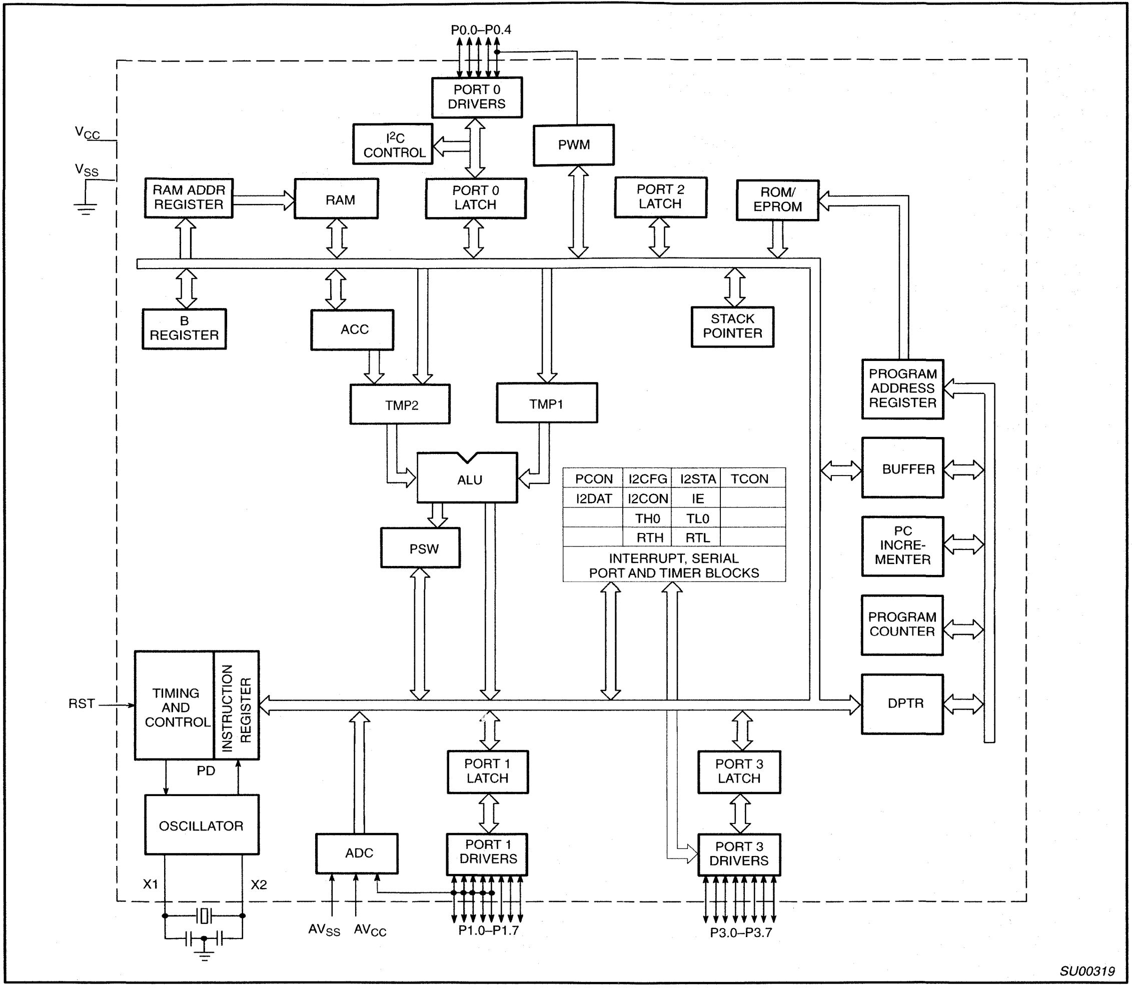
PART NUMBER SELECTION

ROM	EPROM		TEMPERATURE RANGE °C AND PACKAGE	FREQUENCY	DRAWING NUMBER
S83C752–1DB	S87C752–1DB	OTP	0 to +70, 28-pin Plastic Shrink Small Outline Package	3.5 to 12MHz	SOT341-1
S83C752–1N28	S87C752–1N28	OTP	0 to +70, 28-pin Plastic Dual In-line Package	3.5 to 12MHz	SOT117-2
S83C752–2N28	S87C752–2N28	OTP	–40 to +85, 28-pin Plastic Dual In-line Package	3.5 to 12MHz	SOT117-2
S83C752–4DB	S87C752–4DB	OTP	0 to +70, 28-pin Plastic Shrink Small Outline Package	3.5 to 16MHz	SOT341-1
S83C752–4N28	S87C752–4N28	OTP	0 to +70, 28-pin Plastic Dual In-line Package	3.5 to 16MHz	SOT117-2
S83C752–5N28	S87C752–5N28	OTP	–40 to +85, 28-pin Plastic Dual In-line Package	3.5 to 16MHz	SOT117-2
S83C752–1A28	S87C752–1A28	OTP	0 to +70, 28-pin Plastic Leaded Chip Carrier	3.5 to 12MHz	SOT261-3
S83C752–2A28	S87C752–2A28	OTP	–40 to +85, 28-pin Plastic Leaded Chip Carrier	3.5 to 12MHz	SOT261-3
S83C752–4A28	S87C752–4A28	OTP	0 to +70, 28-pin Plastic Leaded Chip Carrier	3.5 to 16MHz	SOT261-3
S83C752–5A28	S87C752–5A28	OTP	–40 to +85, 28-pin Plastic Leaded Chip Carrier	3.5 to 16MHz	SOT261-3
S83C752–6A28	S87C752–6A28	OTP	–55 to +125, 28-pin Plastic Leaded Chip Carrier	3.5 to 12MHz	SOT261-3
S83C752–6N28	S87C752–6N28	OTP	–55 to +125, 28-pin Plastic Dual In-line Package	3.5 to 12MHz	SOT117-2

NOTE:

1. OTP = One Time Programmable EPROM.

BLOCK DIAGRAM



CMOS single-chip 8-bit microcontrollers — 83C754/87C754

DESCRIPTION

The Philips 83C754/87C754 offers many of the advantages of the 80C51 architecture in a small package and at low cost.

The 8XC754 Microcontroller is fabricated with Philips high-density CMOS technology. Philips epitaxial substrate minimizes CMOS latch-up senitivity.

The 8XC754 contains a 4k × 8 ROM (83C754) EPROM (87C754), a single module PCA, a 256 × 8 RAM, 11 I/O lines, two 16-bit counter/timers, a two-priority level interrupt structure, a full duplex serial channel, an on-chip oscillator, and an 8-bit D/A converter.

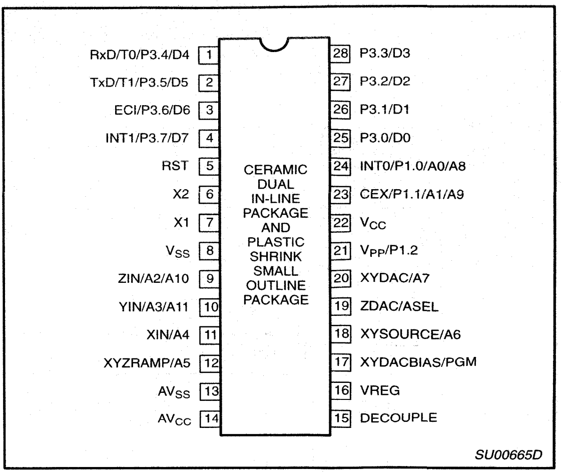
The EPROM version of this device, the 87C754, is available in plastic one-time programmable (OTP) packages. Once the array has been programmed, it ifs functionally equivalent to the masked ROM 83C754. Thus, unless explicitly stated otherwise, all references made to the 87C754 apply equally to the 83C754.

The 8XC754 supports two power reduction modes of operation referred to as the idle mode and the power-down mode.

FEATURES

- Available in erasable quartz lid or One-Time Programmable plastic packages
- 80C51-based architecture
- Small package sizes – 28-pin SSOP
- Wide oscillator frequency range
- Power control modes:
  - Idle mode
  - Power-down mode
- 4k × 8 ROM (83C754)
  EPROM (87C754)
- 256 × 8 RAM
- Two 16-bit auto reloadable counter/timers
- Single module PCA counter/timer
- Full duplex serial channel
- Boolean processor
- CMOS and TTL compatible

PIN CONFIGURATION

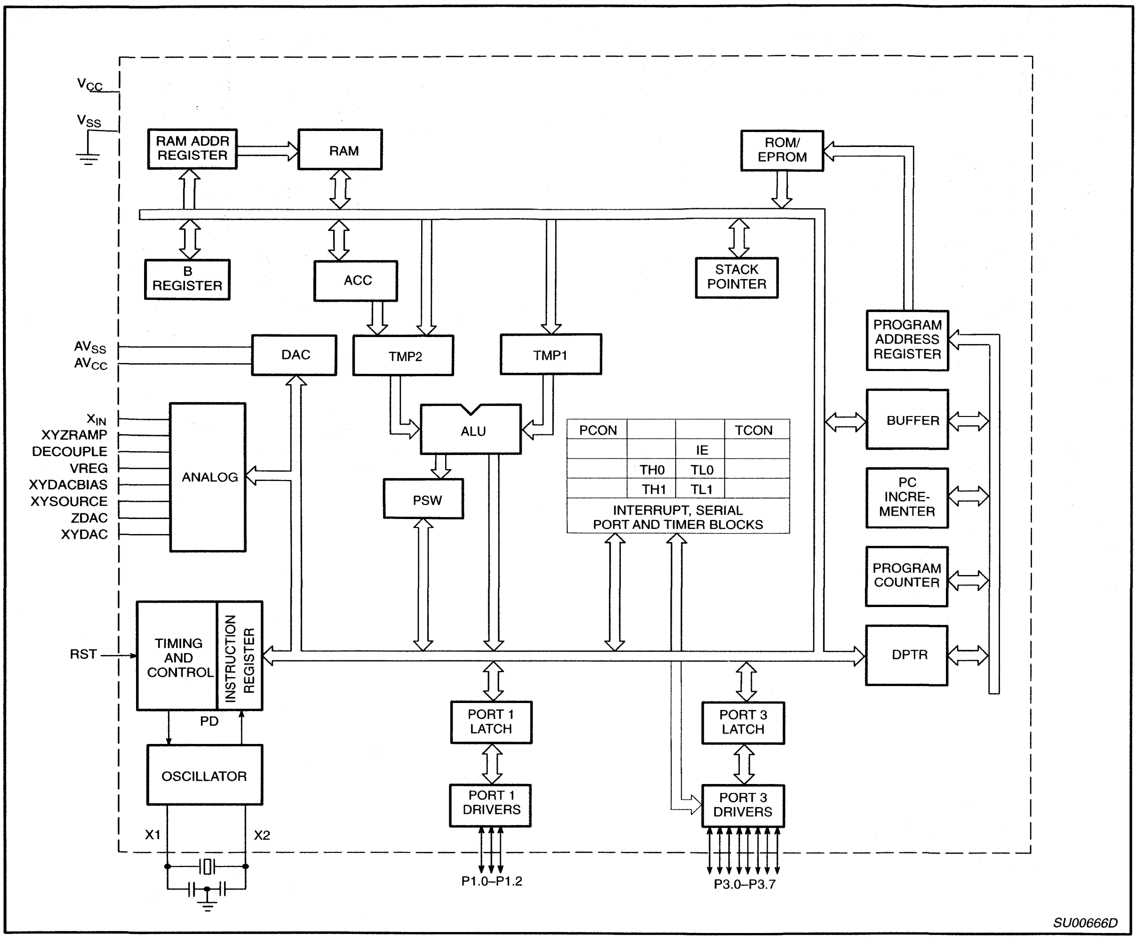


PART NUMBER SELECTION

ROM	EPROM[1]		TEMPERATURE RANGE °C AND PACKAGE	FREQUENCY	DRAWING NUMBER
P83C754EBD DB	P87C754EBD DB	OTP	0 to +70, 28-pin Shrink Small Outline Package	3.5 to 16MHz	SOT341-1

NOTE:
1. OTP = One Time Programmable EPROM. UV = UV Erasable EPROM.

BLOCK DIAGRAM



Microcontroller with TrackPoint™ microcode from IBM — TPM754

The Philips Semiconductors TPM754 is a small package, low cost, ROM-coded 80C51 with IBM®'s TrackPoint™ pointing algorithms and control code. TrackPoint is the result of years of human factors research and innovation at IBM. The result is a "velocity sensitive" pointing solution more efficient and easier to use than "position sensitive" devices such as the mouse, the trackball, or the touchpad.

IBM has licensed Philips Semiconductors to sell microcontrollers with TrackPoint code. By purchasing a TPM from Philips, the purchaser becomes a sub-licensee of Philips. The selling price of Philips' TPM includes the royalties for IBM's intellectual property, which Philips in turn pays to IBM. Customers for TPMs do not need to sign any licensing agreement with either IBM or Philips. This code is the intellectual property of IBM, which is covered by numerous patents, and must be treated accordingly.

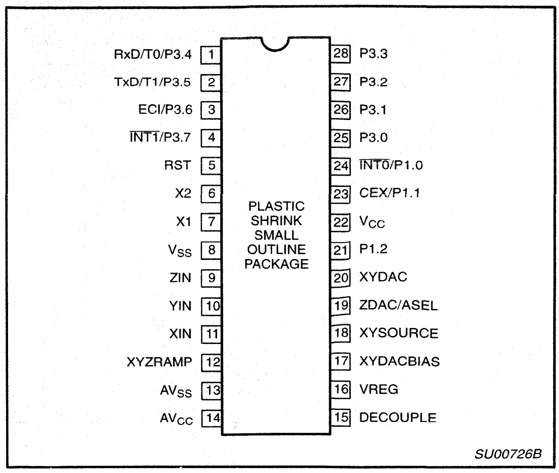
The TPM754 contains IBM® TrackPoint™ code, a single module PCA, a 256 × 8 RAM, 21 I/O lines, two 16-bit counter/timers, a two-priority level interrupt structure, a full duplex serial channel, an on-chip oscillator, and an 8-bit D/A converter.

For identical device without TrackPoint code, see the 8XC754 datasheet.

FEATURES

- 80C51-based architecture
- Small package sizes – 28-pin SSOP
- Power control modes:
  - Idle mode
  - Power-down mode
- 256 × 8 RAM
- Two 16-bit auto reloadable counter/timers
- Single module PCA counter/timer
- Full duplex serial channel
- Boolean processor
- CMOS and TTL compatible

PIN CONFIGURATION



ORDERING INFROMATION

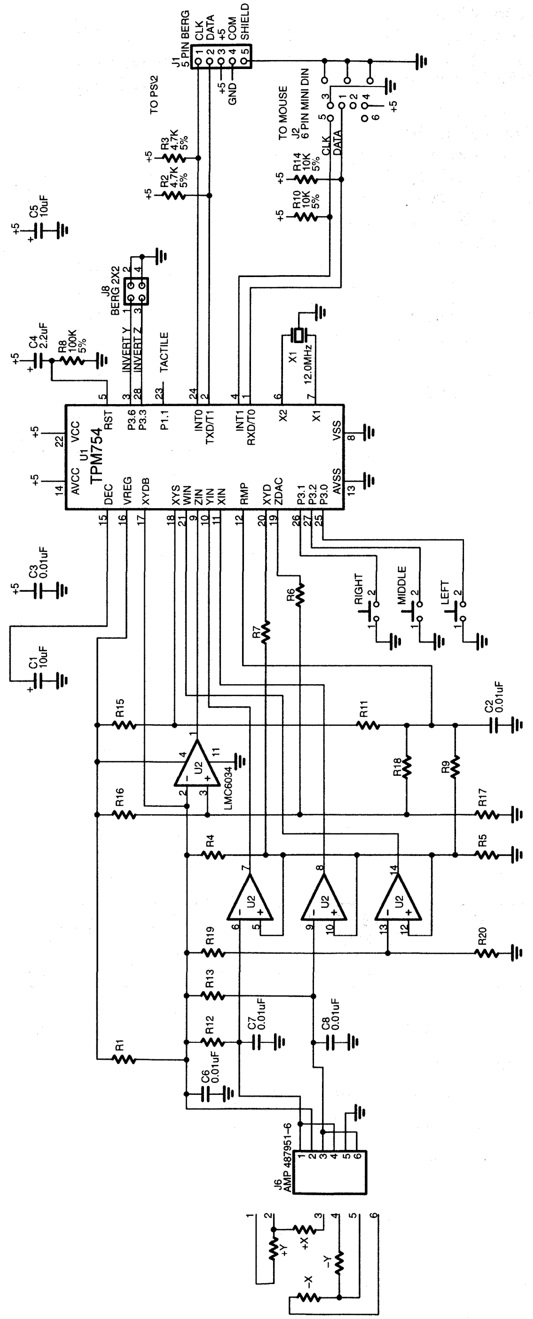
ORDERING CODE	TEMPERATURE RANGE °C AND PACKAGE	FREQUENCY	DRAWING NUMBER
PTPM754 DB	0 to +70, 28-pin Shrink Small Outline Package	3.5 to 12MHz	SOT341-1

For compatible pointing device, contact:

COMPANY
Bokam Engineering
CTS Corporation

IBM is a registered trademark, and TrackPoint is a trademark of IBM Corporation.

SCHEMATIC OF TRACKPOINT SYSTEM



NOTES:
1. Resistor values are dependent on the pointing stick used. The pointing stick manufacturer or Philips Semiconductors can provide these values.
2. R1 may be integrated into some pointing sticks.

SU00815D

Low voltage 8-bit microcontrollers with UART and I^{2}C-bus

P83CL781; P83CL782

1 FEATURES

- Full static 80C51 CPU
- 8-bit CPU, ROM, RAM, I/O in a 40-lead DIP or 44-lead QFP package
- 16 kbytes ROM, expandable externally to 64 kbytes
- 256 bytes RAM, expandable externally to 64 kbytes
- Four 8-bit ports, 32 I/O lines
- Three 16-bit timer/event counters
- External memory expandable up to 128 kbytes: RAM up to 64 kbytes and ROM up to 64 kbytes
- On-chip oscillator suitable for RC, LC, quartz crystal or ceramic resonator
- Fifteen source, fifteen vector interrupt structure with two priority levels
- Full duplex serial port (UART)
- I^{2}C-bus interface for serial transfer on two lines
- Enhanced architecture with:
  - non-page oriented instructions
  - direct addressing
  - four 8 byte RAM register banks
  - stack depth limited only by available internal RAM (maximum 256 bytes)
  - multiply, divide, subtract and compare instructions
- Reduced power consumption through Power-down and Idle modes
- Wake-up via external interrupts at Port 1
- Single supply voltage of 1.8 to 6.0 V
- Operating ambient temperature:
  - 83CL781: −40 to +85 °C
  - 83CL782: −25 to +55 °C.
- Frequency range of DC to 12 MHz
- Very low current consumption.

2 GENERAL DESCRIPTION

The term P83CL78x is used throughout this data sheet to refer to both the P83CL781 and P83CL782; differences between the devices are highlighted in the text.

The P83CL78x is manufactured in an advanced CMOS technology. The P83CL78x has the same instruction set as the 80C51, consisting of over 100 instructions: 49 one-byte, 46 two-byte, and 16 three-byte. The device has low power consumption and a wide range of supply voltage; there are two software-selectable modes of reduced activity for further power reduction: Idle and Power-down. For emulation purposes, the P85CL781 (piggy-back version) with 256 bytes of RAM is recommended.

The P83CL782 is a faster version of the P83CL781 and operates at a maximum frequency of 12 MHz at $V_{DD} \geq 3.1$ V.

This data sheet details the specific properties of the P83CL78x. For details of the 80C51 core and the I^{2}C-bus see *"Data Handbook IC20"*.

3 APPLICATIONS

The P83CL78x is an 8-bit general purpose microcontroller especially suited for cordless telephone applications. The P83CL78x also functions as an arithmetic processor having facilities for both binary and BCD arithmetic plus bit-handling capabilities.

4 ORDERING INFORMATION

TYPE NUMBER(1)	PACKAGE		
	NAME	DESCRIPTION	VERSION
P83CL781HFP	DIP40	plastic dual in-line package; 40 leads (600 mil)	SOT129-1
P83CL782HDP			
P83CL781HFH	QFP44	plastic quad flat package; 44 leads (lead length 2.35 mm); body 14 × 14 × 2.2 mm	SOT205-1
P83CL782HDH			
P83CL781HFH	QFP44	plastic quad flat package; 44 leads (lead length 1.3 mm); body 10 × 10 × 1.75 mm	SOT307-2
P83CL781HDH			

Note

1. Refer to the Order Entry Form (OEF) for this device for the full type number, including options/program.

5 BLOCK DIAGRAM

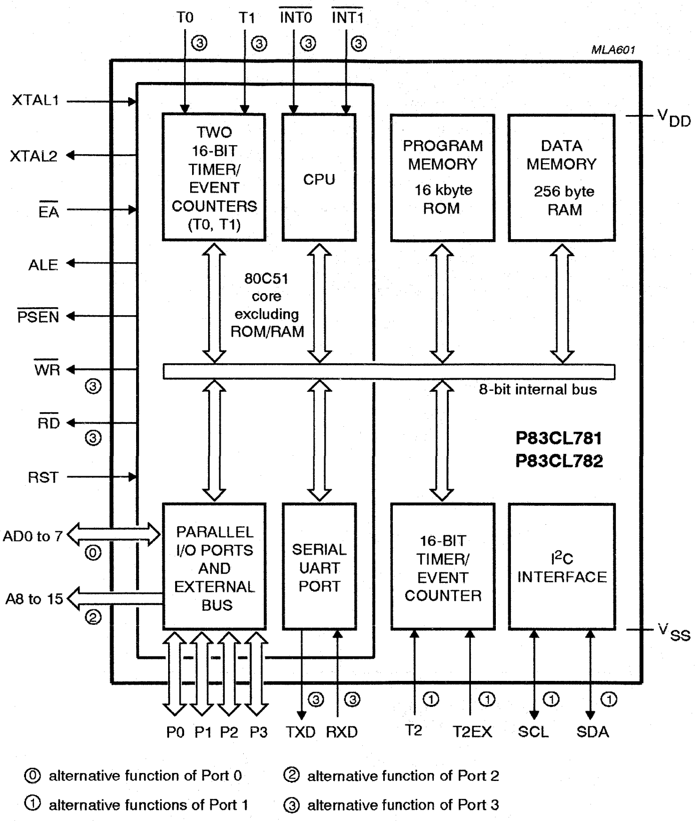

T0
T1
INT0
INT1
③
MLA601
XTAL1
XTAL2
EA
ALE
PSEN
WR
RD
RST
AD0 to 7
A8 to 15
⓪
②
TWO 16-BIT TIMER/ EVENT COUNTERS (T0, T1)
CPU
PROGRAM MEMORY 16 kbyte ROM
DATA MEMORY 256 byte RAM
V_DD
80C51 core excluding ROM/RAM
8-bit internal bus
P83CL781
P83CL782
PARALLEL I/O PORTS AND EXTERNAL BUS
SERIAL UART PORT
16-BIT TIMER/ EVENT COUNTER
I²C INTERFACE
V_SS
①
P0 P1 P2 P3
TXD RXD
T2 T2EX
SCL SDA
⓪ alternative function of Port 0
① alternative functions of Port 1
② alternative function of Port 2
③ alternative function of Port 3

Fig.1 Block diagram.

CMOS single-chip 8-bit microcontroller with on-chip EEPROM

80C851/83C851

DESCRIPTION

The Philips 80C851/83C851 is a high-performance microcontroller fabricated with Philips high-density CMOS technology. The 80C851/83C851 has the same instruction set as the 80C51. The Philips CMOS technology combines the high speed and density characteristics of HMOS with the low power attributes of CMOS. The Philips epitaxial substrate minimizes latch-up sensitivity.

The 80C851/83C851 contains a 4k × 8 ROM with mask-programmable ROM code protection, a 128 × 8 RAM, 256 × 8 EEPROM, 32 I/O lines, two 16-bit counter/timers, a seven-source, five vector, 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 clock circuits.

In addition, the 80C851/83C851 has two software selectable modes of power reduction — idle mode and power-down mode. 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 and EEPROM contents but freezes the oscillator, causing all other chip functions to be inoperative.

FEATURES

- 80C51 based architecture
  - 4k × 8 ROM
  - 128 × 8 RAM
  - Two 16-bit counter/timers
  - Full duplex serial channel
  - Boolean processor
- Non-volatile 256 × 8-bit EEPROM (electrically erasable programmable read only memory)
  - On-chip voltage multiplier for erase/write
  - 50,000 erase/write cycles per byte
  - 10 years non-volatile data retention
  - Infinite number of read cycles
  - User selectable security mode
  - Block erase capability
- Mask-programmable ROM code protection
- Memory addressing capability
  - 64k ROM and 64k RAM
- Power control modes:
  - Idle mode
  - Power-down mode
- CMOS and TTL compatible
- 1.2 to 16MHz
- Three package styles
- Three temperature ranges
- ROM code protection

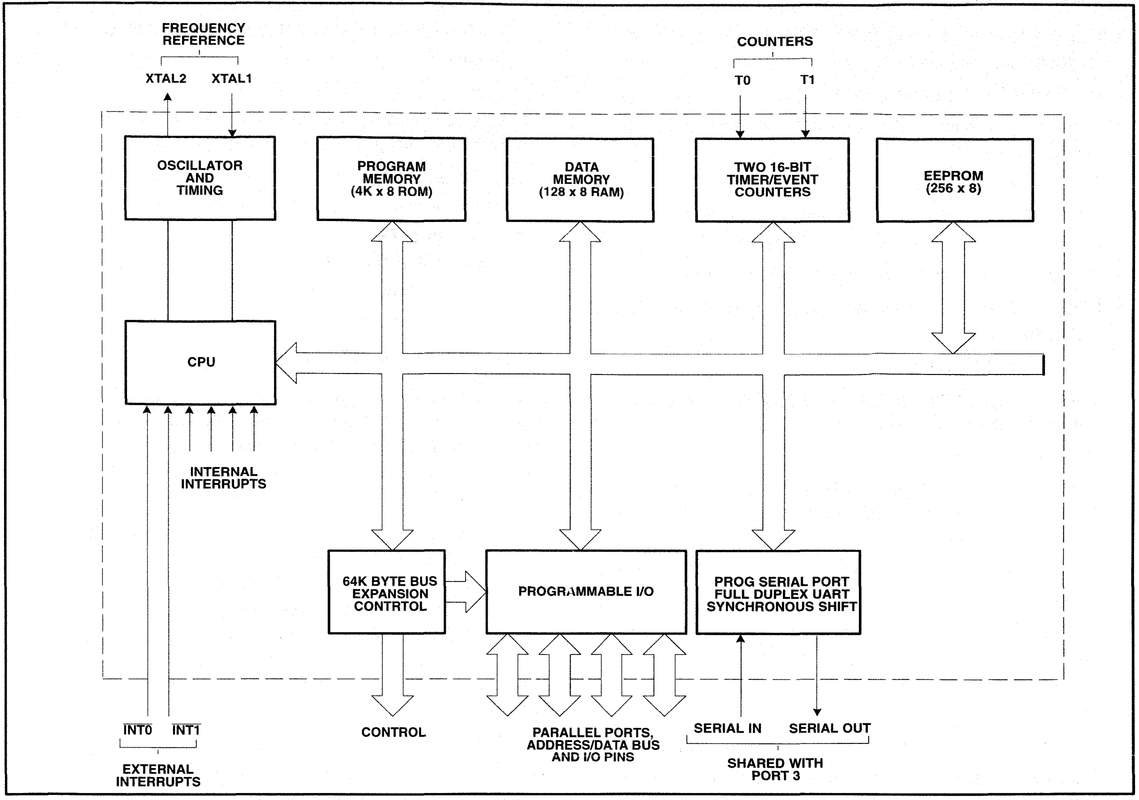
ORDERING INFORMATION

PHILIPS PART ORDER NUMBER PART MARKING		NORTH AMERICA PHILIPS PART ORDER NUMBER		TEMPERATURE RANGE °C AND PACKAGE	FREQ. (MHz)	DRAWING NUMBER
ROMless Version	ROM Version	ROMless Version	ROM Version			
P80C851 FBP	P83C851 FBP	S80C851-4N40	S83C851-4N40	0 to +70, Plastic Dual In-line Package	1.2 to 16	SOT129-1
P80C851 FBA	P83C851 FBA	S80C851-4A44	S83C851-4A44	0 to +70, Plastic Leaded Chip Carrier	1.2 to 16	SOT187-1
P80C851 FBB	P83C851 FBB	S80C851-4B44	S83C851-4B44	0 to +70, Plastic Quad Flat Pack	1.2 to 16	SOT307-2 [1]
P80C851 FFP	P83C851 FFP	S80C851-5N40	S83C851-5N40	–40 to +85, Plastic Dual In-line Package	1.2 to 16	SOT129-1
P80C851 FFA	P83C851 FFA	S80C851-5A44	S83C851-5A44	–40 to +85, Plastic Leaded Chip Carrier	1.2 to 16	SOT187-1
P80C851 FFB	P83C851 FFB	S80C851-5B44	S83C851-5B44	–40 to +85, Plastic Quad Flat Pack	1.2 to 16	SOT307-2 [1]
P80C851 FHP	P83C851 FHP	S80C851-6N40	S83C851-6N40	–40 to +125, Plastic Dual In-line Package	1.2 to 16	SOT129-1
P80C851 FHA	P83C851 FHA	S80C851-6A44	S83C851-6A44	–40 to +125, Plastic Leaded Chip Carrier	1.2 to 16	SOT187-1
P80C851 FHB	P83C851 FHB	S80C851-6B44	S83C851-6B44	–40 to +125, Plastic Quad Flat Pack	1.2 to 16	SOT307-2 [1]

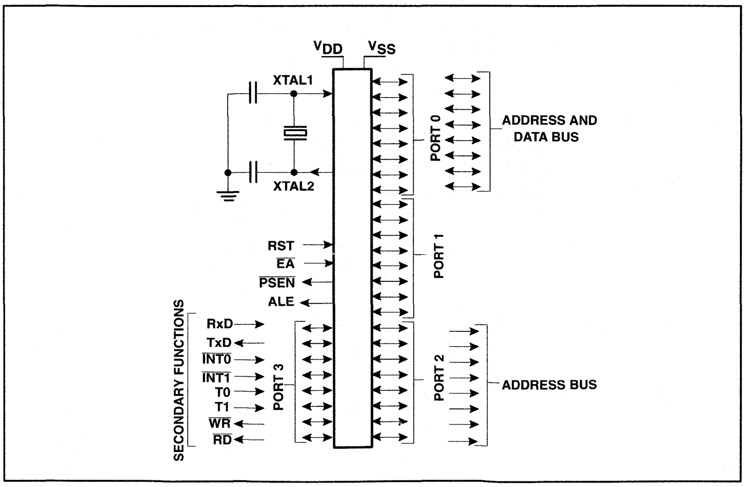
NOTE:

1. SOT311 replaced by SOT307-2.

BLOCK DIAGRAM


FREQUENCY REFERENCE
XTAL2
XTAL1
COUNTERS
T0
T1
OSCILLATOR AND TIMING
PROGRAM MEMORY (4K x 8 ROM)
DATA MEMORY (128 x 8 RAM)
TWO 16-BIT TIMER/EVENT COUNTERS
EEPROM (256 x 8)
CPU
INTERNAL INTERRUPTS
64K BYTE BUS EXPANSION CONTRTOL
PROGRAMMABLE I/O
PROG SERIAL PORT FULL DUPLEX UART SYNCHRONOUS SHIFT
INT0
INT1
EXTERNAL INTERRUPTS
CONTROL
PARALLEL PORTS, ADDRESS/DATA BUS AND I/O PINS
SERIAL IN
SERIAL OUT
SHARED WITH PORT 3

LOGIC SYMBOL



1 FEATURES

- Full static 80C51 CPU (8-bit CPU) with a minimum 6 clocks per instruction
- OTP/ROM program memory
- RAM, expandable externally to 64 kbytes (only on certain devices)
- DTMF generator
- MSK modem including Manchester encoder/decoder for analog cordless telephones (standards CT0/CT1/CT1+)
- Pulse Width Modulated output (8-bit resolution)
- EEPROM data memory, accessed internally via I^{2}C-bus interface
- 8-bit ports, I/O lines
- Three 16-bit timer/event counters, including one with capture, compare and PWM function
- Watchdog Timer
- External memory expandable up to 128 kbytes external ROM up to 64 kbytes and/or RAM up to 64 kbytes (only possible on certain devices)
- On-chip amplitude controlled oscillator (ACO) suitable for quartz crystal or ceramic resonator
- 32 kHz Real-Time Clock (RTC) with programmable interrupt periods
- Twenty source, twenty vector interrupt structure with two priority levels
- Full duplex enhanced UART with double buffering
- I^{2}C-bus interface for 2-wire serial transfer, 400 kHz maximum
- Enhanced architecture with:
  - Non-page oriented instructions
  - Direct addressing
  - Four 8 byte RAM register banks
  - Stack depth limited only by available internal RAM (maximum 256 bytes)
  - Multiply, divide, subtract and compare instructions
- Eight additional interrupts on Port 1
  - Edge or level sensitive triggering selectable via software
  - Power-saving use for keyboard control.
- Improved Power-on/Power-off reset circuitry (POR) with 9 hardware programmable levels
- Low Voltage Detection (LVD) with 11 software programmable levels
- Wake-up from Power-down mode via external interrupts at Port 1, via RTC or via LVD
- Very low current consumption.

2 GENERAL DESCRIPTION

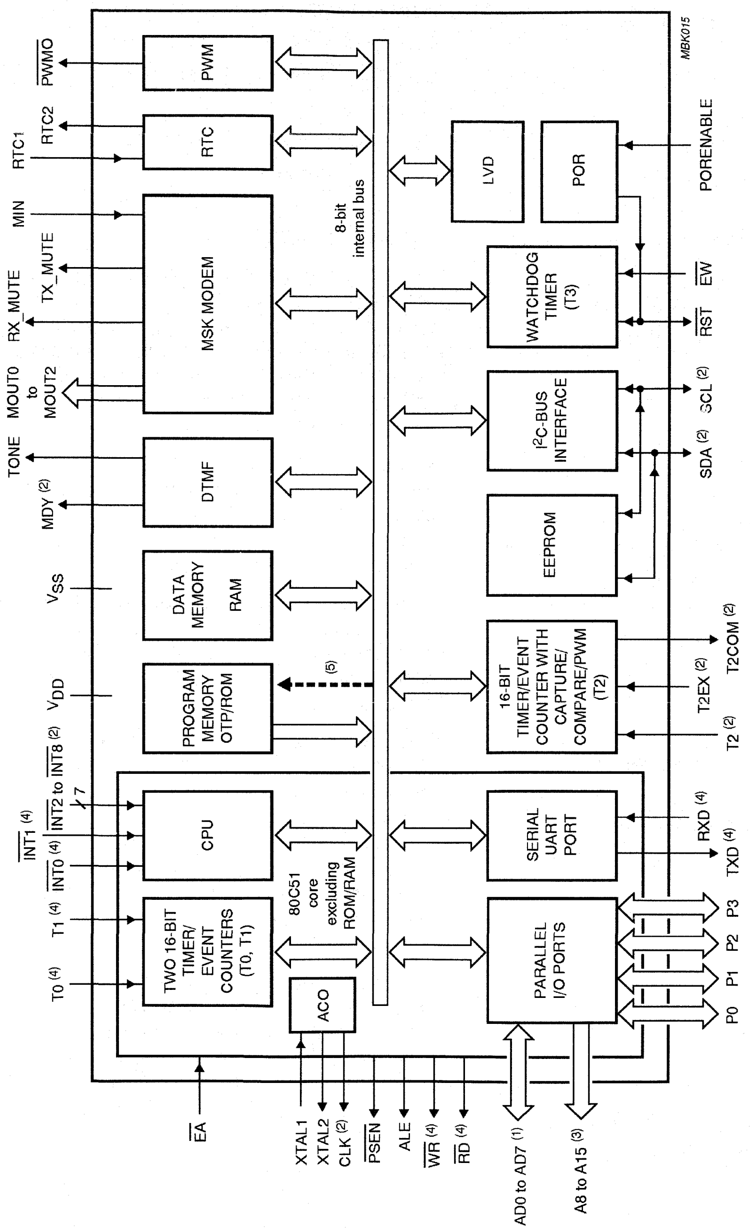
The TELX microcontroller family is manufactured in an advanced CMOS technology and is based on MCM (Multi-Chip-Module) technology as the non-volatile memory parts OTP and EEPROM are integrated on a separate chip.

The TELX family are 8-bit microcontrollers especially suited for analog cordless telephones (CT0, CT1, CT1+ standards), mid/high-end corded telephones and pagers. For this purpose, features like DTMF, EEPROM, MSK modem, PWM, POR/LVD, ACO and RTC are integrated on-chip. The device is optimized for low power consumption. The TELX family has two software selectable features for power reduction: Idle and Power-down modes.

The instruction set of the TELX family is based on that of the 8051. The TELX family 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: 49 one-byte, 46 two-byte, and 16 three-byte.

This data sheet details the shared properties of the TELX family. For a particular microcontroller, read this data sheet in conjunction with the individual data sheet of the specific device. For details on the I^{2}C-bus functions see *"Data Handbook IC12"*.

3 BLOCK DIAGRAM



(1) alternative function of Port 0
(2) alternative function of Port 1
(3) alternative function of Port 2
(4) alternative function of Port 3
(5) in-circuit OTP programming

Fig.1 Block diagram.

TELX microcontrollers for CTO handset/basestation applications

P83CL883; P87CL883; P83CL884; P87CL884

1 FEATURES

- Full static 80C51 CPU; enhanced 8-bit architecture with:
  - minimum 6 cycles per instruction (twice as fast as a standard 80C51 core)
  - non-page oriented instructions
  - direct addressing
  - four 8-byte RAM register banks
  - stack depth limited only by available internal RAM (maximum 256 bytes)
  - multiply, divide, subtract and compare instructions
- 8-bit ports:
  - P83CL883 and P87CL883: 3 (19 I/O lines)
  - P83CL884 and P87CL884: 3 (18 I/O lines)
- Program Memory:
  - P87CL883/P87CL884: 8 kbytes One Time Programmable (OTP)
  - P83CL883/P83CL884: 8 kbytes ROM
- 256 bytes RAM
- 128 bytes EEPROM Data Memory, accessed internally via I^{2}C-bus interface (P83CL884 and P87CL884 only)
- Amplitude Controlled Oscillator (ACO) suitable for quartz crystal or ceramic resonator
- Improved Power-on/Power-off reset (POR) circuitry
- Low Voltage Detection (LVD) with 11 software programmable levels
- Eight interrupts on Port 1:
  - edge or level sensitive triggering selectable via software
  - power-saving use for keyboard control
- Twenty source, twenty vector interrupt structure with two priority levels
- Wake-up from Power-down mode via LVD or external interrupts at Port 1
- DTMF generator (P83CL884/P87CL884 only)
- MSK modem including Manchester encoder/decoder with 2 digital outputs for analog cordless telephones (standards CT0/CT1/CT1+)
- Two standard 16-bit timer/event counters
- Additional 16-bit timer/event counter with Capture, Compare and Auto-reload function
- Watchdog Timer
- Full duplex enhanced UART with double buffering
- I^{2}C-bus interface for serial transfer on two lines, maximum 400 kHz
- Very low current consumption
- Single supply voltage: 2.7 to 3.6 V
- Frequency range: 3.58 MHz
- Operating temperature: –25 to +70 °C
- 28 pins SO package.

2 GENERAL DESCRIPTION

The P8xCL883/P8xCL884 - denoting the P83CL883; P87CL883; P83CL884 and P87CL884 - are manufactured in an advanced CMOS technology. The P83CL883 and P87CL883 are based on single chip technology and the P83CL884 and P87CL884 are based on MCM (Multi-Chip-Module) technology as the EEPROM is integrated on a separate chip.

The P8xCL883/P8xCL884 are 8-bit microcontrollers especially suited for low cost analog cordless telephone applications (CT0, CT1, CT1+ standards). For this purpose, features like DTMF, EEPROM, MSK modem and POR/LVD are integrated on-chip.

The device is optimized for low power consumption. The P8xCL883/P8xCL884 have two software selectable features for power reduction: Idle and Power-down modes. In addition, all derivative blocks can switch off their clock if they are inactive.

The instruction set of the P8xCL883/P8xCL884 is based on that of the 8051. The P8xCL883/P8xCL884 also function as an arithmetic processor having facilities for both binary and BCD arithmetic plus bit-handling capabilities. The instruction set consists of over 100 instructions: 49 one-byte, 46 two-byte, and 16 three-byte. Due to the missing port P2, there is no external data or memory access and the MOVX operations cannot be used.

This data sheet details the specific properties of the P8xCL883/P8xCL884; for details of the P8xCL883/P8xCL884 core and the derivative functions see the *“TELX family”* data sheet and *“Data Handbook IC20; 80C51-based 8-bit Microcontrollers”*.

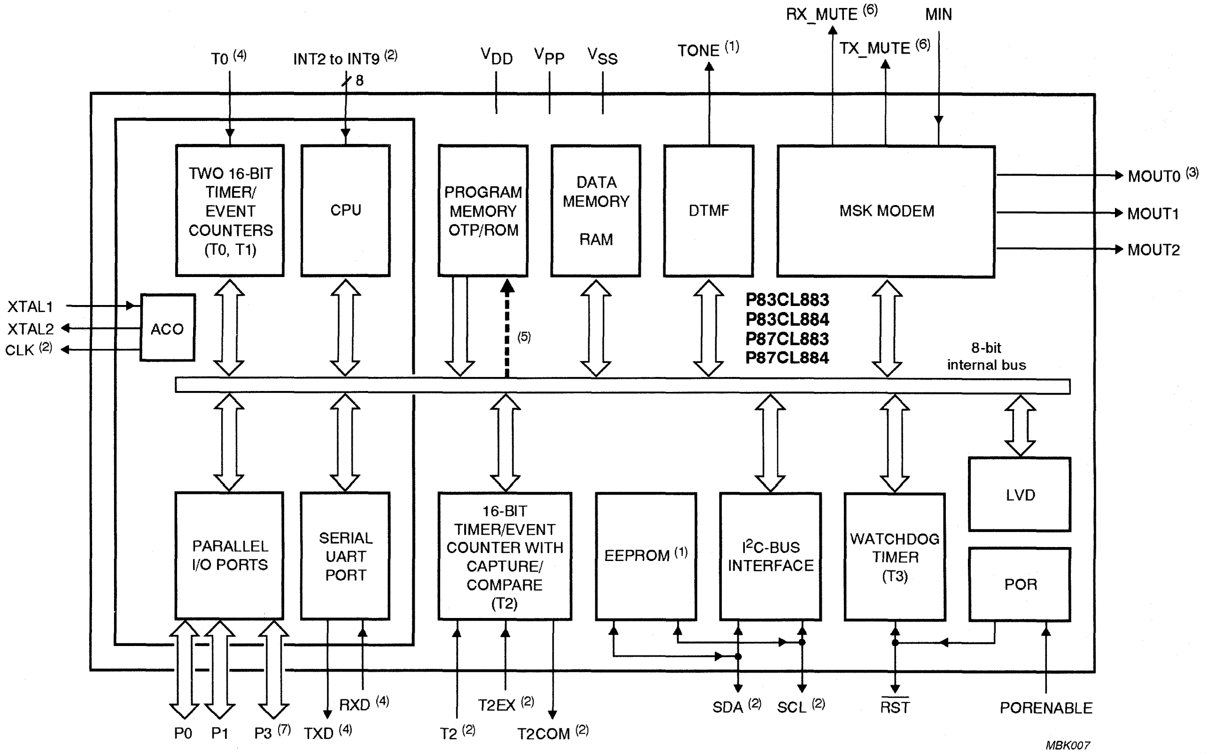
3 ORDERING INFORMATION

TYPE NUMBER	PACKAGE		
	NAME	DESCRIPTION	VERSION
P83CL883DDT	SO28[1]	plastic small outline package; 28 leads; body width 7.5 mm	SOT136-1
P87CL883DDT			
P83CL884DDT			
P87CL884DDT			

Note

1. When using IR reflow soldering it is recommended that the Dry Packing instructions in the *"Quality Reference Pocketbook"* (order number 9398 510 34011) are followed.

4 BLOCK DIAGRAM



(1) Only available on the P83CL884 and P87CL884.

(2) Alternative functions of Port 1.

(3) MOUT0 is the alternative function of P3.1.

(4) Alternative functions of Port 3; T0 is only available on the P83CL883 and P87CL883.

(5) In-circuit OTP programming.

(6) By software, any I/O pin can be used.

(7) Port 3: P3.0, P3.1 and P3.4; P3.4 is only available on the P83CL883 and P87CL883.

Fig.1 Block diagram.

TELX microcontrollers for CT0 handset/basestation applications

P83CL886; P87CL886

1 FEATURES

- Full static 80C51 CPU; enhanced 8-bit architecture with:
  - Minimum 6 cycles per instruction (twice as fast as a standard 80C51 core)
  - Non-page oriented instructions
  - Direct addressing
  - Four 8-byte RAM register banks
  - Stack depth limited only by available internal RAM (maximum 256 bytes)
  - Multiply, divide, subtract and compare instructions
- Three 8-bit ports (18 I/O lines)
- Program Memory:
  - P87CL886: 16 kbytes One Time Programmable
  - P83CL886: 16 kbytes ROM
- 512 bytes RAM
- Amplitude Controlled Oscillator (ACO) suitable for quartz crystal or ceramic resonator
- Improved Power-on/Power-off reset (POR) circuitry
- Low Voltage Detection (LVD) with 11 software programmable levels
- Eight interrupts on Port 1:
  - Edge or level sensitive triggering selectable via software
  - Power-saving use for keyboard control
- Twenty source, twenty vector interrupt structure with two priority levels
- Wake-up from Power-down mode via LVD or external interrupts at Port 1
- DTMF generator
- MSK modem including Manchester encoder/decoder with 2 digital outputs for analog cordless telephones (standards CT0/CT1/CT1+)
- Watchdog Timer
- Two standard 16-bit timer/event counters
- Additional 16-bit timer/event counter with Capture, Compare and Auto-reload function
- Full duplex enhanced UART with double buffering
- I^{2}C-bus interface for serial transfer on two lines, maximum 400 kHz
- Very low current consumption
- Single supply voltage: 2.7 to 3.6 V
- Frequency: 3.58 MHz
- Operating temperature: −25 to +70 °C
- 28 pin SO package.

2 GENERAL DESCRIPTION

The P8xCL886 (denoting the P83CL886 and P87CL886) are manufactured in an advanced CMOS technology and are 8-bit microcontrollers especially suited for low cost analog cordless telephone applications (CT0, CT1 and CT1+ standards). Consequently, features like DTMF, MSK modem and POR/LVD are integrated on-chip.

Both devices are optimized for low power consumption and in addition have two software selectable modes for further power reduction: Idle and Power-down modes. All derivative blocks can switch off their clock if they are inactive.

The instruction set of the P8xCL886 is based on that of the 80C51. The P8xCL886 also function as arithmetic processors having facilities for both binary and BCD arithmetic plus bit-handling capabilities. The instruction set consists of over 100 instructions: 49 one-byte, 46 two-byte, and 16 three-byte. As port P2 is not implemented there is no external data or memory access and MOVX operations cannot be used.

This data sheet details the specific properties of the P8xCL886; for details of the P8xCL886 core and the derivative functions see the *"TELX family"* data sheet and *"Data Handbook IC20"*.

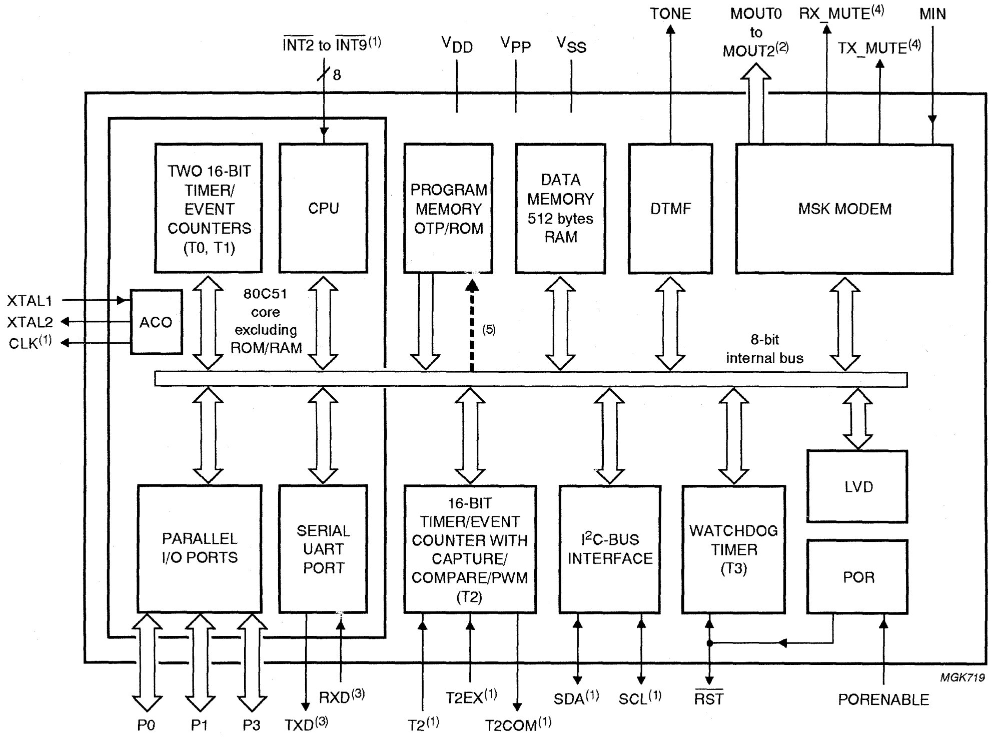
3 ORDERING INFORMATION

TYPE NUMBER	PACKAGE		
	NAME	DESCRIPTION	VERSION
P83CL886DFT	SO28[1]	plastic small outline package; 28 leads; body width 7.5 mm	SOT136-1
P87CL886DFT			

Note

1. When using IR reflow soldering it is recommended that the Dry Packing instructions in the *"Quality Reference Pocketbook"* (order number 9398 510 34011) are followed.

4 BLOCK DIAGRAM



[1] Alternative functions of Port 1.

[2] MOUT0 is the alternative function to P3.1.

[3] Alternative function of Port 3.

[4] By software any I/O port pin can be used.

[5] In-circuit OTP programming.

Fig.1 Block diagram.

TELX microcontrollers for CT0 handset/basestation applications

P83CL887; P87CL887

1 FEATURES

- Full static 80C51 CPU; enhanced 8-bit architecture with:
  - Minimum 6 cycles per instruction (twice as fast as a standard 80C51 core)
  - Non-page oriented instructions
  - Direct addressing
  - Four 8-byte RAM register banks
  - Stack depth limited only by available internal RAM (maximum 256 bytes)
  - Multiply, divide, subtract and compare instructions
- Three 8-bit ports (18 I/O lines)
- Program Memory:
  - P87CL887: 12 kbytes One Time Programmable
  - P83CL887: 12 kbytes ROM
- 512 bytes RAM
- Amplitude Controlled Oscillator (ACO) suitable for quartz crystal or ceramic resonator
- Improved Power-on/Power-off reset (POR) circuitry
- Low Voltage Detection (LVD) with 11 software programmable levels
- Eight interrupts on Port 1:
  - Edge or level sensitive triggering selectable via software
  - Power-saving use for keyboard control
- Twenty source, twenty vector interrupt structure with two priority levels
- Wake-up from Power-down mode via LVD or external interrupts at Port 1
- DTMF generator
- MSK modem including Manchester encoder/decoder with 2 digital outputs for analog cordless telephones (standards CT0/CT1/CT1+)
- Watchdog Timer
- Two standard 16-bit timer/event counters
- Additional 16-bit timer/event counter with Capture, Compare and Auto-reload function
- Full duplex enhanced UART with double buffering
- I^{2}C-bus interface for serial transfer on two lines, maximum 400 kHz
- Very low current consumption
- Single supply voltage: 2.7 to 3.6 V
- Frequency: 3.58 MHz
- Operating temperature: −25 to +70 °C
- 28 pin SO package.

2 GENERAL DESCRIPTION

The P8xCL887 (denoting the P83CL887 and P87CL887) are manufactured in an advanced CMOS technology and are 8-bit microcontrollers especially suited for low cost analog cordless telephone applications (CT0, CT1 and CT1+ standards). Consequently, features like DTMF, MSK modem and POR/LVD are integrated on-chip.

Both devices are optimized for low power consumption and in addition have two software selectable modes for further power reduction: Idle and Power-down modes. All derivative blocks can switch off their clock if they are inactive.

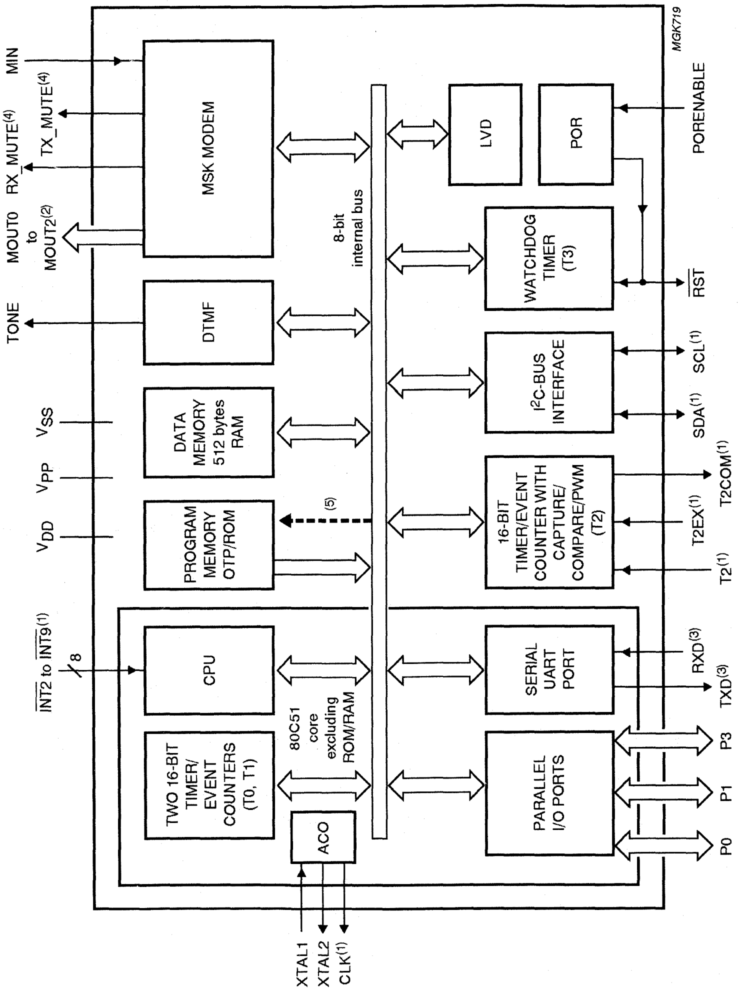
The instruction set of the P8xCL887 is based on that of the 80C51. The P8xCL887 also function as arithmetic processors having facilities for both binary and BCD arithmetic plus bit-handling capabilities. The instruction set consists of over 100 instructions: 49 one-byte, 46 two-byte, and 16 three-byte. As port P2 is not implemented there is no external data or memory access and MOVX operations cannot be used.

This data sheet details the specific properties of the P8xCL887; for details of the P8xCL887 core and the derivative functions see the *"TELX family"* data sheet and *"Data Handbook IC20"*.

3 ORDERING INFORMATION

TYPE NUMBER	PACKAGE		
	NAME	DESCRIPTION	VERSION
P83CL887DFT	SO28	plastic small outline package; 28 leads; body width 7.5 mm	SOT136-1
P87CL887DFT			

4 BLOCK DIAGRAM



(1) Alternative functions of Port 1.

(2) MOUT0 is the alternative function to P3.1.

(3) Alternative function of Port 3.

(4) By software any I/O port pin can be used.

(5) In-circuit OTP programming.

Fig.1 Block diagram.

Low voltage 8-bit microcontroller P87CL881

1 FEATURES

- Full static 80C51 CPU; enhanced 8-bit architecture with:
  - Minimum 6 cycles per instruction (twice as fast as a standard 80C51 core)
  - Non-page oriented instructions
  - Direct addressing
  - Four 8 byte RAM register banks
  - Stack depth limited only by available internal RAM (maximum 256 bytes)
  - Multiply, divide, subtract and compare instructions.
- Very low current consumption
- Single supply voltage of 1.8 to 3.6 V
- Frequency: 1 MHz to 10 MHz
- Operating temperature: –25 to +70 °C
- 44-pin LQFP package
- Four 8-bit ports (32 I/O lines)
- 63 kbytes OTP program memory
- 256 bytes internal RAM
- 1792 bytes internal AUX-RAM
- External address range: 64 kbytes of ROM and 64 kbytes of RAM
- Amplitude Controlled Oscillator (ACO) suitable for use with a quartz crystal or ceramic resonator
- Improved Power-On/Power-Off Reset circuitry (POR)
- Low Voltage Detection (LVD) with 11 software programmable levels
- 8 interrupts on Port 1, edge or level sensitive triggering selectable via software power-saving use for keyboard control
- Twenty source, twenty vector interrupt structure with two priority levels
- Wake-up from Power-down mode via LVD or external interrupts at Port 1
- Two 16-bit timer/event counters
- Additional 16-bit timer/event counters, with capture, compare and PWM function
- Watchdog Timer
- Full duplex enhanced UART with double buffering
- I^{2}C-bus interface for serial transfer on two lines, maximum operating frequency 400 kHz.

2 GENERAL DESCRIPTION

The P87CL881 is an 8-bit microcontroller especially suited for pager applications.

The P87CL881 is manufactured in an advanced CMOS technology and is based on single-chip technology.

The device is optimized for low power consumption and has two software selectable features for power reduction: Idle and Power-down modes. In addition, all derivative blocks switch off their clock if they are inactive.

The instruction set of the P87CL881 is based on that of the 8051. The P87CL881 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: 49 one-byte, 46 two-byte, and 16 three-byte.

This data sheet details the specific properties of the P87CL881; for details of the P87CL881 core and the derivative functions see the *"TELX family"* data sheet and *"8051-Based 8-bit Microcontrollers; Data Handbook IC20"*.

3 ORDERING INFORMATION

TYPE NUMBER(1)	PRODUCT TYPE	PACKAGE		
		NAME	DESCRIPTION	VERSION
P87CL881HDH/000	Blank OTP; note 2	LQFP44	plastic low profile quad flat package; 44 leads; body 10 × 10 × 1.4 mm	SOT389-1
P87CL881HDH/xxx	Pre-programmed OTP			

Notes

1. Please refer to the Order Entry Form (OEF) for this device for the full type number to use when ordering. This type number will also specify the required program and options.
2. No guarantee for OTP retention.

4 BLOCK DIAGRAM

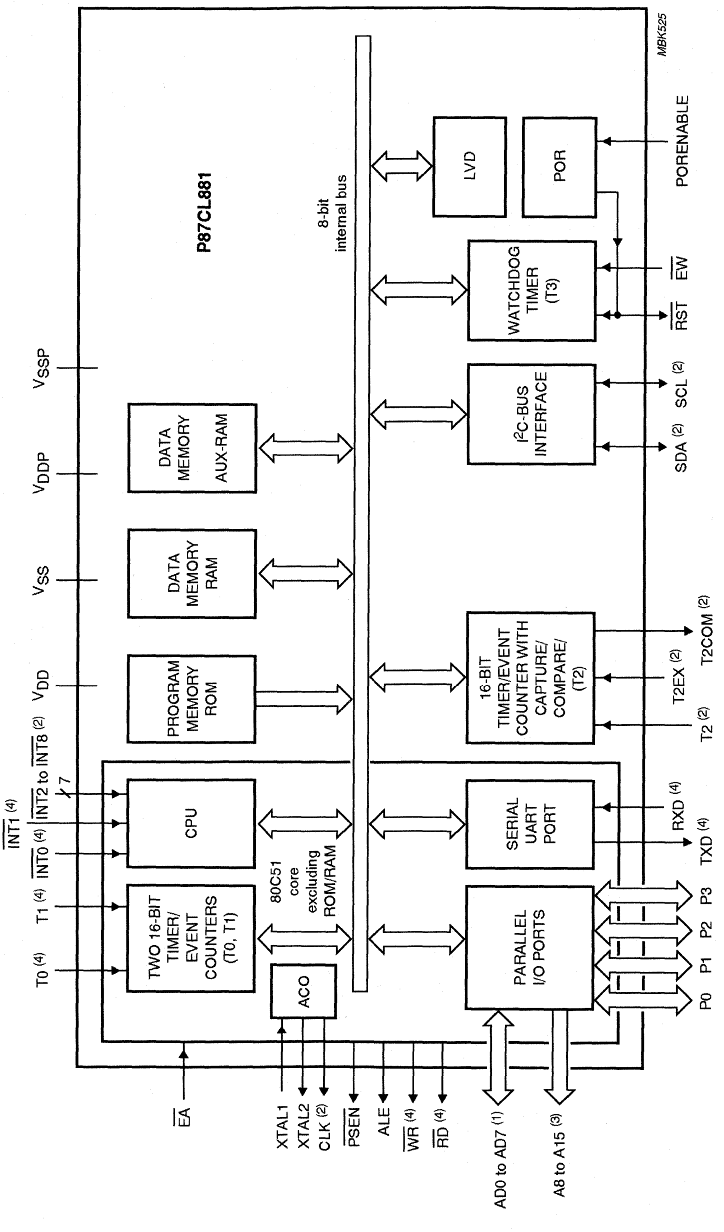


Fig.1 Block diagram.

(1) Alternative function of Port 0.
(2) Alternative function of Port 1.
(3) Alternative function of Port 2.
(4) Alternative function of Port 3.

Section 4
High Performance 16-bit 80C51 XA (eXtended Architecture)

CONTENTS

	80C51XA Architectural overview	(See CDROM)
XA-G1	CMOS single-chip 16-bit microcontroller	(See CDROM)
XA-G2	CMOS single-chip 16-bit microcontroller	(See CDROM)
XA-G3	CMOS single-chip 16-bit microcontroller	(See CDROM)
XA-S3	Single-chip 16-bit microcontroller	(See CDROM)

Section 5

Inter–Integrated Circuit (I^2C) Bus

CONTENTS

The I^2C-bus and how to use it (including specifications) ..... (See CDROM)
I^2C peripheral selection guide .... (See CDROM)
82B715 I^2C bus extender .... (See CDROM)

Section 6
I^{2}C Serial Bus Application Notes & Articles

CONTENTS

AN422	Using the 8XC751 microcontroller as an I^{2}C bus master .	253
AN425	Interfacing the PCF8584 I^{2}C-bus controller to 80C51 family microcontrollers	253
AN430	Using the 8XC751/752 in multimaster I^{2}C applications . .	253
AN433	I^{2}C slave routines for the 83C751	254
AN434	Connecting a PC keyboard to the I^{2}C-bus	254
AN435	I^{2}C byte oriented system driver	254
AN438	I^{2}C routines for 8XC528	255
AN444	Using the P82B715 I^{2}C extender on long cables	255
AN452	One mile long I^{2}C communication using the P82B715	255
EIE/AN91007	I^{2}C driver routines for 8XC751/2 microcontrollers	256
ETV/AN89004	PLM51 I^{2}C software interface IIC51 (version 0.5)	256

Using the 8XC751 microcontroller as an I²C bus master — AN422

Starts with an overview of I²C basics including functions of master & slave, data transfers, addressing and transfer formats and use of sub–addresses. Next, the I²C hardware features of the 8XC751 are described including control, data and configuration registers and Timer I. Finally, a single–master ASM programming example is presented which includes send and receive routines with error recovery.

Interfacing the PCF8584 I²C–bus controller to 80C51 family microcontrollers — AN425

Describes the PCF8584 standalone ¹²C bus interface IC including pin description, block diagram and schematic illustrating the connection to 80C51 family MCUs. A variety of PCF8584 driver routines are described including initialization, data transfer and error handling. An ASM programming example uses these routines to demonstrate operation with a selection of I²C peripheral ICs including the PCD8577 LCD driver, PCF8574P I/O expander, PCF8583P real–time clock and PCF8591 A/D converter.

Using the 8XC751/752 in multimaster I²C applications — AN430

Extends the basic concepts presented in AN422 ("Using the 8XC751 microcontroller as an I²C bus master") with special focus on multi–master configurations including the arbitration mechanism and handshake by clock synchronization. A description of various software routines is followed by an ASM example illustrating their use in a multi–master configuration with bus fault detection and recovery.

I^{2}C slave routines for the 83C751 AN433

Author: Greg Goodhue

Presents short and simple I^{2}C software routines that support only slave (rather than master or master & slave) operation and an ASM demonstration program. The slave–only software in this app note complements the master mode software presented in AN422 ("Using the 8XC751 microcontroller as an I^{2}C bus master").

Connecting a PC keyboard to the I^{2}C–bus AN434

Illustrates the connection of a standard PC/AT keyboard to the I^{2}C bus via an 8XC751. Starts with a description of the PC/AT keyboard clock serial protocol and schematic of the keyboard interface to the MCU. An ASM software example reads data from the keyboard under interrupt control and handles scan–code to ASCII translation using a lookup table.

I^{2}C byte oriented system driver AN435

Presents a very versatile turnkey driver routine for byte oriented I^{2}C interfaces on microcontrollers such as the 8XC552, 8XC562, 8XC652, 8XC654, and others. The purpose of this driver is to make the low level operations of the I^{2}C bus transparent to the user. When this driver is included in an application program, it is controlled via command structures containing simple macro directives.

I^{2}C routines for 8XC528 AN438

Philips Semiconductors Application Note EIE/AN90015

Starts by describing the 'bit–level' I^{2}C interface provided on the 8XC528 MCU including data, control, status and interrupt SFRs. Next, the functions in a library of multi–master capable routines are described at a high level, followed by a slave–only routine written in ASM. Finally, software examples in ASM, PL/M–51 and 'C' demonstrate use of the library routines to access a PCF8583P real–time clock and PCF8577 driven LCD.

Using the P82B715 I^{2}C extender on long cables AN444

Author: Don Sherman, Sunnyvale

The P82B715 is a buffer/amplifier IC designed to extend the range of the full–speed (100 kHz) I^{2}C bus to 50m and beyond. This app note illustrates use of the chip and evaluates the specifications and test results for a variety of cable types (ex: audio, twisted pair, ribbon). The test configuration utilizes the software from AN430 ("Using the 8XC751/752 in multimaster I^{2}C applications").

One mile long I^{2}C communication using the P82B715 AN452

Author: Don Sherman, Applications Engineer

Expands on AN444 ("Using the P82B715 I2C extender on long cables") by exploring alternatives that allow greatly extending the range of I^{2}C. Describes the problems posed by interference, voltage drop & spikes and capacitance. Discusses and verifies alternative solutions including reduced data rate, minimum stub–length topology and an innovative power distribution scheme relying on a local storage capacitor at each node.

I^2C driver routines for 8XC751/2 microcontrollers — AN91007

Author: J.C.P.M. Pijnenburg, Eindhoven

Contains a complete listing and description of an I^2C library written in ASM for 8XC751/2 microcontrollers. The library is built around a message handler which invokes the various routines providing high–level access including error detection and automatic retries. Examples in ASM, PLM–51 and 'C' demonstrate use of the library with the PCF8583P real–time clock/calendar and PCF8577 LCD driver ICs.

PLM51 I^2C software interface IIC51 (version 0.5) — AN89004

Author: R.C.J. Brink, Eindhoven

Documents the PLM–51 interface to a library of ASM I^2C routines including initialization, data transfer, error detection & automatic retry and bus fault detection & recovery. The resource requirements (ex: code and data size) for both master & slave versions of the library are explained as well as rules and constraints for application specific modifications.

Section 7
87C750, 8XC751, 8XC752 Application Notes

CONTENTS

AN423	Software driven serial communication routines for the 83C751 and 83C752 microcontrollers	259
AN426	Controlling air core meters with the 87C751 and SA5775	259
AN427	Timer I for the 83/87C748/749 and the 83/87C751/752 (non-I^{2}C applications) microcontrollers	259
AN428	Using the ADC and PWM of the 83C752/87C752	260
AN429	Airflow measurement using the 83/87C752 and "C"	260
AN436	"Opti-Mizer" power management for notebook computers using the 8XC752 microcontroller	260
AN439	87C751 fast NiCad charger	261
AN442	(BCM) 87C751 Specification for a bus-controlled monitor	261
AN446	A software duplex UART for the 751/752	261
AN453	Using the 87C751 microcontroller to gang program PCF8582/PCF8581 EEPROMs	262
AN454	Interfacing the 83C576/87C576 to the ISA bus	262
AN459	8XC750 watering controller	262
EIE/AN91007	I^{2}C driver routines for 8XC751/2 microcontrollers	(See Section 6)

Software driven serial communication routines for the 83C751 and 83C752 microcontrollers

AN423

Discusses issues related to software implementation of UARTs using general purpose I/O (i.e. 'bit–banging') for MCUs, including the 8XC75X and others, that don't have built–in UART hardware. An ASM routine implementing a 9600–baud half–duplex UART is described, along with tips for modifications such as changing the baud rate, using a different crystal, etc.

Controlling air core meters with the 87C751 and SA5775

AN426

An air core meter consists of a disc wound with two coils in which the needle position represents the vector sum of each coil's magnetic field. With no need for return springs or zero & centering adjustments, air core meters are widely used in automotive instrument clusters. This application example combines the 8XC751 MCU with an SA5775 Serial Gauge Driver IC. The hardware schematic and ASM demonstration program are described, including a shift & subtract software divide routine.

Timer I for the 83/87C748/749 and the 83/87C751/752 (non–I^{2}c applications) microcontrollers

AN427

Timer I is normally used to support I^{2}C applications, for instance to provide a timeout recovery mechanism from bus faults. However, in non–I^{2}C applications it can also be put to effective use. Two examples are described by way of ASM demonstration programs. The first uses Timer I to generate a periodic interrupt while the second implements a watchdog timer.

Using the ADC and PWM of the 83C752/87C752 AN428

The 8XC752 MCU extends the popular 8XC751 with the addition of a 5–channel (multiplexed) analog to digital converter (ADC) and hardware pulse width modulator (PWM). These functions are exercised with an ASM program that samples all five A/D inputs and outputs the results via RS–232 (using the software UART described in AN423). The program also demonstrates use of the PWM by generating an output waveform with duty cycle corresponding to the voltage present on the ADC input.

Airflow measurement using the 83/87C752 AND "C" AN429

The luxury of programming in a high–level language (HLL) isn't confined to big ticket CPUs. This application note describes an airflow meter design based on the 8XC752 which is completely written in 'C'. The meter incorporates pushbuttons, discrete and 7–segment LEDs, setpoint potentiometer and relay and air velocity, pressure and temperature sensors. Includes schematic, software listing and theory of operation.

"Opti–Mizer" power management for notebook computers using the 8XC752 microcontroller AN436

Illustrates use of the 8XC752, a low–cost MCU with integrated A/D, as a power controller for battery driven systems. In this example, a notebook PC is the vehicle used to illustrate a hierarchical power management regime. However, the techniques described (including activity and battery monitoring, power state transitions, sub–system power switching, clock control, etc.) are useful in any 'green' application.

87C751 fast NiCad charger AN439

Explains the chemistry and charging characteristics of NiCad batteries and describes a delta–peak voltage fast charge algorithm. An example charger design combines the 8XC751 with a UCD3843D–based buck converter current source driven by the MCU pulse width modulator (PWM). For voltage sense, a low–cost, yet high resolution, slope A/D converter matches the unique requirements of the charging algorithm. Includes complete schematic and 'C' software listing.

(BCM) 87C751 specification for a bus–controlled monitor AN442

To accommodate diverse graphics standards, modern CRT monitors automatically adjust various operating parameters, such as horizontal and vertical frequency, to match the incoming video. They also feature 'soft' controls in which user adjustments such as brightness and contrast are set digitally. This app note provides a complete hardware description (including theory of operation, schematic and PCB layout) of a digital monitor controller based on the 8XC751 and I^2C peripherals including the TDA8444 octal DAC and PCF8582 EEPROM.

A software duplex UART for the 751/752 AN446

Greg Goodhue

Significantly enhances the 'bit–banging' UART concept presented in AN423 ("Software driven serial communication routines for the 83C751 and 83C752 microcontrollers") with full– (vs. half–) duplex operation and hardware (RTS/CTS) flow control. Performance is evaluated with a test program that functions as a bidirectional traffic generator. Includes theory of operation and complete ASM routine listing.

Using the 87C751 microcontroller to gang program PCF8582/PCF8581 EEPROMs

AN453

Author: David Chen, Shanghai, Philips Technolory Applications Lab

This application exploits the 8XC751 built–in I^{2}C interface in a 10–socket I^{2}C EEPROM gang programmer. Both the PCF8581(128 byte) and PCF8582 (256 byte) EEPROMs are supported. The programmer features copy and reset pushbuttons, global programming status LEDs and discrete pass/fail LEDs at each socket. Includes schematic, PCB layout and ASM program listing.

Interfacing the 83C576/87C576 to the ISA bus

AN454

The 8XC576 is uniquely well suited for application as a subsystem controller working with a host CPU thanks to it's built–in universal peripheral interface (UPI). The UPI offers ports for bidirectional command & data exchange that accommodates polling and interrupt driven transfers. In this example the host is a PC so the UPI is connected to the ISA bus and shown to meet required timing specs. The hardware schematic is supplemented with software listings for both the host CPU and MCU (in 'C' and ASM respectively).

8XC750 watering controller

AN459

Author: Peious Yoseph/Jean–PhilippeChevreau

Illustrates the use of an 8XC750 in an extremely cost–sensitive application. Besides the microcontroller, the design includes TRIACs for sprinkler valve control and a humidity sensor to avoid over–watering. The user interface is comprised of four pushbuttons and an HP HDSP–2111 monolithic 8 character bit–mapped LED display. The LED display hardware and software interface is examined in detail and a complete schematic is included.

Section 8
80C51 Application Notes & Articles

CONTENTS

AN408 80C451 operation of port 6 265

AN417 256k Centronics printer buffer using the 87C451 microcontroller 265

AN418 Counter/timer 2 of the 83C552 microcontroller 265

AN420 Using up to 5 external interrupts on 80C51 family microcontrollers 266

AN424 8051 family warm boot determinations 266

AN440 RAM loader program for 80C51 family applications 266

AN443 IEEE Micro Mouse using the 87C751 microcontroller 267

AN447 Automatic baud rate detection for the 80C51 267

AN448 Determining baud rates for 8051 UARTs and other UART issues 267

AN449 Measure resistance and capacitance without an A/D 268

AN455 In-circuit programming of the Philips 87C576 microcontroller 268

AN456 Using LC oscillator circuits with Philips microcontrollers 268

AN457 80C51 External Memory Interfacing 269

AN458 Dual data pointers for '51 family 269

EIE/AN91001 Workbench EMC evaluation method 269

EIE/AN91006 A/D conversion with P83CL410 PCF1252-x 270

EIE/AN91009 Driver for 8xC851 E2PROM 270

EIE/AN92001 Low RF-emission applications with a P83CE654 microcontroller 270

EIE/AN93017 Using the analog-to-digital converter of the 8XC552 microcontroller 271

ESG89001 Electro magnetic compatibility and
printed circuit board (PCB) constraints 271

Chips push CAN bus into embedded world 272

Add Text Overlay to Any Video Display 274

80C451 operation of port 6 AN408

Similar to the universal peripheral interface described in AN454, the 80C451 includes an 8–bit port with dedicated input and output strobe, port enable and command/status pins. This application note describes three ways to utilize port 6 including multiprocessor interconnection, standard quasi–bidirectional parallel I/O and Centronics–style printer interface. ASM program examples accompany the description of each alternative.

256k Centronics printer buffer using the 87C451 microcontroller AN417

Building on the Centronics–style printer interface discussed in AN408, this application note presents the complete design (including schematic, flowchart and ASM software listing) for an 87C451–based 256KB printer buffer. The DRAM interface is handled under software control including row/column addressing, RAS/CAS generation and refresh. Possible design enhancements including serial– and multi– printer support are discussed.

Counter/timer 2 of the 83C552 microcontroller AN418

The 8XC552 MCU extends the 8051 base architecture with extra memory, 10–bit A/D, serial & parallel I/O and counter/timers. This application note examines one of the latter, counter/timer 2, in detail. The workings of each major function block (16–bit counter with 3–bit prescaler, input capture and output compare) are described in detail. Operation of the unit is described in context of a hypothetical automotive engine control application.

Using up to 5 external interrupts of 80C51 family microcontrollers

AN420

Most 80C51 family derivatives offer at least two dedicated external interrupt inputs. However, interrupts generated by on–chip peripherals can mimic additional external interrupt inputs. This application note describes how to utilize the MCU on–chip timer and UART to provide three additional external interrupts and includes an ASM program listing that shows how to configure the peripherals accordingly.

8051 family warm boot determinations

AN424

For original members of the 8051 family that don't provide a specific power fail indicator, there isn't an obvious way to differentiate between power–up reset (i.e. 'cold boot') and other resets such as generated by a switch or watchdog timer. This application note describes (and illustrates via ASM example) a technique to determine whether power cycling has occurred by checking the contents of on–chip RAM for a known value.

RAM loader program for 80C51 family applications

AN440

Author: Greg Goodhue

For debugging and test purposes or to accommodate upgrades and bug fixes, applications using external code memory can take advantage of the ability to download programs via serial port. This application note includes the ASM listing for a high speed serial download routine. It works with the most popular format known as 'Intel Hex' which is widely supported by hardware (ex: EPROM programmer) and software (ex: ASM , 'C' compiler) tool suppliers.

IEEE Micro Mouse using the 87C751 microcontroller — AN443

Author: Tracy Ching

The IEEE Micro Mouse contest challenges autonomous (self–powered, local intelligence only) robots to navigate a maze as quickly as possible. This application note describes such a robot based on the 8XC751 in complete detail including theory of operation, schematic and ASM listing. Along with the MCU, the design includes UCN–5804B stepper motor drivers for propulsion and OP–240A IR emitters working in conjunction with OPL–560–OC detectors for position sensing.

Automatic baud rate detection for the 80C51 — AN447

Author: Greg Goodhue

In many situations, it's desirable for the MCU to automatically determine the baud rate of a connected system, rather than relying on a fixed value in software or the need for DIP switch, jumpers or other external setting. An ASM routine is presented that uses a timer to measure the duration of a received character and compares it with a table of known values associated with standard baud rates.

Determining buad rates for 8051 UARTs and other UART issues — AN448

Most applications that utilize the UARTs built into many 8051–family MCUs utilize an on–chip timer for baud rate generation. This application note presents a table showing the timer values associated with standard baud rates (ex: 300 – 153.6k bps) for a variety of crystal frequencies (ex: 1.8432 – 36.864 MHz). Other practical concerns are addressed including the use of off–frequency crystals and UART initialization.

Measure resistance and capacitance without an A/D — AN449

Author: Tracy Ching

Exploiting the 'one–shot' concept provides a way to convert an analog value into a time delay which is easily measured with an MCU timer. Common examples rely on an RC network comprised of a potentiometer (i.e. variable resistance) and a fixed capacitor. This application note discusses a number of variations (ex: voltage comparator, HC/HCT logic, direct connect to MCU I/O pin) and characterizes the performance (ex: resolution & accuracy) of each.

In–circuit programming of the Philips 87C576 microcontroller — AN455

The 87C576 microcontroller features an in–circuit programming capability that streamlines production flow and accommodates last second code revisions. The programming scheme uses an 'Intel Hex'–like format (see AN440 "RAM loader program for 80C51 family applications"). This application note defines the commands for in–circuit programming (ex: program data, read data, set timing parameters, initialize & verify security bits, etc.) and includes schematic and PCB layout for a programmer .

Using LC oscillator circuits with Philips microcontrollers — AN456

Though not as precise as crystals, an LC (inductor & capacitor) oscillator is a cost–saving option when absolute clock accuracy isn't required. This application note characterizes the suitability of a selection of commercial inductors and capacitors, showing the clock rate derived from various combinations. Supply voltage variation as well as temperature sensitivity and compensation are evaluated using actual circuit examples.

80C51 External memory interfacing AN457

For those applications that require external EPROM or SRAM memory, ever–increasing clock rates require close attention be paid to interface timing. Starting with an overview of the memory interface, this application note then itemizes the key specs for various memory speed selections. Finally, a detailed timing analysis is performed for a 33 MHz 80C51 system including EPROM, SRAM and TTL using actual device specifications.

Dual data pointers for '51 family AN458

Ongoing improvements to the '51 architecture include the addition of a second data pointer (DPTR) register that increases speed and reduces code size for most applications. A review of data pointer fundamentals is followed by a description of the architectural details associated with the second DPTR. ASM examples and 'C' compiler benchmarks are used to illustrate the improvement possible for common operations such as block move and interrupt context switch.

Workbench EMC evaluation method EIE/AN91001

Electronic products are increasingly subject to global regulations concerning electro–magnetic compatibility (EMC) which dictate emission and immunity requirements. Proper PCB design is critical to avoid expensive physical shielding countermeasures. This application note describes various regulations and test procedures, discusses workbench EMC evaluation techniques and itemizes emission and immunity concerns specific to audio, video and digital designs.

A/D conversion with P83CL410 PCF1252–x AN91006

Author: Th. v. Daele, Product Concept & Application Laboratory, Eindhoven, The Netherlands

Presents a variation of the concepts described in AN449 "Measure resistance and capacitance without an A/D". In this example, a low–power MCU (P83CL410) utilizes a power–on reset IC (PCF1252–x) as a voltage comparator. The output of the comparator connects to an MCU interrupt input. The time to charge & discharge a known RC to Vcc is then measured using an MCU timer and serves as an index into a voltage lookup table. Includes schematic and ASM program listing.

Driver for 8xC851 E2PROM AN91009

The 8xC851 is a version of the 80C51 that includes 256 bytes of E^2PROM (electrically erasable PROM). This application note describes a library of routines supporting access to the E^2PROM including initialization, status check, read/write byte & block, block erase and security. Example programs utilizing the routine are provided in ASM, PL/M51 and 'C'. A set of E^2PROM–related macros (ex: read, write, copy, erase) that work with the XRAY51 debugger are described.

Low RF–emission applications with a P83CE654 microcontroller AN92001

The techniques described in AN91001 "Workbench EMC evaluation method" are used to measure various 'single–chip' (i.e. no high speed external address & data bus) configurations. Notably, the advantages of QFP package relative to DIP (ex: smaller area, shorter leads, multiple grounds) are quantified as are improvements associated with multi–layer boards. Includes specific recommendations for QFP–based board layout and bypassing.

Using the analog-to-digital converter of the 8XC552 microcontroller

EIE/AN93017

The 8XC552 microcontroller is notable for inclusion of an 8 channel 10–bit analog–to–digital converter (ADC). However, achieving the full accuracy offered by the unit requires some design considerations. This application starts by defining the architecture and implementation of the ADC in detail. Subsequently, a number of guidelines are presented covering such topics as slew rate, source impedance, grounding, decoupling and PCB layout. Finally, both interrupt and polled versions of 'C' programs that exercise the ADC are provided.

Electro magnetic compatibility and printed circuit board (PCB) constraints

ESG89001

A comprehensive and quantitative examination of electro–magnetic compatibility (EMC) issues including PCB layout, material and format (i.e. layers), power distribution, decoupling, cable & connector shielding and placement and filtering components. Measurements are taken from a test board (schematic and layout included) which is configurable to utilize various combinations of the EMC countermeasures.

Chips push CAN bus into embedded world

This week:
Special Report
Surface-mount technology
gets bigger as packages
get smaller

Electronic Engineering
TIMES

Monday
August 24, 1992
Issue 707

A CMP Publication

THE INDUSTRY NEWSPAPER FOR ENGINEERS AND TECHNICAL MANAGEMENT

Chips push CAN bus into embedded world

Sunnyvale, Calif. — The Controller Area Network (CAN) serial bus is being thrust into a key role in embedded systems, boosted by a series of recent silicon announcements. Delta-t GmbH, Intel Corp., Motorola Semiconductor and Philips/Signetics Co. have all prepared new chips that should slash the cost of a CAN bus connection, increase bus functionality and accelerate the bus's migration from a little-known automotive standard to a widely used industrial interconnect scheme.

Developed by Intel with Robert Bosch GmbH in the mid-1980s to solve cabling and control problems in vehicles, the CAN bus combines an inexpensive, one-wire or two-wire medium with multimaster, error-correcting protocol and very high resistance to electromagnetic interference. Unlike most multi-master buses, however, CAN also guarantees a maximum latency, often in the neighborhood of 1 ms, for high-priority messages while making room for lower-priority traffic as well.

Thus, although the bus was originally conceived as a way to reduce the literally miles of wire in a modern automobile, it is in many ways ideally suited for a wide range of industrial control applications as well.

General-purpose bus

CAN has since emerged as a general-purpose sensor/actuator bus system for distributed real-time control applications that extend beyond the automotive realm. "CAN was spotted by the industry as a very promising field-bus technology in the area of industrial automation," said Tom Suters, systems architect at Philips Medical Systems, in Da Best, the Netherlands.

"We are beginning to see CAN as a widely used standard in trucks and farm equipment, industrial automation and even some medical equipment," agreed Signetics marketing manager Mike Thomson.

But CAN's opponents have often criticized its high implementation cost as well as the lack of controller chips and supporting tools. Proponents have argued that volume production for the automotive market would inevitably bring down the silicon prices, but that has yet to happen. Now vendors are pointing to a new generation of low-cost or highly integrated controllers as the solution.

Motorola is sampling a controller, dubbed the 68HC705X4, that will be supported by an evaluation board and an emulator. Other implementations based on the 68HC11 and the 68HC16/683XX are in the pipeline.

Intel Corp., working in conjunction with design partner Bosch, has sampled a next-generation CAN chip, the 82527, that offers support for two sizes of message indentifiers—11 bit and 29 bit—as specified in CAN Spec. 2.0, released in September.

Craig Szydlowski, product marketing engineer for CAN at Intel, said the original CAN 1.2 standard allowed for only 11-bit message indentifiers. Supporters of J1850—the competing automotive bus favored by the Society of Automotive Engineers—fought for an optional larger message field so 1850-style 27-bit commands could be easily mapped into CAN protocols.

"This gives CAN the ability to broadcast commands using what had been essentially an address field," explained Signetics' Thomson. "For example, a CPU might send out a command telling all of the lamps in a vehicle to test themselves for conductivity and report any burn-outs."

The new format does not imply that CAN and J1850 physical buses can be easily interfaced, Szydlowski emphasized, only that the message structure would be similar.

Just the next step

The Intel chip "looks like a smart RAM," Szydlowski said, with RAM space on-chip to store 14 8-byte receive/transmit message objects (with a fifteenth area for received message objects). The message objects are stored at fixed RAM locations. One on-chip filter is dedicated to the receive message object, with a global acceptance filter mask used for the other 14 message objects.

Intel's new chip is merely the next step in integration, to be followed by efforts to embed 527 functionality into a standard Intel 16-bit controller architecture.

Philips/Signetics has added to its product line as well but but from a different direction. The company has integrated a CAN bus controller into a heavily configured 8051-type MCU to come out with the 8XC592. The chip combines large ROM and RAM space with five 8-bit I/O ports, a 10-bit A/D, two 8-bit-resolution PWM outputs, and the usual counter/timers and UART.

The announcement puts Philips and Intel in a confrontation over processing power. Intel will pursue a strategy of integrating CAN controllers into 16-bit MCUs and driving up node performance. The Dutch giant is going in the opposite direction, starting with a high-end commodity 8-bit part and heading downward in cost.

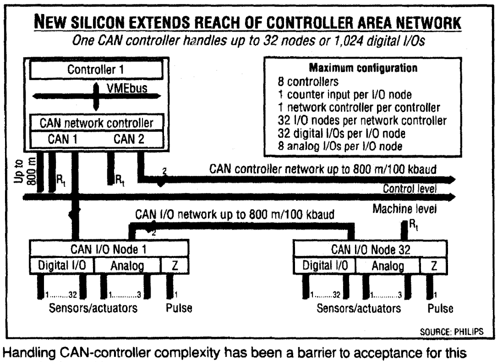
"Customers will want a part for a real simple node, without even a CPU on it," claimed Thomson. "You don't hang an 8051 on a light bulb. We are working on a controller solution for under $1." Philips has also announced a high-speed CAN transceiver that can be hooked directly to its controllers.

The IX controller from Delta-t GmbH, meanwhile, should prove useful to manufacturers of sub-systems for multiprotocol environments when it debuts sometime next year. The protocol used for each application will be able to be programmed into a flash-memory section of the controller.

The design activity promises to both lower the cost of a simple CAN-bus node and increase the computing power that can be integrated into an expensive node. Both moves should expand the market deeper into industrial control and instrumentation applications, although CAN partisans see little hope of displacing J1850 from the big three domestic makers.

CAN is a multimaster bus topology network that connects several stations. An International Standards Organization draft from 1990 specifies the first two layers of the Open Systems Interconnect (OSI) model: the physical layer and the data-link layer. Philips Medical Systems developed an application layer, using CAN to control X-ray diagnostic systems, control real-time image acquisition and connect user-interface devices. Today, the three layers together form the CAN Reference Model.

–Addtional reporting
by Ron Wilson

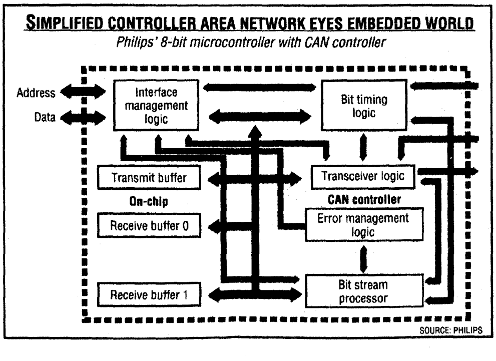

NEW SILICON EXTENDS REACH OF CONTROLLER AREA NETWORK

One CAN controller handles up to 32 nodes or 1,024 digital I/Os

Maximum configuration
8 controllers
1 counter input per I/O node
1 network controller per controller
32 I/O nodes per network controller
32 digital I/Os per I/O node
8 analog I/Os per I/O node

SOURCE: PHILIPS

Handling CAN-controller complexity has been a barrier to acceptance for this control-oriented bus.


SIMPLIFIED CONTROLLER AREA NETWORK EYES EMBEDDED WORLD

Philips' 8-bit microcontroller with CAN controller

SOURCE: PHILIPS

Chips push CAN bus into embedded world

NEWS

Controller Area Network (CAN) Products From Philips Semiconductors

8XC592 CAN Microcontroller - The 8XC592 is a stand-alone high-performance microcontroller based on the 80C51 architecture. Its on-chip CAN interface makes it ideal for applications with a harsh, noisy environment.

The **8XC592 Features** include:

- 16K x 8 ROM (83C592)
- ROMless (80C592)
- 512 x 8 RAM, expandable externally to 64k bytes
- Three Standard 16-bit timer/counters
- A 10-bit ADC with 8 Multiplexed Analog Inputs
- Two 8-bit Resolution, Pulse Width Modulation Outputs
- Five 8-bit I/O Ports Plus One 8-bit Input Port Shared with Analog Inputs
- CAN Controller with DMA Transfer between Internal Data RAM and CAN Registers
- Standard 80C51 UART
- On-Chip Watchdog Timer

82C200 CAN Interface - The 82C200 is a highly integrated stand-alone controller for CAN. The 82C200 with a simple bus line connection performs all the functions of the physical and data-link layers. The application layer of an electronic control unit (ECU) is provided by a microcontroller, to which the 82C200 provides versatile interface.

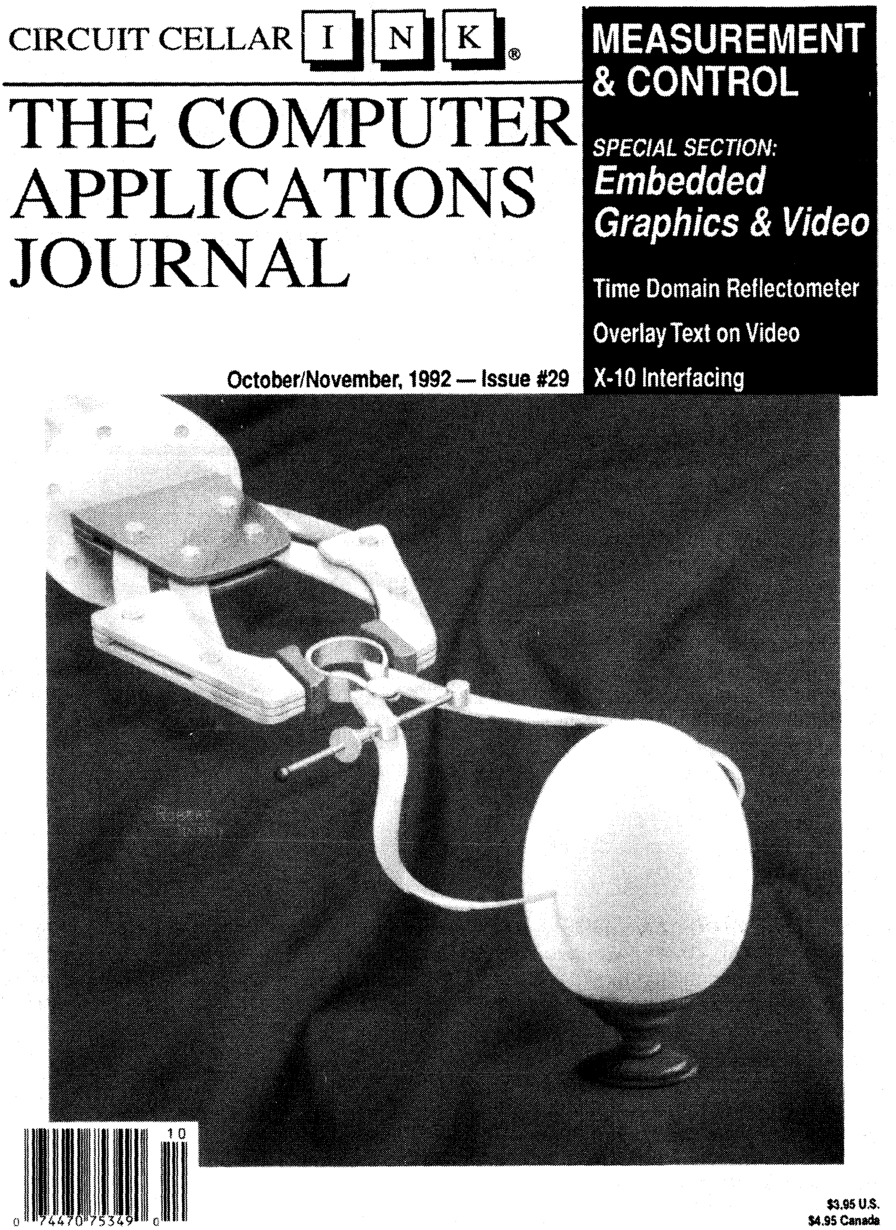
The **82C200 Features** include:

- Multi-Master Architecture
- Interfaces with a Large Variety of Microcontrollers
- 2032 Message Identifiers
- Powerful Error Handling Capability
- Configurable Bus Interface

82C150 Serial Linked I/O CAN Interface - The 82C150 Serial Linked I/O CAN is a single-chip 16-bit I/O device with an on-chip CAN-controller. 82C150 is a very cost-effective way of increasing the I/O-capability of a microcontroller as well as reducing the amount and complexity of wiring. Advanced safety provided by the CAN protocol combined with low-cost makes the 82C150 a very attractive product for a wide variety of applications.

The **82C150 Features** include:

- Single-Chip I/O Device with CAN Protocol Controller
- 16 Configurable Digital or Analog I/O Ports
- Each Port Individually Configurable via the CAN bus
- 10-bit A/D converter with up to 6 Multiplexed Inputs
- Bit Rate 20 kbit/sec to 125 kbit/sec



CIRCUIT CELLAR INK®

THE COMPUTER APPLICATIONS JOURNAL

MEASUREMENT & CONTROL

SPECIAL SECTION:
Embedded Graphics & Video

Time Domain Reflectometer

Overlay Text on Video

X-10 Interfacing

October/November, 1992 — Issue #29

$3.95 U.S.
$4.95 Canada

Add Text Overlay to Any Video Display

> All the latest VCRs have on-screen programming in an effort to simplify the notorious task. Now you can have your HCS II or any other computer send you messages while you're watching your favorite TV show.

SPECIAL SECTION

Bill Houghton

Suppose, for the moment, you've built and installed the HCS II home control system described primarily in issues 25 and 26 (February/March and April/May '92) of the *Computer Applications Journal*. In issue 27 (June/July '92), Ed Nisley described an add-on LCD output device as a way to obtain information about the status of the system and its various nodes. It's a nice addition to the network, but useful only when you're near to the display module. What do you do if you're across the room watching TV settled into your favorite armchair? You could get up and venture across the room. Or, if you build the interface described in this article, you could hit a button on your HCS II IR remote and see network information displayed on your TV set overlaid onto the program you are watching.

This article describes an On-Screen Display (OSD) terminal for HCS II (we'll call it "TV-Link" to match the other HCS II modules) that allows color text characters to be displayed on top of a background color video signal. The terminal is built around the Philips/Signetics 87C054 OSD microcontroller.

FEATURES OF 87C054

The 87C054 is an 80C51-based microcontroller designed to provide an advanced OSD for TV and video applications. It can produce characters in eight foreground and eight background colors. In addition, the background color can be removed, showing through the original video. It also has nine pulse-width modulator outputs for controlling analog functions. Similar to a standard 80C51, it has 28 digital I/O pins, two external interrupts, and two timer/counters. RAM and ROM spaces on the 87C054 are larger than the 80C51: 192 bytes of RAM and 16K bytes of EPROM. (The OSD has *additional* RAM and EPROM areas that are not part of the normal 80C51 memory map.)

One unique feature of the 87C054 is what Signetics describes as a "soft-

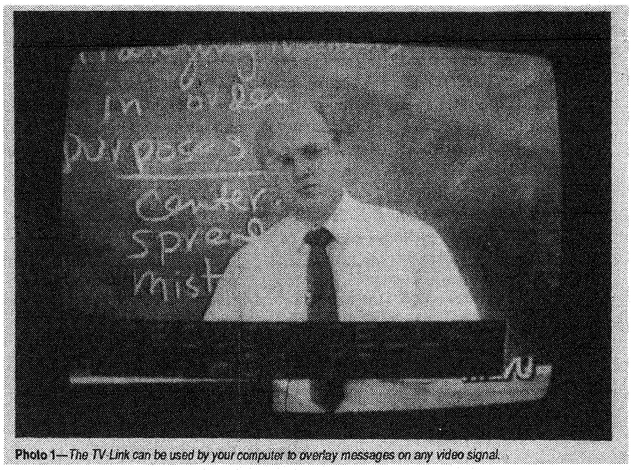


Photo 1—*The TV-Link can be used by your computer to overlay messages on any video signal.*

Add Text Overlay to Any Video Display

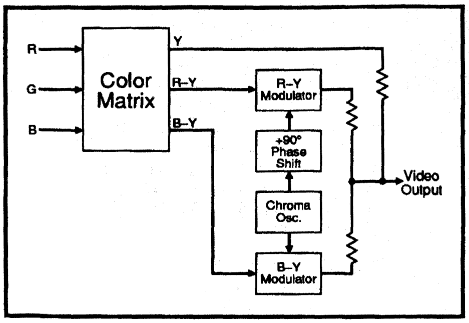


Figure 1—*Converting from RGB to NTSC involves modulating and adding difference signals.*

ware ADC." This ADC consists of an internal 4-bit DAC that feeds one input of a comparator. The other comparator input can be connected to one of four I/O pins. The output of the comparator is tied to a status bit in a register that is testable by software. A TV set often uses this logic for measuring the AGC voltage during tuning. A real-time clock and other low-precision analog measurements can also use it as a zero-crossing detector.

The OSD of the 87C054 consists of a 128-character RAM array (OSDRAM), a 64-character font EPROM, a video clock oscillator, and the OSD logic. The OSD logic accepts horizontal sync (*HSYNC*) and vertical sync (*VSYNC*) signals and provides three digital video outputs for character information. In the datasheet for this part, these outputs are called *VID0*, *VID1*, and *VID2*, but they can also (and perhaps better) be thought of as *RED*, *GREEN*, and *BLUE*. A multiplexer control output is also present to indicate when to display character data or original video information.

The video clock oscillator provides timing for the character dots. In most applications, this oscillator is simply an LC tank circuit connected to the *VCLK* pins. The frequency controls the character width. One nice feature is that the video clock is killed at the leading edge of *HSYNC* and restarted at the trailing edge of *HSYNC*, which causes the video clock to start in the same phase on every line, ensuring the dots align vertically from one scan line to the next.

The character font stores the binary pattern for the individual characters. Characters are 14 dots wide and 18 scan lines high.

The OSDRAM stores the characters to be displayed on the screen along with certain attribute data pertaining to those characters. Once a character has been written to the OSD, no further CPU intervention is required to "refresh" the screen.

Many OSD architectures have been developed over the years for use in the consumer television market. Almost all of them have required fixed character row formats, limiting the designer's flexibility in designing video menus and screens.

The 87C054 was designed to avoid such constraints, and there are no architectural limits on the number of characters in a row of text or the number of rows of text to a screen. (There are physical limits imposed by the dot clock frequency and the scan rate, of course.)

The *HSTART* and *VSTART* parameters in the *OSORG* (on-screen origin) register define the intial position of the start of the OSD. Once the initial vertical and horizontal positions have been found, the 87C054 will "fetch" characters from the OSDRAM and place them sequentially on the screen. In order to have multiple rows of text, a special character has been defined and is referred to as *NEWLINE*. The *NEWLINE* character is much like a carriage return/line feed sequence on a computer in that it terminates the current row of characters and starts a new row of text. One advantage of this architecture is that it eliminates the need to pad display memory with space characters. The fetching and painting of rows of text will continue until either a new vertical sync pulse is detected or until an *END* attribute is fetched along with a *NEWLINE* character.

NOW FOR THE DETAILS...

Now that you understand the concepts of an OSD operation and the capabilities of the 87C054, focus your attention on the details required to overlay characters onto live video.

The 87C054 OSD has a multiplexer output for switching video sources. Simply switching between the input video signal and the OSD character data would be nice. Unfortunately, you can't because the input video (from our home entertainment center) is in NTSC format and the character data is in RGB format. (Keep in mind that the goal is to input live video, add on-screen text, and present the result as a video signal at the output of our circuit.)

One solution is to decode the input video into separate red, green, blue, *HSYNC*, and *VSYNC* signals. Then you could perform the multiplexing between video information and character data in RGB format. The resultant signals could then be encoded back into baseband video. If there were other reasons for the conversion into RGB, such a conversion

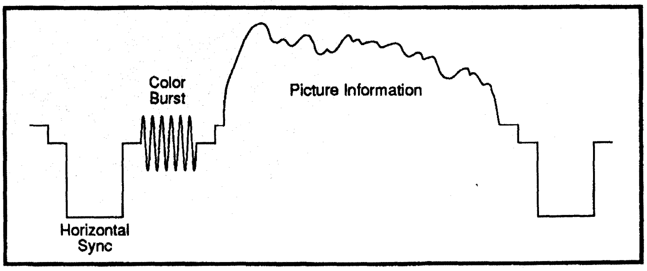


Figure 2—*A typical line of NTSC video consists of an initial horizontal sync pulse, followed by a short color burst signal, then the actual picture information.*

Add Text Overlay to Any Video Display

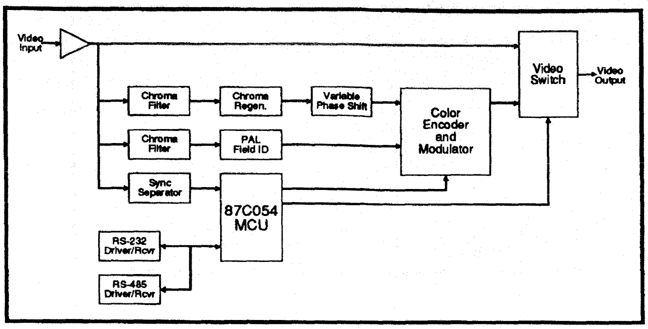


Figure 3—*At the core of the TV-Link is the Signetics 87C054 microcontroller. The unit supports both RS-232 and RS-485 communications with a host computer.*

would be the way to go. However, the process of converting to RGB and then converting back to video introduces some distortion that could be visible on the screen.

Another solution is to find a way to encode the RGB data from the OSD microcontroller into video. Then you can simply switch between the two sources. Sound simple? The situation gets a little more complex when you consider the issues of making the characters appear with the proper color in NTSC. Reviewing how color is encoded in NTSC is in order.

COLOR TELEVISION

When you first look at video, you often wonder why in the world it was done the way it was. A long time ago, before Americans had ever heard the names of Japanese TV makers, RCA research labs were developing color televeision. One of the requirements imposed on designers by the FCC was that the broadcast signal needed to be compatible with existing black-and-white TV sets and had to be contained within the same bandwidth as a B/W signal. Such requirements meant that some component of the signal had to contain overall brightness information, which is the main reason why they could not simply transmit separate R, G, and B channels within the video bandwidth allowed. To make a very long story short, the engineers involved decided that the scene brightness (which they called "Y" or luminance) could be described by the relationship

$$Y = 0.59G + 0.3R + 0.11B$$

Someone observed that if they took two copies of the luminance signal and subtracted one copy from one of the colors (say, RED) and did the same with a different color (say, BLUE), the result would be two signals that contained all of the information needed to represent color. These resultant signals, R–Y and B–Y, are the color difference signals.

Now that you have two signals, how can they be transmitted on one RF carrier? The answer they came up with was to modulate one of the signals (B–Y) with an RF carrier. The

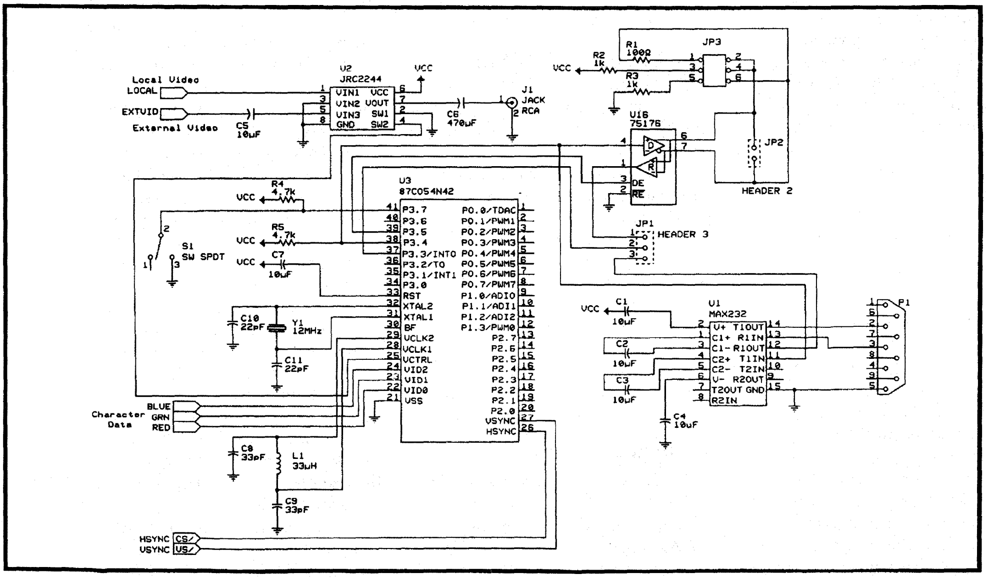


Figure 4a—*The JRC2244 switches between the incoming video signal and the on-screen characters under control of the 87C054 MCU.*

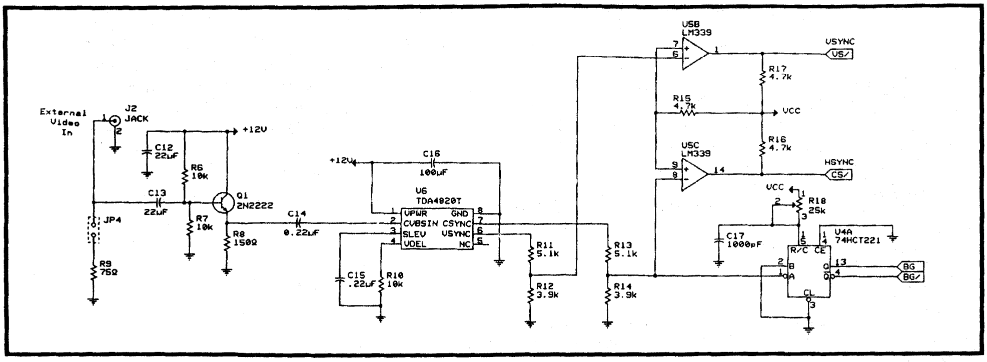


Figure 4b—*The TDA4820T sync separator provides the processor with horizontal and vertical sync signals.*

other signal (R–Y) was to be modulated with the same RF carrier, but the carrier would be shifted by 90°. When the outputs of the two modulators are added together, the result is the *vector sum* of the two signals, containing both an amplitude and a direction (phase angle). See Figure 1.

We now have a single signal that contains all of the color information. The TV receiver needs just one more piece of information to demodulate this signal. It needs a reference for the carrier used for the modulation, that is, the receiver needs to know where 0° of the color carrier is (in video, this color carrier is referred to as the *chroma subcarrier*). In order to give this reference to the receiver, a small number, or "burst," of cycles of the color subcarrier (hence, the term color burst) are transmitted on the back porch of the horizontal sync pulse. In most NTSC systems, this chroma subcarrier has a frequency of approximately 3.58 MHz. A typical line of color NTSC video is shown in Figure 2.

In order to convert the character data from the OSD into NTSC, you will need to sum the data into R–Y and B–Y components. Then you will need to modulate these components with a chroma subcarrier at 0° for B–Y and +90° for R–Y.

One more item to consider. If you have an output video signal composed of a video source with characters overlaid onto it, the chroma subcarrier reference (i.e., color burst) present on the output video signal is the color burst provided in the original input video. In order for the receiver/monitor to interpret the color of the OSD characters correctly, the chroma subcarrier used to modulated the OSD's R–Y and B–Y components must have exactly the same frequency and phase as the color burst on the original video input signal.

Once you get the OSD information in the form just described, you can switch between this "OSD video" and the original input video to produce the final output.

THE TV-LINK HARDWARE SOLUTION

Figure 3 shows a block diagram of the TV-Link, while Figure 4 shows the schematic.

Referring to the schematic, the original input video connects to J2 and is AC coupled into buffer amplifier, Q1. This amplifier provides load isolation between the video signal source and the circuits on the TV-Link board. JP4 is a jumper allowing for a 75-ohm termination resistor to be connected to J2. The output of the Q1 buffer amplifier feeds the sync separator, the video switch, and the chroma subcarrier regenerator circuits.

SYNC SEPARATION

The sync separator consists of U6, a TDA4820T Philips sync separator. The video signal is coupled into the TDA4820T through capacitor C14, where it is amplified with a gain of 15. The black level clamping voltage is stored in capacitor C14. From the stored black level voltage and the peak sync voltage, the 50% value of the peak sync voltage is generated and stored in capacitor C15. A slicing level control circuit ensures a constant 50% peak sync value regardless of the picture content amplitude provided the sync pulse amplitude is between 50 mV and 500 mV. A comparator in the composite sync slicing stage compares the amplified video signal with the DC voltage derived from the 50% peak sync voltage, producing the composite sync output. Vertical slicing circuits compare the composite sync signal with a DC level equal to 40% of the peak sync signal, producing the vertical sync output. The reduced vertical slicing level ensures more energy for the vertical pulse integration. The slope is double integrated to eliminate the effects of interference caused by noise or line reflections. The value of resistor R10 sets the vertical integration delay time.

The outputs of the sync separator are positive-going signals with peak amplitudes exceeding 10 V. Resistor pairs R11/R12 and R13/R14 serve as voltage dividers for the *VSYNC* and *CSYNC* outputs, respectively. An LM339 comparator, U5, serves as an inverter for the sync signals because the modulator circuits require active low sync signals.

There is a great tendency with video circuits to make the coupling capacitors very large to pass the low-

Add Text Overlay to Any Video Display

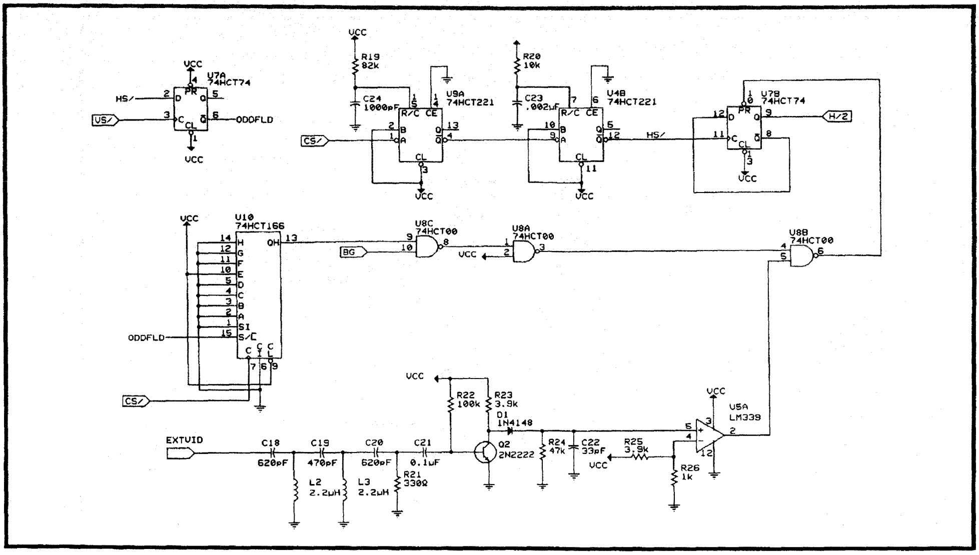


Figure 4c—*In order to properly overlay colors onto a PAL signal, you must know whether you're on an odd or an even field, so extra circuitry must be included on the board to support PAL.*

frequency sync components (60/50 Hz, typically) into low-impedance nodes. The TDA4820T has a moderately high input impedance on pin 2. Because the black level is stored in C14, the value of C14 should be kept close to 0.22 μF.

THE 87C054 MCU WITH OSD

The 87C054 microcontroller, U3, accepts composite sync and vertical sync signals from the sync separator and provides RGB digital outputs for character data. The multiplexer control output, *VCTRL*, connects to the video switch, U2, a JRC2244.

Inductor L1 and capacitors C8 and C9 form a video clock oscillator that determines the width of a character font dot. The values of these components are not critical but are typically chosen such that a video dot width is equal to the spacing between scan lines. This oscillator is killed at the leading edge of the *HSYNC* signal and allowed to startup at the trailing edge. Such synchronization causes the oscillator to start at exactly the same point from one scan line to the next, causing character dots to appear in exactly the same spot on each line.

In addition to the OSD functions, the 87C054 also performs network interfacing and protocol tasks. This microcontroller has plenty of performance bandwidth because the OSD logic is self-refreshing and independent of the MCU core.

VIDEO SIGNAL SWITCHING

The JRC2244 video switch, U2, contains three video inputs, two of which are used in this application. One of these inputs, *VIN1*, is capacitively coupled to the OSD video signal. This signal is the 87C054's RGB data after encoding into baseband video. The other input, *VIN3*, is capacitively coupled to the original video input signal. The JRC2244 provides internal bias sources to provide DC restoration to its video inputs. The JRC2244 accepts a switching control signal from the 87C054 and switches its output between the original video input and the OSD video signal. The video switch also has an internal 75-ohm line driver in its output stage.

The JRC2244 has a moderate input impedance of about 15k ohms, allowing 10-μF coupling capacitors to be used. The output coupling capacitor is large because this signal can be used to drive 75-ohm loads.

RGB ENCODING

The LM1886, U13, and the LM1889, U14, encode the RGB data from the 87C054 into baseband video. The LM1886 has three DACs, one for each color. Each of these DACs has 3-bit inputs, but because the 87C054 data is digital, the inputs to the LM1886 DACs are tied together yielding an output for each DAC that is either full-scale or zero. The outputs of the three DACs are internally summed to produce the luminance, R–Y, and B–Y amplitudes. The LM1889 accepts the regenerated chroma subcarrier, modulates the R–Y and B–Y signals, and produces baseband video on pin 13. Transistor Q5 is used as a buffer amplifier with voltage dividers R49 and R50 producing proper levels for the video switch. Note that the LM1889 accepts an external subcarrier signal at the junction of R46 and C52, but this subcarrier undergoes a phase shift caused by the resistor and capacitor networks associated with pins 1

Note: 8XC054 replaced by 9XC055.

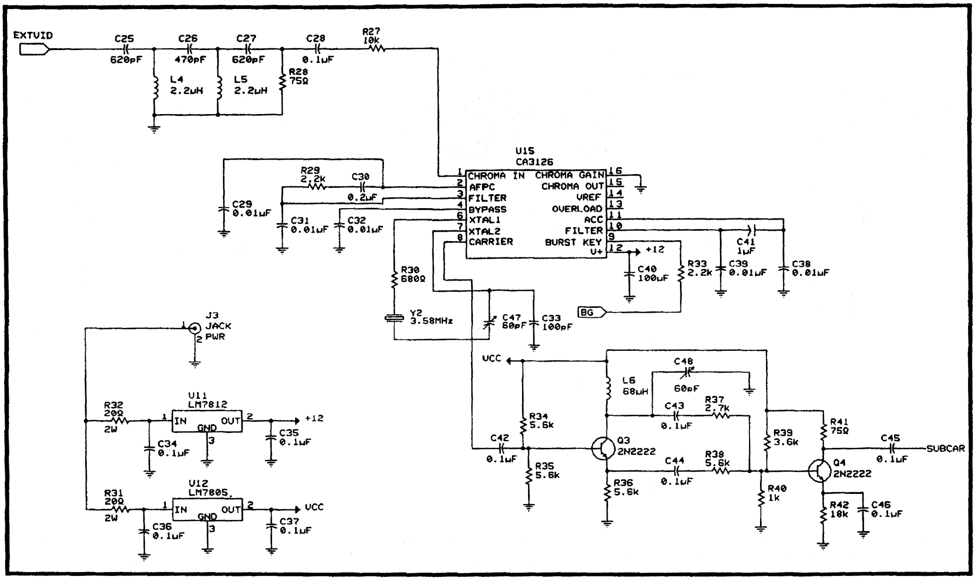


Figure 4d—*The CA3126 TV chroma processor is designed specifically for regenerating chroma subcarriers.*

and 18 of the LM1889. This phase shift will need to be considered when the subcarrier is regenerated.

CHROMA SUBCARRIER REGENERATION

The circuits that reproduce a chroma subcarrier in the same frequency and phase as the color burst consist of a high-pass filter, a sample-and-hold phase-locked loop (PLL), and a phase shift network and amplifier.

The passive high-pass filter consists of inductors L4 and L5, resistor R28, and capacitors C25, C26, and C27. The filter starts passing signals at about 3.2 MHz, allowing the chroma subcarrier to pass through to the CA3126, U15.

The CA3126 is a TV Chroma Processor IC designed specifically for regenerating chroma subcarriers. This IC contains a VCO and a PLL with sample-and-hold circuits in the error correction loop. As a result, the VCO-generated carrier is compared with the chroma signal from the high-pass filter during the time that color burst is present, indicated by the burst gate pulse (which I will describe later).

The regenerated carrier output is present on pin 8 of the CA3126. Even though this carrier is phase locked to the color burst, it is not at exactly the same phase as the color burst. The nature of a PLL is such that the output will be locked but will always have some constant fixed phase delay relative to the input. Also, recall that the input circuits of the LM1889 added an additional constant phase shift to the injected carrier.

The phase shift network and amplifier consisting of the Q3 and Q4 stages compensate for these fixed phase delays. This circuit provides an output whose phase is adjustable by means of variable capacitor C48, and has a tuning range of approximately 0° to 160° of phase shift. For a given input signal amplitude, the output signal amplitude is constant, regardless of the phase shift introduced. The output of this circuit is the signal injected into the LM1889 circuits.

CONNECTING TO THE HCS II

Now that you have a working terminal circuit for overlaying text onto live video, you still need to connect it to the HCS II network. In order to do this, you will need a serial interface compatible with the network, some software that handles network message formats, and software that interprets network messages and creates responses or actions (or both) to those HCS II network messages.

This particular design includes both an RS-232 and an RS-485 interface. U1, a MAX232, provides the RS-232–to–TTL conversion for both the transmitter and receiver. U16, a 75176, provides the RS-485–to–TTL conversion. JP1 connects the receiver pin of the 87C054 to either the RS-232 or RS-485 interfaces. The transmitter pin of the 87C054 connects to both the RS-232 and RS-485 interfaces. One pin of the 87C054, P3.5, controls the driver enable of the 75176, allowing for selective talking on the HCS II network. JP3 provides for termination of the network.

"But, wait a minute! The 87C054 doesn't have a UART," you say. True. There is no built-in UART on the 87C054 and the part does not have a transmitter pin or receiver pin.

Add Text Overlay to Any Video Display

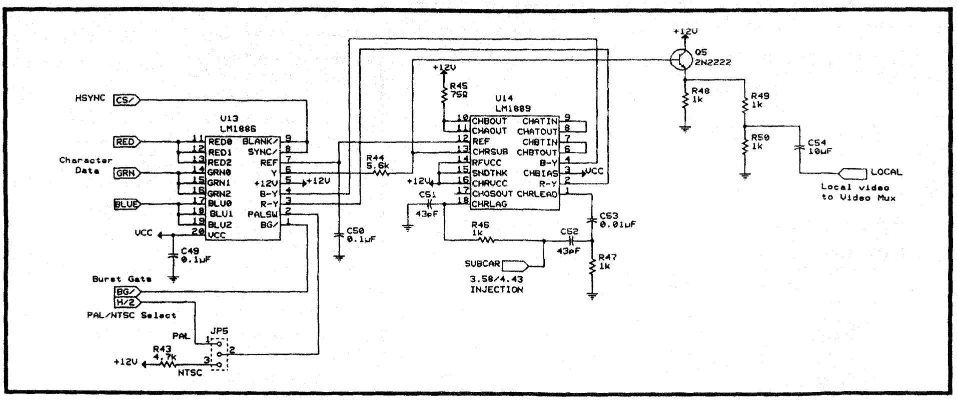


Figure 4e—*Encoding the RGB data into baseband video is accomplished with an LM1886, which contains three DACs, and an LM1889, which accepts the regenerated chroma subcarrier, modulates the R–Y and B–Y signals, and produces baseband video.*

In this application, the serial data transmission and reception has been performed in software. The routine that handles serial transmission and reception was taken from the Signetics BBS ([800] 451-6644). It was originally designed for the 87C751 and had to be slightly modified to operate with one of the 87C054's timer/counters. The technique is often called "bit banging" and has the advantage of saving some hardware if you can afford the necessary time required of the software.

NETWORK PROTOCOL PROCESSING

As I indicated earlier, in addition to the serial interface software, you need code that handles network message formats. The code starts by waiting until either a "#" or an "!" is received, either of which indicates the start of a network message, then the entire message is stored in a buffer.

Once the carriage return has been stored, the beginning character of the message is checked to see whether the message includes a checksum. If the message does not contain a checksum, the packet is assumed to be valid and the contents of the packet are processed. If a checksum is included, then the VERIFY routine is called to perform a checksum calculation on the packet. If the checksum matches, the packet is processed; otherwise, it is ignored and I return to waiting for the next network message.

My original plans for handling network checksums included a checksum generator for sending network responses and a checksum checker for received messages. However, when I flowcharted the needs of both routines, I found that an awful lot of the logic was common to both. I went back and looked at the suggestions that Ed Nisley had provided for handling the checksums and understand now why he made those suggestions. My VERIFY routine's logic is based on Ed's previous work.

The VERIFY routine performs two functions. First, it takes the checksum digits in the packet, converts them to binary numbers, and stores them in temporary variables. Next, the checksum digits are replaced with ASCII zeros and the checksum of the string is calculated. If the checksum matches, the error flag, CHKERR, is cleared; otherwise, it is set. The checksum that was calculated is converted to ASCII and stuffed into the checksum digits position, replacing the ASCII zeros.

To prepare a string for transmission, all that is necessary is to stuff the message in the buffer with the checksum digits set to ASCII zeros and call the VERIFY routine. To check a message for correct checksum, simply call the VERIFY routine and check the CHKERR flag on return.

Once the checksum verification (if required) has been performed, you still need to process the packet to see if it belongs to this terminal, and if it does, then you need to determine what action the network controller is asking you to take.

The PROCESS routine first scans the packet, converting characters into upper case until the end of the packet has been reached. Next, the first character is examined to determine if the packet has checksums or not and a pointer is set to the NODEID position of the packet. The NODEID in the packet is compared with the NODEID variable. If there is no match, the packet is ignored and you wait for the next network message. If it does belong do this terminal, you can process the body of the network message.

NETWORK COMMANDS AND SYNTAX

The real essence of a network message is to carry a command from the network controller to the terminal or carry a response from the terminal back to the network controller. Table 1 shows the syntax of the commands available for operating the TV-Link terminal. These commands allow the HCS II Supervisory Controller to manipulate ports on the 87C054, format text for display, implement special built-in display functions such as color bars, and to read and write

A = string	Set HCS II network address to string
Fx	Execute special function 0 Initialize screen 1 Display on 2 Display off 3 Display color bars 4 Wipe on 5 Wipe off
Hxy	Set Px.y high
Lxy	Clear Px.y low
Nn	Network response mode N0 = normal network interface, no auto error or acknowledge responses N1 = auto error and acknowledge response
Px	Query port x 0 = Port 0 1 = Port 1 2 = Port 2 3 = Port 3
Px= nn	Write to port x where nn= two-digit hex value 0 = Port 0 1 = Port 1 2 = Port 2 3 = Port 3
Rx	Query register x; returns two-digit hex number 0 = OSDT (contents undefined) 1 = OSAT 2 = OSCON 3 = OSORG 4 = OSMOD 5 = Default char. attribute 6 = Default background space attr 7 = Default NEWLINE attribute
Rx = nn	Write to register x; for use from outside of a string of text; writes to these registers from within a string; should use the \Wxnn command 0 = OSDT 1 = OSAT 2 = OSCON 3 = OSORG 4 = OSMOD 5 = Default char. attribute 6 = Default background space attr 7 = Default NEWLINE attribute
S= string	String for OSD display; can include escape sequences for text formating, color, selection, etc.
\Wxnn	Write to register x; for use within a string; functionally equiv. to the Rx = nn command
Special characterss for use within a string of text	
\E	End of Display at current position
\B	Background Space
\S	Split Background Space
\N	NEWLINE

Table 1—*The set of supported commands resembles that of most of the other HCS II network modules.*

OSD registers directly, giving full control of the OSD to the HCS II.

CONCLUSIONS

Developing this application was interesting and enjoyable. It also presented some challenges.

The 87C054 proved well suited to this application in large part because of the 80C51 core and that the OSD is independent of the CPU. Once characters have been written to the OSD, you can forget the OSD until you want to change the display, and the CPU is free to pursue other tasks.

The on-screen display and the microcontroller operations are primarily digital functions. The question of how to combine this technology with an analog video signal can be perplexing to most system designers whose professional experiences have been mostly digital circuits. One of the most perplexing issues during this project was how to re-create the chroma subcarrier. I knew that every color TV set had to perform this function, but finding out solutions took some searching before I discovered the CA3126. I'm hopeful you can profit from my experiences on this project.

My thanks to Herb Kniess and George Ellis of Signetics for their help. Thanks also to Greg Goodhue from Signetics, who wrote the software-based UART code for the 87C751 that I modified for this project.

Bill Houghton is an Applications Engineer at Signetics specializing in 80C51-based microcontrollers.

SOFTWARE

Software for this article is available from the Circuit Cellar BBS and on Software On Disk for this issue. Please see the end of "ConnecTime" in this issue for downloading and ordering information.

SOURCES

Requests for literature on Signetics/Philips microcontrollers including the "80C51-Based 8-Bit Microcontroller Data Handbook" may be directed to Sharon Baker at (408) 991-3518.

Contact Bill Houghton at (408) 991-3560 with technical questions specific to the 87C054 and for information on the availability of a PC board and components for this project.

Section 9

Control Area Network (CAN) Bus

CONTENTS

Control Area Network (CAN) overview 285

P82C150 CAN Serial Linked I/O device (SLIO) with digital and analog port functions 286

PCA82C250 CAN controller interface 288

SJA1000 Stand–alone CAN controller 290

P8xC592 8–bit microcontroller with on–chip CAN 292

P8xCE598 8–bit microcontroller with on–chip CAN 294

Control Area Network (CAN) overview

CAN OVERVIEW

The Control Area Network (CAN) is a multiplexed wiring protocol developed by Bosch for use in automotive applications. As products supporting CAN have been made available by Philips Semiconductors and other semiconductor manufacturers, the protocol has become used in other industries including: industrial automation, machine tools, medical equipment, and building environmental control, to name a few. The CAN protocol is attractive for use in a wide range of applications because it has powerful error detection capabilities and features differential drive, and can be used comfortably in critical high noise environments. CAN is also very flexible in terms of the transmission media and the connection scheme, and is generally easy to adapt to most applications.

Philips offers a wide range of parts that support the CAN protocol, including stand-alone parts as well as microcontrollers with integrated CAN interfaces. Datasheets for the 82C200 (Stand-alone CAN controller), 82C150 (CAN serial Linked I/O device), and 82C250 (CAN transceiver) are included in this section of this databook. Datasheets for the microcontrollers that have an integrated CAN interface (8XC592 and 8XC598) are included with Philips' 80C51 family products in Section 3 of the IC20 databook.

CAN Serial Linked I/O device (SLIO) with digital and analog port functions

P82C150

1 FEATURES

- Single-chip I/O device with CAN protocol controller
- Meets CAN protocol specification version 2.0 A and B (passive) with restricted bit timing
- Fully integrated clock oscillator (no crystal required)
- 16 configurable digital or analog I/O port pins
- Each of the port pins individually configurable via CAN-bus: port direction, port mode and event capture facilities for inputs (event driven or polling)
- Up to sixteen digital inputs; automatic transmission of a CAN message on a change on inputs individually selectable
- Up to sixteen 3-state outputs
- Up to two quasi-analog outputs with 10-bit accuracy
- 10-bit analog-to-digital converter with up to six multiplexed analog input channels (for accuracy see Section 11.6)
- Two general purpose comparators
- Bit rate from 20 kbit/s up to 125 kbit/s using internal oscillator
- Automatic bit rate detection and calibration
- Up to sixteen P82C150 nodes for one CAN-bus system
- Four identifier bits programmable
- SLIO functions controlled by a single intelligent node ('host')
- Sleep-mode with wake-up via CAN-bus
- Differential CAN-bus input comparator and CAN-bus output driver
- Supply voltage: 5 V ±4%
- Operating temperature: two ranges −40 to +85 °C and −40 to +125 °C.

2 GENERAL DESCRIPTION

The P82C150 is a single-chip 16-bit I/O device including a Controller Area Network (CAN) protocol controller with automatic bit rate detection and calibration. It features 16 configurable I/O port pins with programmable direction, digital and analog modes.

The P82C150 provides a configurable event capture facility supporting automatic transmission caused by a change on the port input pins.

The clock oscillator requires no external components, thus, the cost of the CAN link is reduced significantly.

The P82C150 is a very cost-effective way to increase the I/O capability of a microcontroller based CAN node as well as to reduce the amount and complexity of wiring. Advanced safety is provided by the CAN protocol.

Applications:

- Body electronics and instrumentation in automotive applications
- Sensor/actuator interface in automotive and general industrial applications
- Extension of I/O capabilities of microcontroller based CAN nodes.

3 ORDERING INFORMATION

TYPE NUMBER	PACKAGE			TEMPERATURE RANGE (°C)
	NAME	DESCRIPTION	VERSION	
P82C150 AFT	SO28	plastic small outline package; 28 leads; body width 7.5 mm	SOT136-1	−40 to +85
P82C150 AHT				−40 to +125

4 BLOCK DIAGRAM

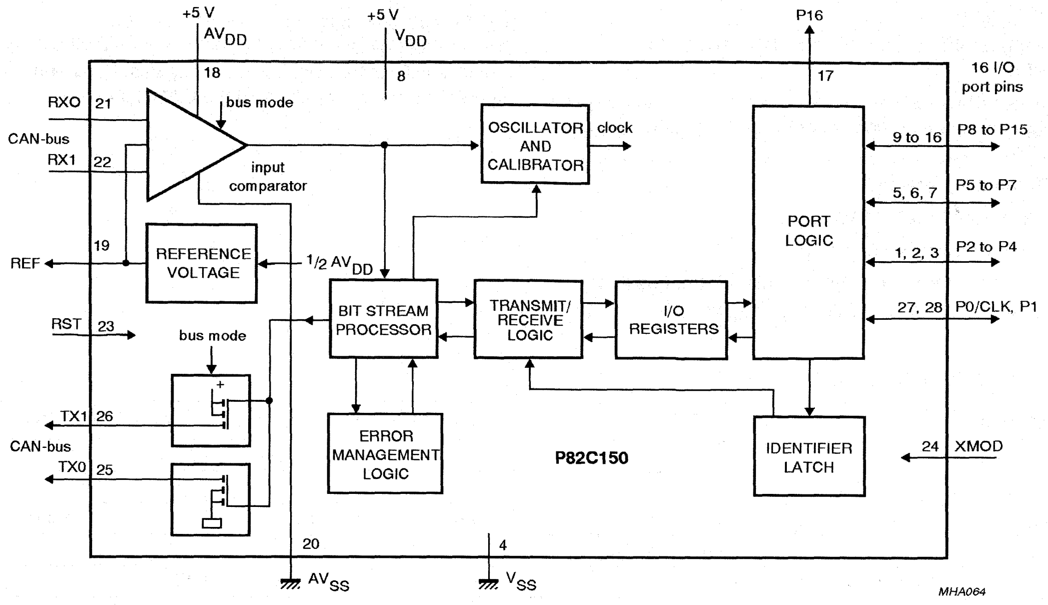


Fig.1 Block diagram.

CAN controller interface | PCA82C250

FEATURES

- Fully compatible with the *"ISO/DIS 11898"* standard
- High speed (up to 1 Mbaud)
- Bus lines protected against transients in an automotive environment
- Slope control to reduce radio frequency interference (RFI)
- Differential receiver with wide common-mode range for high immunity against electromagnetic interference (EMI)
- Thermally protected
- Short-circuit proof to battery and ground
- Low current standby mode
- An unpowered node does not disturb the bus lines
- At least 110 nodes can be connected.

APPLICATIONS

- High-speed applications (up to 1 Mbaud) in cars.

GENERAL DESCRIPTION

The PCA82C250 is the interface between the CAN protocol controller and the physical bus. The device provides differential transmit capability to the bus and differential receive capability to the CAN controller.

QUICK REFERENCE DATA

SYMBOL	PARAMETER	CONDITIONS	MIN.	MAX.	UNIT
V_{CC}	supply voltage		4.5	5.5	V
I_{CC}	supply current		–	170	µA
$1/t_{bit}$	maximum transmission speed	non-return-to-zero	1	–	Mbaud
V_{CAN}	CANH, CANL input/output voltage		–8	+18	V
ΔV	differential bus voltage		1.5	3.0	V
t_{pd}	propagation delay	high-speed mode	–	50	ns
T_{amb}	operating ambient temperature		–40	+125	°C

ORDERING INFORMATION

TYPE NUMBER	PACKAGE		
	NAME	MATERIAL	CODE
PCA82C250	DIP8	plastic dual in-line package; 8 leads (300 mil)	SOT97-1
PCA82C250T	SO8	plastic small outline package; 8 leads; body width 3.9 mm	SOT96-1

BLOCK DIAGRAM

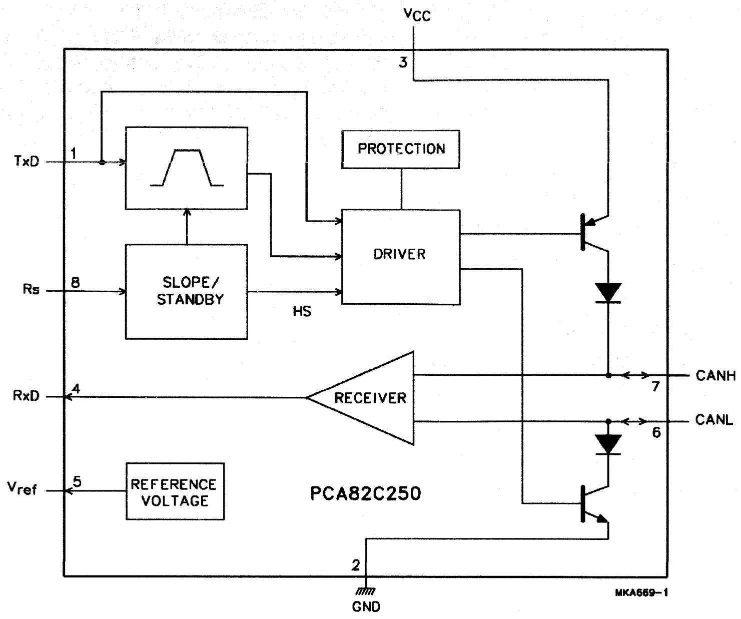


Fig.1 Block diagram.

PINNING

SYMBOL	PIN	DESCRIPTION
TxD	1	transmit data input
GND	2	ground
V_{CC}	3	supply voltage
RxD	4	receive data output
V_{ref}	5	reference voltage output
CANL	6	LOW level CAN voltage input/output
CANH	7	HIGH level CAN voltage input/output
Rs	8	slope resistor input

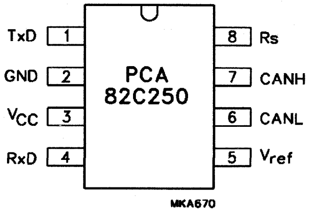


Fig.2 Pin configuration.

Stand-alone CAN controller SJA1000

1 FEATURES

- Pin compatibility to the PCA82C200 stand-alone CAN controller
- Electrical compatibility to the PCA82C200 stand-alone CAN controller
- Software-compatibility mode to the PCA82C200 (BasicCAN mode is default)
- Extended receive buffer (64-byte FIFO)
- CAN 2.0B protocol compatibility (extended frame passive in PCA82C200 compatibility mode)
- Supports 11-bit identifier as well as 29-bit identifier
- Bit rates up to 1 Mbits/s
- PeliCAN mode extensions:
  - Error counters with read/write access
  - Programmable error warning limit
  - Last error code register
  - Error interrupt for each CAN-bus error
  - Arbitration lost interrupt with detailed bit position
  - Single-shot transmission (no re-transmission)
  - Listen only mode (no acknowledge, no active error flags)
  - Hot plugging support (software driven bit rate detection)
  - Acceptance filter extension (4-byte code, 4-byte mask)
  - Reception of ‘own’ messages (self reception request)
- 24 MHz clock frequency
- Interfaces to a variety of microprocessors
- Programmable CAN output driver configuration
- Extended ambient temperature range (–40 to +125 °C).

2 GENERAL DESCRIPTION

The SJA1000 is a stand-alone controller for the Controller Area Network (CAN) used within automotive and general industrial environments. It is designed to be hardware and software compatible to the PCA82C200 CAN controller (BasicCAN) from Philips Semiconductors. Additionally, a new mode of operation is implemented (PeliCAN) which supports the CAN 2.0B protocol specification with several new features.

3 ORDERING INFORMATION

TYPE NUMBER	PACKAGE		
	NAME	DESCRIPTION	VERSION
SJA1000	DIP28	plastic dual in-line package; 28 leads (600 mil)	SOT117-1
SJA1000T	SO28	plastic small outline package; 28 leads; body width 7.5 mm	SOT136-1

4 BLOCK DIAGRAM

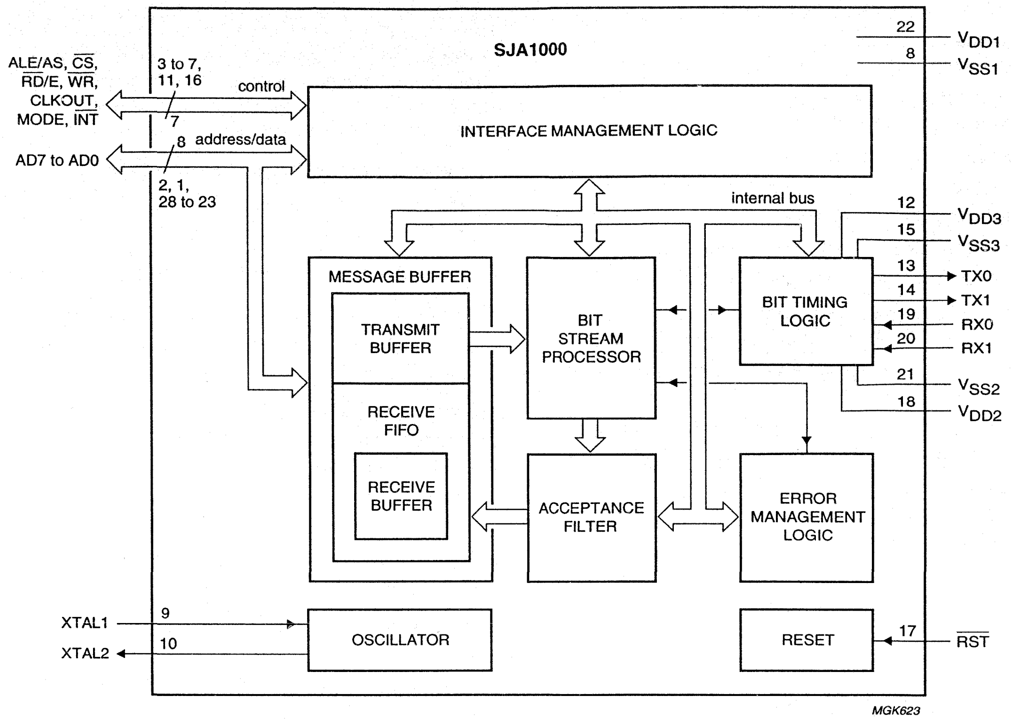


Fig.1 Block diagram.

8-bit microcontroller with on-chip CAN P8xC592

1 FEATURES

- 80C51 central processing unit (CPU)
- 16 kbytes on-chip ROM, externally expandible to 64 kbytes
- 2 × 256 bytes on-chip RAM, externally expandible to 64 kbytes
- Two standard 16-bit timers/counters
- One additional 16-bit timer/counter coupled to four capture and three compare registers
- 10-bit ADC with 8 multiplexed analog inputs
- Two 8-bit resolution Pulse Width Modulated outputs
- 15 interrupt sources with 2 priority levels (2 to 6 external interrupt sources possible)
- Five 8-bit I/O ports, plus one 8-bit input port shared with analog inputs
- CAN-controller (CAN = Controller Area Network) with DMA data transfer facility to internal RAM
- 1 Mbit/s CAN-controller with bus failure management facility
- $\frac{1}{2}AV_{DD}$ reference voltage
- Full-duplex UART compatible with the standard 80C51
- On-chip Watchdog Timer (WDT)
- 1.2 to 16 MHz clock frequency.

2 GENERAL DESCRIPTION

The P8xC592 is a single-chip 8-bit high-performance microcontroller with on-chip CAN-controller, derived from the 80C51 microcontroller family. It uses the powerful 80C51 instruction set.
Figure 1 shows a block diagram of the P8xC592.

The P8xC592 is manufactured in an advanced CMOS process, and is designed for use in automotive and general industrial applications. In addition to the 80C51 standard features, the device provides a number of dedicated hardware functions for these applications.

Two versions of the P8xC592 will be offered:

- P80C592 (without ROM)
- P83C592 (with ROM).

Hereafter these versions will be referred to as P8xC592.

The temperature range includes (max. f_{CLK} = 16 MHz):

- −40 to +85 °C version, for general applications
- −40 to +125 °C version for automotive applications.

The P8xC592 combines the functions of the P8xC552 (microcontroller) and the PCA82C200 (Philips CAN-controller) with the following enhanced features:

- 16 kbytes Program Memory
- 2 × 256 bytes Data Memory
- DMA between CAN Transmit/Receive Buffer and internal RAM.

The main differences between P8xC592 and P8xC552 are:

- 16 kbytes programmable ROM (P8xC552 has 8 kbytes)
- Additional 256 bytes RAM
- A CAN-controller instead of the I^2C-serial interface.

3 ORDERING INFORMATION

TYPE NUMBER	PACKAGE			TEMPERATURE RANGE (°C)	FREQ. (MHz)
	NAME	DESCRIPTION	VERSION		
Without ROM					
P80C592FFA	PLCC68	plastic leaded chip carrier; 68 leads	SOT188-2	−40 to +85	1.2 to 16
P80C592FHA				−40 to +125	
With ROM					
P83C592FFA	PLCC68	plastic leaded chip carrier; 68 leads	SOT188-2	−40 to +85	1.2 to 16
P83C592FHA				−40 to +125	

4 BLOCK DIAGRAM

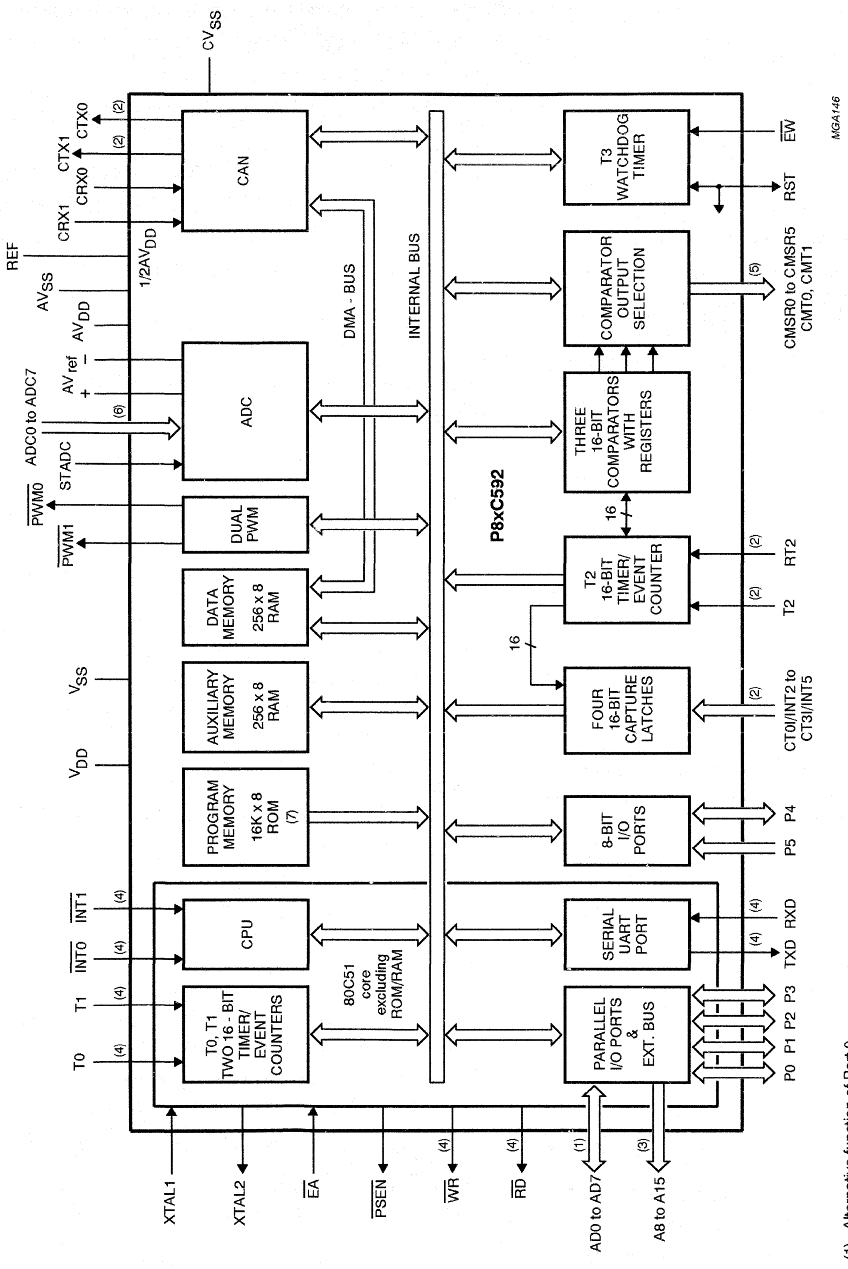


Fig.1 Block diagram.

(1) Alternative function of Port 0.
(2) Alternative function of Port 1.
(3) Alternative function of Port 2.
(4) Alternative function of Port 3.
(5) Alternative function of Port 4.
(6) Alternative function of Port 5.
(7) Not present in P80C592.

1 FEATURES

- 80C51 central processing unit (CPU)
- 32 kbytes on-chip ROM, externally expandible to 64 kbytes
- 2 × 256 bytes on-chip RAM, externally expandible to 64 kbytes
- Two standard 16-bit timers/counters
- One additional 16-bit timer/counter coupled to four capture and three compare registers
- 10-bit ADC with 8 multiplexed analog inputs
- Two 8-bit resolution Pulse Width Modulated outputs
- 15 interrupt sources with 2 priority levels (2 to 6 external interrupt sources possible)
- Five 8-bit I/O ports, plus one 8-bit input port shared with analog inputs
- CAN-controller (CAN = Controller Area Network) with DMA data transfer facility to internal RAM
- 1 Mbit/s CAN-controller with bus failure management facility
- ½AV_{DD} reference voltage
- Full-duplex UART compatible with the standard 80C51
- On-chip Watchdog Timer (WDT)
- 1.2 to 16 MHz clock frequency
- Improved Electromagnetic Compatibility (EMC).

2 GENERAL DESCRIPTION

The P8xCE598 is a single-chip 8-bit high-performance microcontroller with on-chip CAN-controller, derived from the 80C51 microcontroller family.
It uses the powerful 80C51 instruction set.
Figure 1 shows a block diagram of the P8xCE598.

The P8xCE598 is manufactured in an advanced CMOS process, and is designed for use in automotive and general industrial applications. In addition to the 80C51 standard features, the device provides a number of dedicated hardware functions for these applications.

Two versions of the P8xCE598 will be offered:

- P80CE598 (without ROM)
- P83CE598 (with ROM)

Hereafter these versions will be referred to as P8xCE598.

The temperature range includes (max. f_{CLK} = 16 MHz):

- −40 to +85 °C version, for general applications
- −40 to +125 °C version for automotive applications.

The P8xCE598 combines the functions of P8XC552 (microcontroller) and the PCA82C200 (Philips CAN-controller) with the following enhanced features:

- 32 kbytes Program Memory
- 2 × 256 bytes Data Memory
- DMA between CAN Transmit/Receive Buffer and internal RAM.

The main differences to the P8xC552 microcontroller are:

- 32 kbytes programmable ROM (P8xC552 has 8 kbytes)
- Additional 256 bytes RAM
- A CAN-controller instead of the I^2C-serial interface.

2.1 Electromagnetic Compatibility (EMC)

Primary attention is paid to the reduction of electromagnetic emission of the microcontroller P8xCE598. The following features reduce the electromagnetic emission and additionally improve the electromagnetic susceptibility:

- One analog part power supply pin (AV_{DD}) and one analog part ground pin (AV_{SS}), placed as a pair of pins on one side of the package (see Fig.3), providing power supply (+5V) and ground for ADC, CAN receiver and reference voltage.
- Four digital part supply voltage pins (V_{DD1} to V_{DD4}) and four digital part ground pins (V_{SS1} to V_{SS4}) are provided on the package. These pins, one V_{DD} and one V_{SS} as a pair of pins are placed on each of the four sides of the package to provide:
  - V_{DD1}/V_{SS1} for internal logic (CPU, Timers/counters, Memory, CAN, UART, ADC)
  - V_{DD2}/V_{SS2} for Port 1, Port 3 and Port 4, and $\overline{\text{PWM0}}$ and $\overline{\text{PWM1}}$ outputs
  - V_{DD3}/V_{SS3} for the on-chip oscillator
  - V_{DD4}/V_{SS4} for the Port 0, Port 2, ALE output and $\overline{\text{PSEN}}$ output.
- External capacitors should be connected across associated V_{DDx} and V_{SSx} pins (i.e. V_{DD1} and V_{SS1}). Lead length should be as short as possible. Ceramic chip capacitors are recommended (100 nF).
- One CAN supply voltage pin (CV_{DD}) and one CAN ground pin (CV_{SS}) as a pair of pins placed on one side of the package providing (digital part) power supply (+5V) and ground for the CAN transmitter outputs.
- Internal decoupling capacitance improves the EMC radiation behaviour and the EMC immunity.

2.2 Recommendation on ALE

For application that require no external memory or temporarily no external memory: the ALE output signal (pulses at a frequency of $\frac{1}{6} f_{OSC}$) can be disabled under software control (bit 5 in PCON SFR: **'RFI**'); if disabled, no ALE pulse will occur. ALE pin will be pulled down internally, switching an external address latch to a quiet state. The MOVX instruction will still toggle ALE as a normal MOVX.

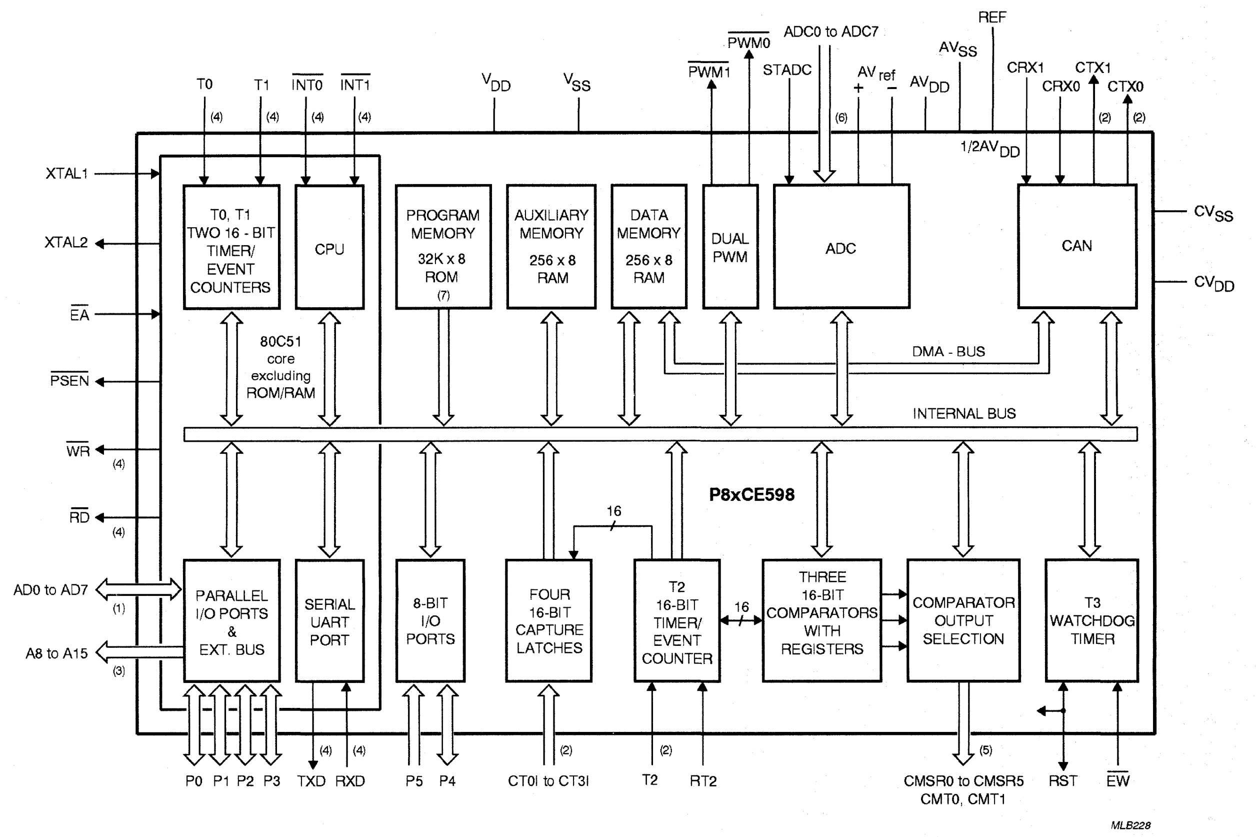
ALE will retain its normal HIGH value during Idle mode and a LOW value during Power-down mode while in the '**RFI** reduction mode'.

Additionally during internal access ($\overline{EA}$ = 1) ALE will toggle normally when the address exceeds the internal Program Memory size. During external access ($\overline{EA}$ = 0) ALE will always toggle normally, whether the flag **'RFI'** is set or not.

3 ORDERING INFORMATION

TYPE NUMBER	PACKAGE			TEMPERATURE RANGE (°C)	FREQ. (MHz)
	NAME	DESCRIPTION	VERSION		
Without ROM					
P80CE598FFB	QFP80	plastic quad flat package; 80 leads (lead length 1.95 mm); body 14 × 20 × 2.7 mm; high stand-off height	SOT318-1	−40 to +85	1.2 to 16
P80CE598FHB				−40 to +125	
With ROM					
P83CE598FFB	QFP80	plastic quad flat package; 80 leads (lead length 1.95 mm); body 14 × 20 × 2.7 mm; high stand-off height	SOT318-1	−40 to +85	1.2 to 16
P83CE598FHB				−40 to +125	

4 BLOCK DIAGRAM



(1) Alternative function of Port 0.
(2) Alternative function of Port 1.
(3) Alternative function of Port 2.
(4) Alternative function of Port 3.
(5) Alternative function of Port 4.
(6) Alternative function of Port 5.
(7) Not present in P80CE598.

Fig.1 Block diagram.

Section 10
80C51 Support Chips

CONTENTS

PCF1252–X family 299

Threshold detector and reset generator PCF1252-X family

FEATURES

- Very low current consumption, typically 10 μA
- 10 factory programmed threshold voltages available covering trip voltages from 4.75 to 2.55 V
- ±50 mV trip point accuracy over full temperature range
- Variable RESET delay
- RESET pulse polarity selection
- Defined outputs at 0.6 V (typ.)
- Comparator for second level detection (e.g. overvoltage detection)
- Advance warning of power fail
- Operating temperature range –40 to +85 °C.

GENERAL DESCRIPTION

The PCF1252-Xs are low-power CMOS voltage threshold detectors designed especially for supervision of microcontroller/microprocessor systems for detection of power-on/off conditions and generation of a system reset pulse. The PCF1252-X also provides a $\overline{\text{POWF}}$ (power fail) output which is activated at a precise factory-programmed trip point. A system RESET output has a built-in delay with duration determined by an external capacitor (C_{CT}).

A second comparator (comparator 2) has been included to enable the possibility of a second monitoring point in the system.

ORDERING INFORMATION

TYPE NUMBER	PACKAGE		
	NAME	DESCRIPTION	VERSION
PCF1252-XP[1]	DIP8	plastic dual in-line package; 8 leads (300 mil)	SOT97-1
PCF1252-XT[1]	SO8	plastic small outline package; 8 leads; body width 3.9 mm	SOT96-1

Note

1. X = 0 to 9; depending on threshold voltage.

BLOCK DIAGRAM

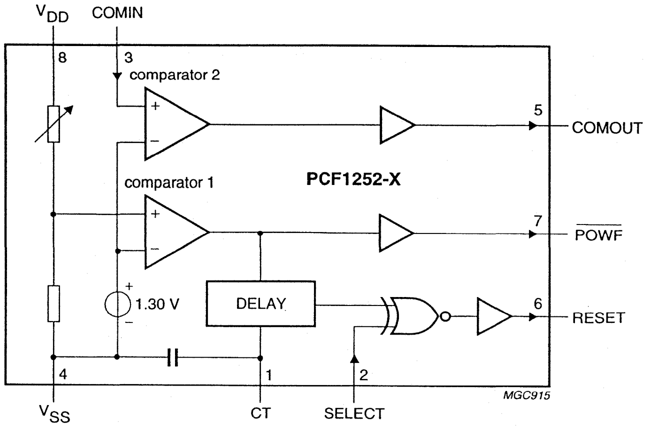


Fig.1 Block diagram.

Section 11
Development Support

CONTENTS

Microcontroller development tool contact information 303

8-bit microcontroller demonstration and evaluation boards 305

MICROCONTROLLER DEVELOPMENT TOOL CONTACT INFORMATION

Company	World Wide Web Address	Telephone	Fax
Software: Compilers, Assemblers, Simulators			
2500AD Software, Inc.	www.2500ad.com	1–719–395–8683	719–395–8206
Archimedes Software, Inc.	www.archimedesinc.com	1–206–822–6300	1–206–822–8632
Avocet Systems, Inc.	www.midcoast.com/~avocet	1–207–236–9055	1–207–236–6713
Dunfield Development Systems	www.dunfield.com/	1–613–256–5820	1–613–256–5821
HITECH Software	www.htsoft.com	+61–7–3300–5011	+61–7–3300–5246
Keil Software, Inc.	www.keil.com	1–214–735–8052	1–214–735–8055
Production Languages Corp.	www.plcorp.com/	1–817–599–8363	1–817–599–5098
Tasking	www.tasking.nl/	1–781–320–9400	1–781–320–9212
Emulators			
Ashling (U.S.: Orion Instruments)	www.ashling.com	1–408–747–0440	1–408–747–0688
Ashling Microsystems (Ireland)	www.ashling.com	+353–61–334466	+353–61–334477
Ceibo (ISRAEL)	www.ceibo.com	+972.9.9555387	+972.9.9553297
Ceibo (USA)	www.ceibo.com	1–314–830–4084	1–314–830–4083
Lauterbach	www.lauterbach.com	1–508–620–4521	1–508–620–4522
MetaLink Corp.	www.metaice.com	1–602–926–0797	1–602–926–1198
Microtek International Inc.	www.microtekintl.com	+886.35.772155	+886.35.772598
Nohau Corp. (USA)	www.nohau.com/nohau	1–408–866–1820	1–408–378–7869
Nohau Elektronik AB ()	www.nohau.com/nohau	+46 40 592200	+46 40 592229
Signum Systems	www.signum.com	1–805–371–4608	1–805–371–4610
Programmers			
Advin Systems	www.wco.com/~advin/	1–408–243–7000	–
BP Microsystems	www.bpmicro.com/	1–713–461–9430	–
Ceibo (ISRAEL)	www.ceibo.com	+972.9.9555387	+972.9.9553297
Ceibo (USA)	www.ceibo.com	1–314–830–4084	1–314–830–4083
Data I/O Corp.	sirius.data–io.com/	1–206–881–6444	–
Needham's Electronics	www.needhams.com/	1–916–924–8037	–
Programming Adapters			
EDI Corp.	–	1–702–735–4997	1–702–735–8339
Emulation Technology, Inc.	www.emulation.com	1–408–982–0660	1–408–982–0664
Ironwood Electronics	www.ironwoodelectronics.com	1–612–452–8100	1–612–452–8400
Logical Systems	www.logicalsys.com	1–315–478–0722	1–315–479–6753

Microcontroller development tool contact information

I^{2}C			
Micro Computer Control (MCC)	www.mcc–us.com/	1–609–466–1751	1–609–466–4116
Real–Time Operating Systems			
CMX	www.cmx.com	1–508–872–7675	1–508–620–6828
Embedded Systems Products (ESP)	www.esphou.com	1–800–525–4302 1–713–561–9990	1–713–561–9980
Development Boards			
Ceibo (ISRAEL)	www.ceibo.com	+972.9.9555387	+972.9.9553297
Ceibo (USA)	www.ceibo.com	1–314–830–4084	1–314–830–4083
Future Designs, Inc.	members.aol.com/teamfdi/teamfdi.htm	1–205–830–4116	1–205–830–9421

8-bit microcontroller demonstration and evaluation boards

PRODUCT	DESCRIPTION
OM4151, S87C00K	I^{2}C demonstration board based on 80C51 derivatives
OM4238, P8051DB	8051 family demonstration board
OM4128	8XC552 evaluation board PEB552
OM4130, PCAN-EVAL	CAN controller evaluation board
OM4239	8XC592 evaluation board PEB592
OM4240	8XCE598 evaluation board PEB598
OM4241	8XCE598 evaluation board PDB598
OM4160, SM68070	68070 and 66470 demonstration and evaluation board Microcore 1
OM4160/2	68070 evaluation board Microcore 2
OM4162	9XCE201 evaluation board Microcore 4
OM4280, P83C852DEM	83C852 demonstration kit
OM4281 [1]	83C852 software evaluation kit
P8051DB	80C51 family development board
OM4717	83CL410 solar powered demonstration board
OM5005, P80CLEVAL	80CL51 evaluation board
DS750	8XC750 microcontroller in-circuit emulation development tool

NOTE:

1. The OM4281 is now available only from Ashling Microsystems Ltd. as type SCPC4281.

Section 12
Package Outlines

80C51-Based
8-Bit Microcontrollers

CONTENTS

Soldering Package information ... 309

Plastic Dual In-Line Package

DIP8: plastic dual in-line package; 8 leads (300 mil) ... SOT97-1 311
DIP24: plastic dual in-line package; 24 leads (300 mil) ... SOT222-1 ... 312
DIP28: plastic dual in-line package; 28 leads (600 mil); long body ... SOT117-2.... 313
DIP40: plastic dual in-line package; 40 leads (600 mil) ... SOT129-1 ... 314

Plastic Shrink Dual In-Line Package

SDIP42: plastic shrink dual in-line package; 42 leads (600 mil) ... SOT270-1 ... 315

Plastic Leaded Chip Carrier

PLCC28: plastic leaded chip carrer; 28 leads; pedestal ... SOT261-3 ... 316
PLCC44: plastic leaded chip carrier; 44 leads ... SOT187-2 ... 317
PLCC68: plastic leaded chip carrier; 68 leads ... SOT188-2 ... 318
PLCC68: plastic leaded chip carrier; 68 leads; pedestal ... SOT188-3 ... 319

Plastic Quad Flat Package

QFP44: plastic quad flat package; 44 leads (lead length 1.3 mm); body 10 x 10 x 1.75 mm ... SOT307-2 ... 320
QFP44: plastic quad flat package; 44 leads (lead length 2.35 mm); body 14 x 14 x 2.2 mm ... SOT205-1 ... 321
LQFP44: plastic low profile quad flat package; 44 leads; body 10 x 10 x 1.4 mm ... SOT389-1 ... 322
QFP64: plastic quad flat package; 64 leads (lead length 1.95mm); body 14 x 20 x 2.8 mm ... SOT319-2 ... 323
QFP80: plastic quad flat package; 80 leads (lead length 1.95 mm); body 14 x 20 x 2.7 mm; high stand-off height ... SOT318-1 ... 324
QFP80: plastic quad flat package; 80 leads (lead length 1.95 mm); body 14 x 20 x 2.8 mm ... SOT318-2 ... 325

Plastic Small Outline Package

SO8: plastic small outline package; 8 leads; body width 3.9mm ... SOT96-1 326
VSO40: plastic very small outline package; 40 leads ... SOT158-1 ... 327
VSO56: plastic very small outline package; 56 leads ... SOT190-1 ... 328
SO28 plastic small outline package; 28 leads; body width 7.5mm ... SOT136–1 ... 329

Plastic Shrink Small Outline Package

SSOP24: plastic shrink small outline package; 24 leads; body width 5.3 mm ... SOT340-1 ... 330
SSOP28: plastic shrink small outline package; 28 leads; body width 5.3mm ... SOT341-1 ... 331

INTRODUCTION

There is no soldering method that is ideal for all IC packages. Wave soldering is often preferred when though-hole and surface mounted components are mixed on one printed-circuit board. However, wave soldering is not always suitable for surface mounted ICs, or for printed-circuits with high population densities. In these situations reflow soldering is often used.

This text gives a very brief insight to a complex technology. A more in-depth account of soldering ICs can be found in our *"IC Package Databook"* (order code 9398 652 90011).

THROUGH-HOLE MOUNTED PACKAGES

Table 1. Types of through-hole mounted packages

TYPE	DESCRIPTION
DIP	plastic dual in-line package
SDIP	plastic shrink dual in-line package
HDIP	plastic heat-dissipating dual in-line package
DBS	plastic dual in-line bent from a single in-line package
SIL	plastic single in-line package

Soldering by dipping or wave

The maximum permissible temperature of the solder is 260°C; solder at 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 to the seating plane, but the temperature of the plastic body must not exceed the specified maximum storage temperature ($T_{stg\ max}$). 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.

Repairing soldered joints

Apply a low voltage soldering iron (less than 24V) to the lead(s) of the package, below the seating plane or not more than 2mm above it. If the temperature of the soldering iron bit is less than 300°C it may remain in contact for up to 10 seconds. If the bit temperature is between 300 and 400°C, contact may be up to 5 seconds.

SURFACE MOUNTED PACKAGES

Table 2. Types of surface mounted packages

TYPE	DESCRIPTION
SO	plastic small outline package
SSOP	plastic shrink small outline package
TSSOP	plastic thin shrink small outline package
VSO	plastic very small outline package
QFP	plastic quad flat package
LQFP	plastic low profile quad flat package
SQFP	plastic shrink quad flat package
TQFP	plastic thin quad flat package
PLCC	plastic leaded chip carrier

Reflow soldering

Reflow soldering techniques are suitable for all SMD packages, ease of soldering varies with the type of package as indicated in Table 3.

The choice of heating method may be indluenced by larger plastic packages (QFP or PLCC with 44 leads, or more). If infrared or vapor phase heating is used and the large packages are not absolutely dry (less than 0.1% moisture content by weight), vaporization of the small amount of moisture in them can cause cracking of the plastic body. For more information on moisture prevention, refer to the Drypack chapter in our *"Quality Reference Manual"* (order code 9398 510 63011).

Reflow soldering requires solder paste (a suspension of fine solder particles, flux and binding agent) to be applied to the printed-circuit board by screen printing, stenciling or pressure-syringe dispensing before package placement.

Several techniques exist for reflowing; for example, thermal conduction by heated belt. Dwell times vary between 50 and 300 seconds depending on heating method. Typical reflow temperatures range from 215 to 250°C.

Preheating is necessary to dry the paste and evaporate the binding agent. Preheating duration: 45 minutes at 45°C.

Table 3. Suitability of surface mounted packages for various soldering methods

Rating from ‘a’ to ‘d’: **‘a’ indicates most suitable** (soldering is not difficult); **‘d’ indicates least suitable** (soldering is achievable with difficulty).

TYPE	REFLOW METHOD					DOUBLE WAVE METHOD
	INFRARED	HOT BELT	HOT GAS	VAPOR PHASE	RESISTANCE	
SO	a	a	a	a	d	a
SSOP	a	a	a	c	d	c
TSSOP	b	b	b	c	d	d
VSO	b	b	a	b	a	b
QFP	b	b	a	c	a	c
LQFP	b	b	a	c	d	d
SQFP	b	b	a	c	d	d
TQFP	b	b	a	c	d	d
PLCC	c	b	b	d	d	b

Wave soldering

Wave soldering is **not** recommended for SSOP, TSSOP, QFP, LQFP, SQFP or TQFP packages. This is because of the likelihood of solder bridging due to closely-spaced leads and the possibility of incomplete solder penetration in multi-lead devices.

If wave soldering cannot be avoided, the following conditions must be observed:

- A double-wave (a turbulent wave with high upward pressure followed by a smooth laminar wave) soldering technique should be used.
- For SSOP, TSSOP and VSO packages, the longitudinal axis of the package footprint must be parallel to the solder flow **and** must incorporate solder theives at the downstream end.
- For QFP, LQFP and TQFP packages, the footprint must be at and angle of 45° to the board direction **and** must incorporate solder thieves downstream and at the side corners.

Even with these conditions, only consider wave soldering for the following package types:

- SO
- VSO
- PLCC
- SSOP **only with body width 4.4mm**, e.g., SSOP16 (SOT369-1) or SSOP20 (SOT266-1).
- QFP **except** QFP52 (SOT379-1), QFP100 (SOT317-1, SOT317-2 and SOT382-1) and QFP160 (SOT322-1); these are **not** suitable for wave soldering.
- LQFP **except** LQFP32 (SOT401-1), LQFP48 (SOT313-1, SOT313-2), LQFP64 (SOT314-2), LQFP80 (SOT315-1); these are **not** suitable for wave soldering.
- TQFP **except** TQFP64 (SOT357-1), TQFP80 (SOT375-1) and TQFP100 (SOT386-1); these are **not** suitable for wave soldering.

SQFP are **not** suitable for wave soldering.

During placement and before soldering, the package must be fixed with a droplet of adhesive. The adhesive can be applied by screen printing, pin transfer or syringe dispensing. The package can be soldered after the adhesive is cured.

Maximum permissible solder temperature is 260°C, and maximum duration of package immersion in solder is 10 seconds, if cooled to less than 150°C within 6 seconds. Typical dwell time is 4 seconds at 250°C.

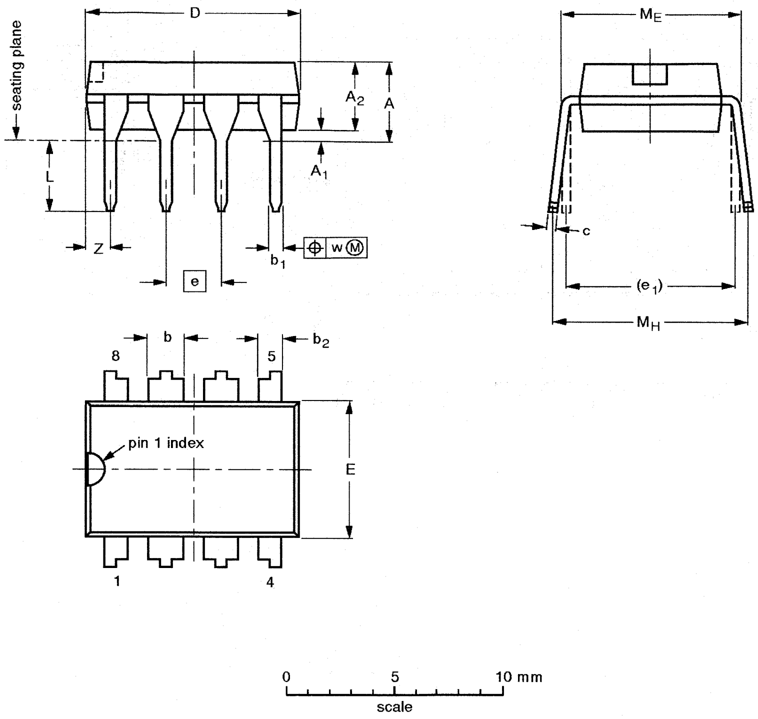
A mildly-activated flux will eliminate the need for removal of corrosive residues in most applications.

Repairing soldered joints

Fix the component by first soldering two diagonally-opposite end leads. Use only a low voltage soldering iron (less than 24V) applied to the flat part of the lead. Contact time must be limited to 10 seconds at up to 300°C. When using a dedicated tool, all other leads can be soldered in one operation within 2 to 5 seconds at between 270 and 320°C.

Package outlines

DIP8: plastic dual in-line package; 8 leads (300 mil) SOT97-1



DIMENSIONS (inch dimensions are derived from the original mm dimensions)

UNIT	A max.	A_1 min.	A_2 max.	b	b_1	b_2	c	D[1]	E[1]	e	e_1	L	M_E	M_H	w	Z[1] max.
mm	4.2	0.51	3.2	1.73 1.14	0.53 0.38	1.07 0.89	0.36 0.23	9.8 9.2	6.48 6.20	2.54	7.62	3.60 3.05	8.25 7.80	10.0 8.3	0.254	1.15
inches	0.17	0.020	0.13	0.068 0.045	0.021 0.015	0.042 0.035	0.014 0.009	0.39 0.36	0.26 0.24	0.10	0.30	0.14 0.12	0.32 0.31	0.39 0.33	0.01	0.045

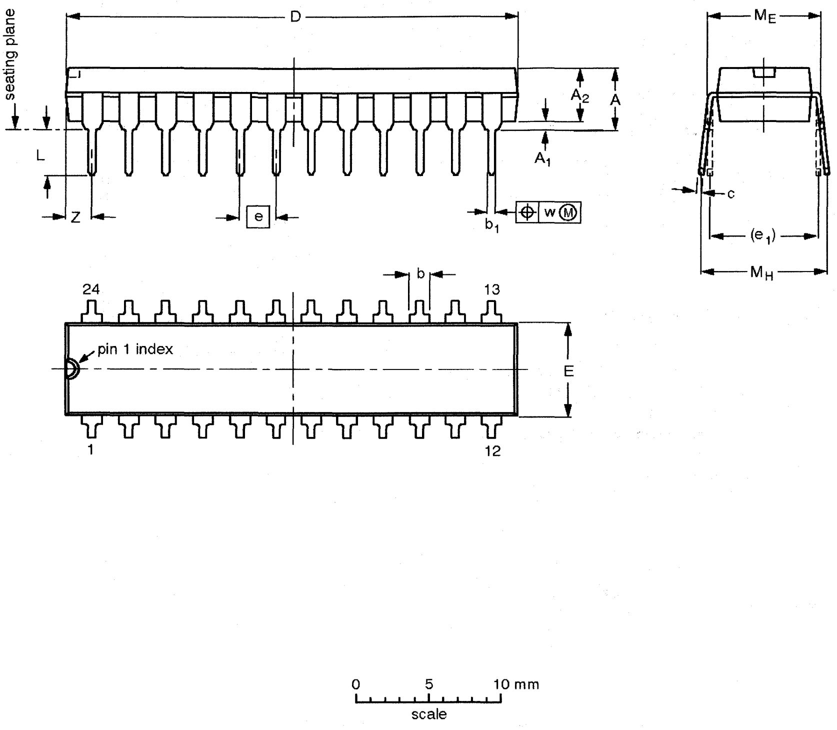
Note

1. Plastic or metal protrusions of 0.25 mm maximum per side are not included.

OUTLINE VERSION	REFERENCES				EUROPEAN PROJECTION	ISSUE DATE
	IEC	JEDEC	EIAJ			
SOT97-1	050G01	MO-001AN				~~92-11-17~~ 95-02-04

Package outlines

DIP24: plastic dual in-line package; 24 leads (300 mil) SOT222-1



DIMENSIONS (millimetre dimensions are derived from the original inch dimensions)

UNIT	A max.	A_1 min.	A_2 max.	b	b_1	c	D(1)	E(1)	e	e_1	L	M_E	M_H	w	Z(1) max.
mm	4.70	0.38	3.94	1.63 1.14	0.56 0.43	0.36 0.25	31.9 31.5	6.73 6.48	2.54	7.62	3.51 3.05	8.13 7.62	10.03 7.62	0.25	2.05
inches	0.185	0.015	0.155	0.064 0.045	0.022 0.017	0.014 0.010	1.256 1.240	0.265 0.255	0.100	0.300	0.138 0.120	0.32 0.30	0.395 0.300	0.01	0.081

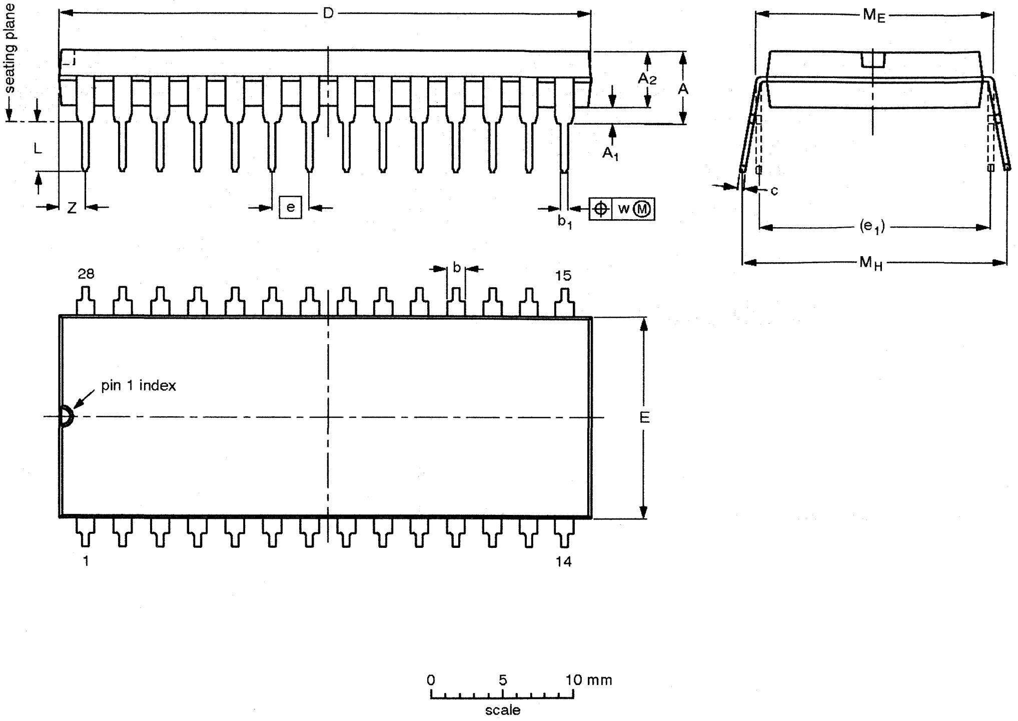
Note

1. Plastic or metal protrusions of 0.01 inches maximum per side are not included.

OUTLINE VERSION	REFERENCES				EUROPEAN PROJECTION	ISSUE DATE
	IEC	JEDEC	EIAJ			
SOT222-1		MS-001AF				95-03-11

Package outlines

DIP28: plastic dual in-line package; 28 leads (600 mil); long body SOT117-2



DIMENSIONS (millimetre dimensions are derived from the original inch dimensions)

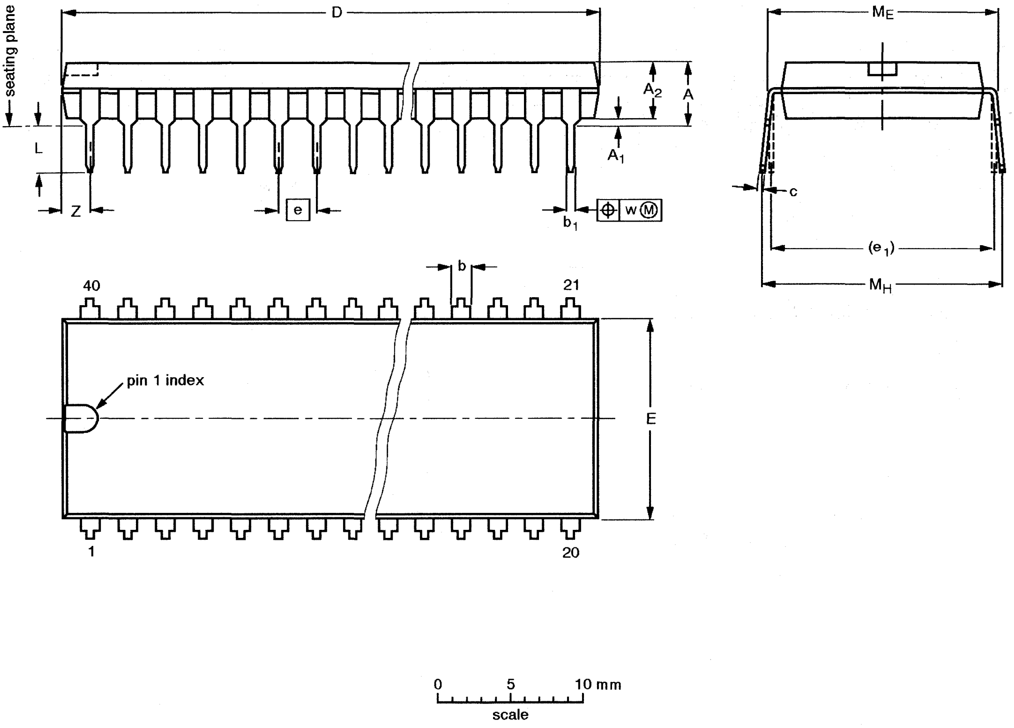
UNIT	A max.	A_1 min.	A_2 max.	b	b_1	c	D[(1)]	E[(1)]	e	e_1	L	M_E	M_H	w	Z[(1)] max.
mm	5.08	0.51	3.94	1.63 1.14	0.56 0.43	0.38 0.25	37.08 35.94	14.22 13.84	2.54	15.24	3.51 3.05	15.75 15.24	17.65 15.24	0.25	2.10
inches	0.200	0.020	0.155	0.064 0.045	0.022 0.017	0.015 0.010	1.460 1.415	0.560 0.545	0.100	0.600	0.138 0.120	0.62 0.60	0.695 0.600	0.01	0.083

Note

1. Plastic or metal protrusions of 0.01 inches maximum per side are not included.

OUTLINE VERSION	REFERENCES				EUROPEAN PROJECTION	ISSUE DATE
	IEC	JEDEC	EIAJ			
SOT117-2		MS-011AB				95-03-11

DIP40: plastic dual in-line package; 40 leads (600 mil) SOT129-1



DIMENSIONS (inch dimensions are derived from the original mm dimensions)

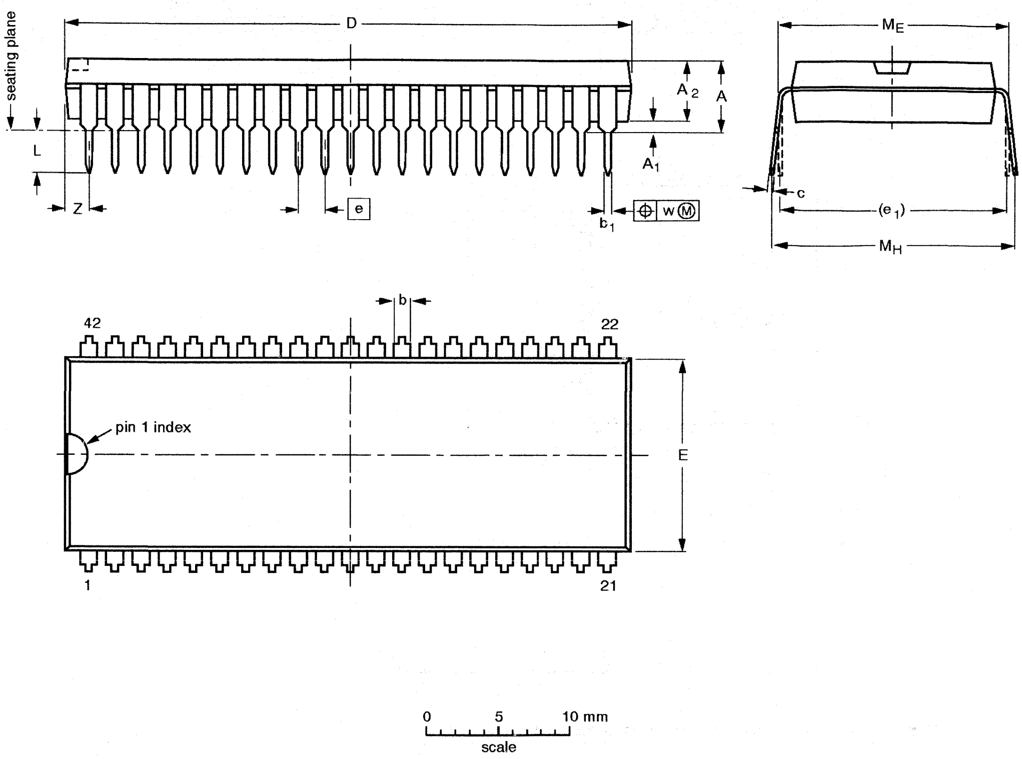
UNIT	A max.	A_1 min.	A_2 max.	b	b_1	c	D (1)	E (1)	e	e_1	L	M_E	M_H	w	Z (1) max.
mm	4.7	0.51	4.0	1.70 1.14	0.53 0.38	0.36 0.23	52.50 51.50	14.1 13.7	2.54	15.24	3.60 3.05	15.80 15.24	17.42 15.90	0.254	2.25
inches	0.19	0.020	0.16	0.067 0.045	0.021 0.015	0.014 0.009	2.067 2.028	0.56 0.54	0.10	0.60	0.14 0.12	0.62 0.60	0.69 0.63	0.01	0.089

Note

1. Plastic or metal protrusions of 0.25 mm maximum per side are not included.

OUTLINE VERSION	REFERENCES				EUROPEAN PROJECTION	ISSUE DATE
	IEC	JEDEC	EIAJ			
SOT129-1	051G08	MO-015AJ				~~92-11-17~~ 95-01-14

SDIP42: plastic shrink dual in-line package; 42 leads (600 mil) SOT270-1



DIMENSIONS (mm are the original dimensions)

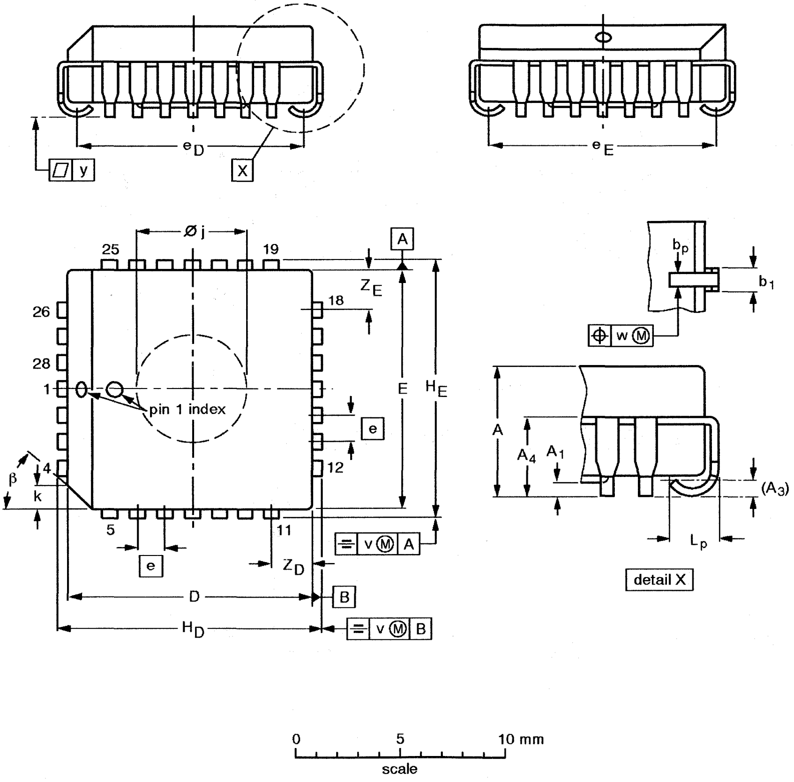
UNIT	A max.	A_1 min.	A_2 max.	b	b_1	c	D (1)	E (1)	e	e_1	L	M_E	M_H	w	Z (1) max.
mm	5.08	0.51	4.0	1.3 0.8	0.53 0.40	0.32 0.23	38.9 38.4	14.0 13.7	1.778	15.24	3.2 2.9	15.80 15.24	17.15 15.90	0.18	1.73

Note

1. Plastic or metal protrusions of 0.25 mm maximum per side are not included.

OUTLINE VERSION	REFERENCES				EUROPEAN PROJECTION	ISSUE DATE
	IEC	JEDEC	EIAJ			
SOT270-1						~~90-02-13~~ 95-02-04

PLCC28: plastic leaded chip carrer; 28 leads; pedestal SOT261-3



DIMENSIONS (millimetre dimensions are derived from the original inch dimensions)

UNIT	A	A_1 min.	A_3	A_4 max.	b_p	b_1	D[1]	E[1]	e	e_D	e_E	H_D	H_E	k	Øj	L_p	v	w	y	Z_D[1] max.	Z_E[1] max.	β
mm	4.57 4.19	0.13	0.25	3.05	0.53 0.33	0.81 0.66	11.58 11.43	11.58 11.43	1.27	10.92 9.91	10.92 9.91	12.57 12.32	12.57 12.32	1.22 1.07	5.69 5.54	1.44 1.02	0.18	0.18	0.10	2.06	2.06	45°
inches	0.180 0.165	0.005	0.01	0.12	0.021 0.013	0.032 0.026	0.456 0.450	0.456 0.450	0.05	0.430 0.390	0.430 0.390	0.495 0.485	0.495 0.485	0.048 0.042	0.224 0.218	0.057 0.040	0.007	0.007	0.004	0.081	0.081	

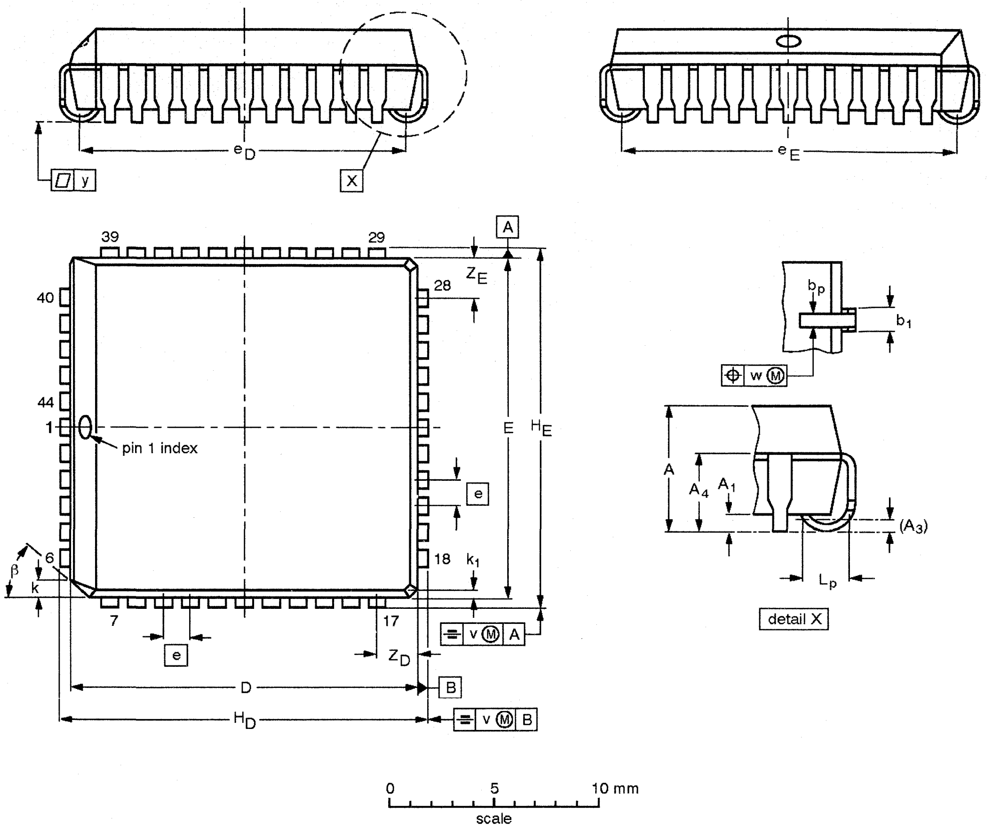
Note

1. Plastic or metal protrusions of 0.01 inches maximum per side are not included.

OUTLINE VERSION	REFERENCES				EUROPEAN PROJECTION	ISSUE DATE
	IEC	JEDEC	EIAJ			
SOT261-3		MO-047AB				~~92-11-17~~ 95-02-25

Package outlines

PLCC44: plastic leaded chip carrier; 44 leads SOT187-2



DIMENSIONS (millimetre dimensions are derived from the original inch dimensions)

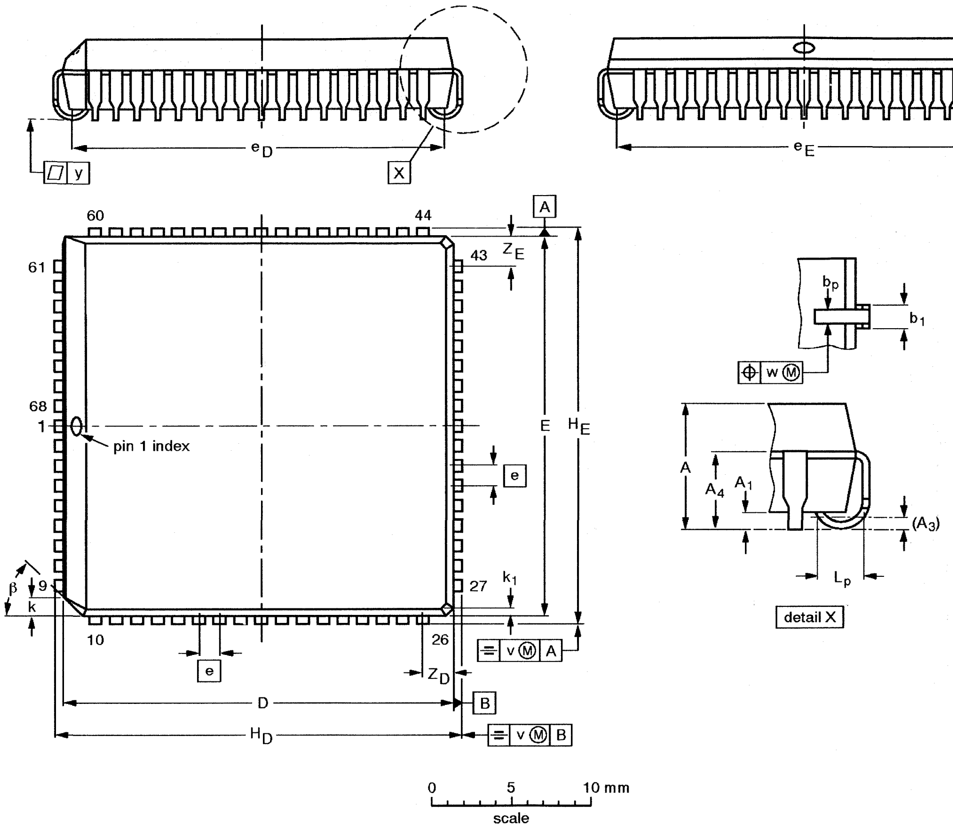
UNIT	A	A_1 min.	A_3	A_4 max.	b_p	b_1	$D^{(1)}$	$E^{(1)}$	e	e_D	e_E	H_D	H_E	k	k_1 max.	L_p	v	w	y	$Z_D^{(1)}$ max.	$Z_E^{(1)}$ max.	β
mm	4.57 4.19	0.51	0.25	3.05	0.53 0.33	0.81 0.66	16.66 16.51	16.66 16.51	1.27	16.00 14.99	16.00 14.99	17.65 17.40	17.65 17.40	1.22 1.07	0.51	1.44 1.02	0.18	0.18	0.10	2.16	2.16	45°
inches	0.180 0.165	0.020	0.01	0.12	0.021 0.013	0.032 0.026	0.656 0.650	0.656 0.650	0.05	0.630 0.590	0.630 0.590	0.695 0.685	0.695 0.685	0.048 0.042	0.020	0.057 0.040	0.007	0.007	0.004	0.085	0.085	

Note

1. Plastic or metal protrusions of 0.01 inches maximum per side are not included.

OUTLINE VERSION	REFERENCES				EUROPEAN PROJECTION	ISSUE DATE
	IEC	JEDEC	EIAJ			
SOT187-2	112E10	MO-047AC				~~92-11-17~~ 95-02-25

PLCC68: plastic leaded chip carrier; 68 leads SOT188-2



DIMENSIONS (millimetre dimensions are derived from the original inch dimensions)

UNIT	A	A_1 min.	A_3	A_4 max.	b_p	b_1	$D^{(1)}$	$E^{(1)}$	e	e_D	e_E	H_D	H_E	k	k_1 max.	L_p	v	w	y	$Z_D^{(1)}$ max.	$Z_E^{(1)}$ max.	β
mm	4.57 4.19	0.51	0.25	3.30	0.53 0.33	0.81 0.66	24.33 24.13	24.33 24.13	1.27	23.62 22.61	23.62 22.61	25.27 25.02	25.27 25.02	1.22 1.07	0.51	1.44 1.02	0.18	0.18	0.10	2.16	2.16	45°
inches	0.180 0.165	0.020	0.01	0.13	0.021 0.013	0.032 0.026	0.958 0.950	0.958 0.950	0.05	0.930 0.890	0.930 0.890	0.995 0.985	0.995 0.985	0.048 0.042	0.020	0.057 0.040	0.007	0.007	0.004	0.085	0.085	

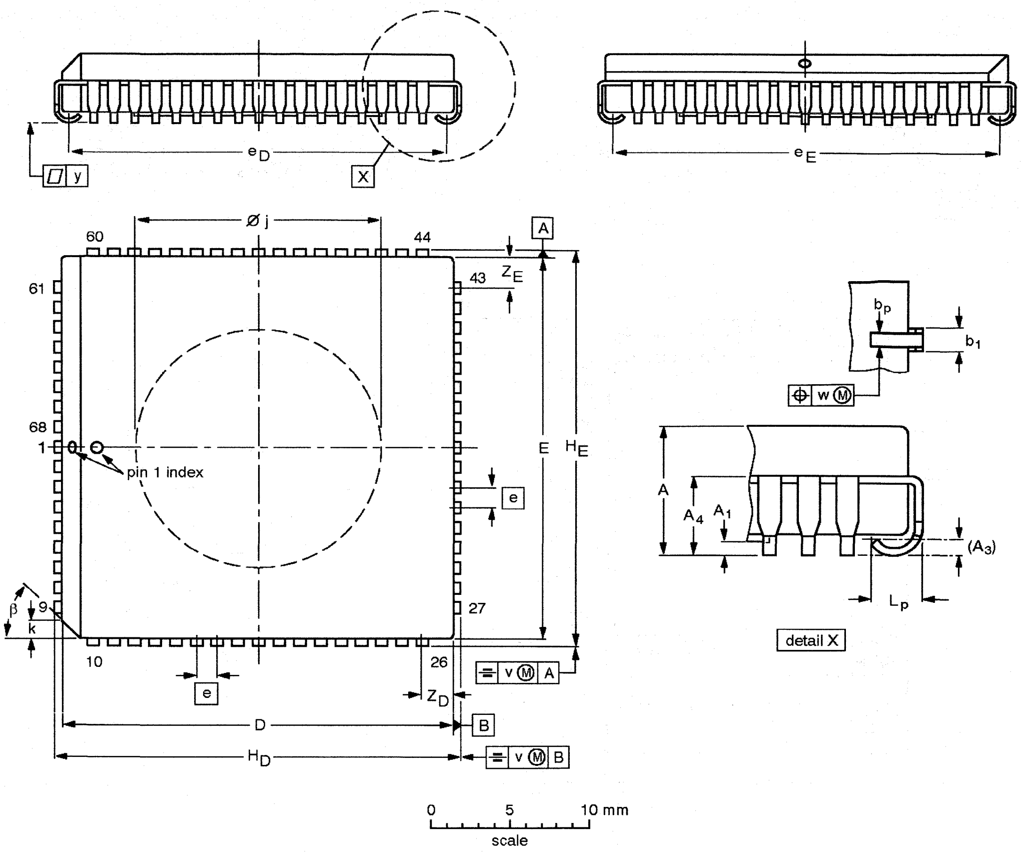
Note

1. Plastic or metal protrusions of 0.01 inches maximum per side are not included.

OUTLINE VERSION	REFERENCES				EUROPEAN PROJECTION	ISSUE DATE
	IEC	JEDEC	EIAJ			
SOT188-2	112E10	MO-047AC				~~92-11-17~~ 95-03-11

Package outlines

PLCC68: plastic leaded chip carrier; 68 leads; pedestal — SOT188-3



DIMENSIONS (millimetre dimensions are derived from the original inch dimensions)

UNIT	A	A_1 min.	A_3	A_4 max.	b_p	b_1	$D^{(1)}$	$E^{(1)}$	e	e_D	e_E	H_D	H_E	k	Øj	L_p	v	w	y	$Z_D^{(1)}$ max.	$Z_E^{(1)}$ max.	β
mm	4.57 4.19	0.13	0.25	3.05	0.53 0.33	0.81 0.66	24.33 24.13	24.33 24.13	1.27	23.62 22.61	23.62 22.61	25.27 25.02	25.27 25.02	1.22 1.07	15.34 15.19	1.44 1.02	0.18	0.18	0.10	2.06	2.06	45°
inches	0.180 0.165	0.005	0.01	0.12	0.021 0.013	0.032 0.026	0.958 0.950	0.958 0.950	0.05	0.930 0.890	0.930 0.890	0.995 0.985	0.995 0.985	0.048 0.042	0.604 0.598	0.057 0.040	0.007	0.007	0.004	0.081	0.081	

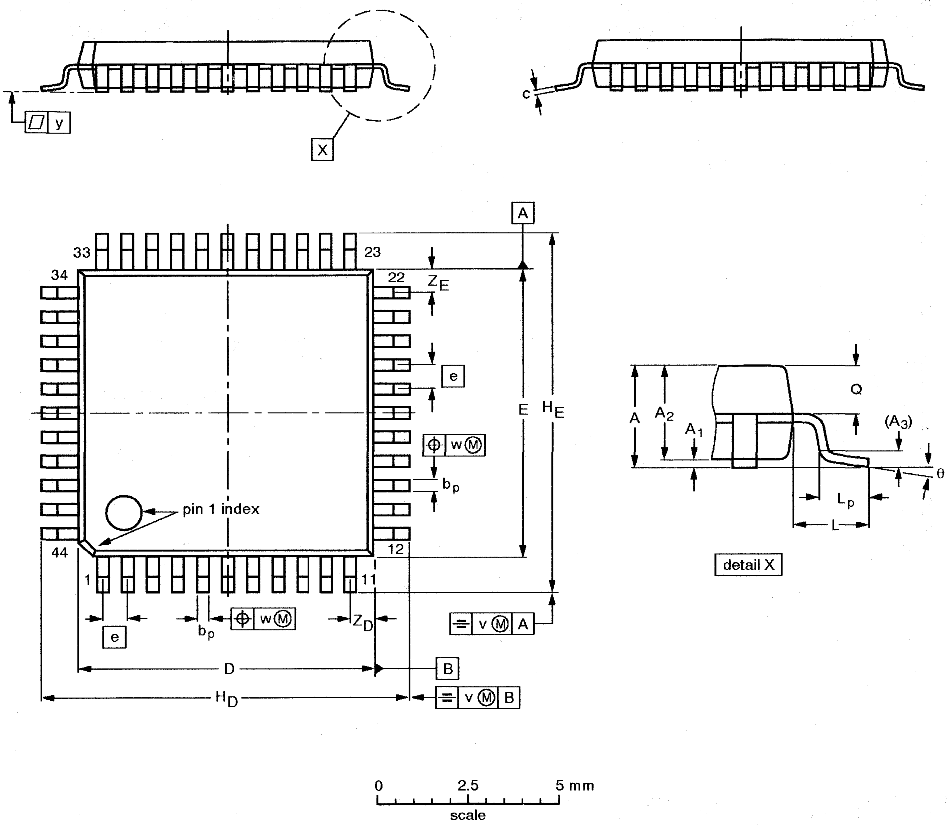
Note

1. Plastic or metal protrusions of 0.01 inches maximum per side are not included.

OUTLINE VERSION	REFERENCES				EUROPEAN PROJECTION	ISSUE DATE
	IEC	JEDEC	EIAJ			
SOT188-3	112E10	MO-047AE				~~92-11-17~~ 95-02-25

Package outlines

QFP44: plastic quad flat package; 44 leads (lead length 1.3 mm); body 10 x 10 x 1.75 mm SOT307-2



DIMENSIONS (mm are the original dimensions)

UNIT	A max.	A_1	A_2	A_3	b_p	c	$D^{(1)}$	$E^{(1)}$	e	H_D	H_E	L	L_p	Q	v	w	y	$Z_D^{(1)}$	$Z_E^{(1)}$	θ
mm	2.10	0.25 0.05	1.85 1.65	0.25	0.40 0.20	0.25 0.14	10.1 9.9	10.1 9.9	0.8	12.9 12.3	12.9 12.3	1.3	0.95 0.55	0.85 0.75	0.15	0.15	0.1	1.2 0.8	1.2 0.8	10° 0°

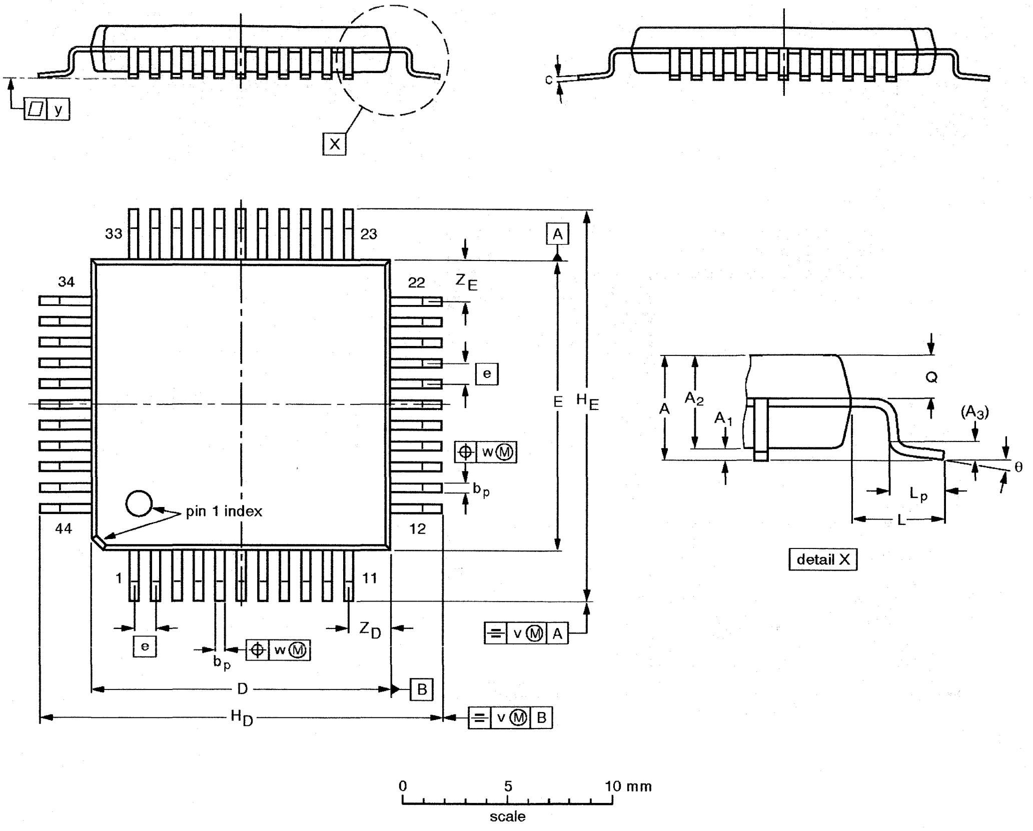
Note

1. Plastic or metal protrusions of 0.25 mm maximum per side are not included.

OUTLINE VERSION	REFERENCES				EUROPEAN PROJECTION	ISSUE DATE
	IEC	JEDEC	EIAJ			
SOT307-2						~~92-11-17~~ 95-02-04

Package outlines

QFP44: plastic quad flat package; 44 leads (lead length 2.35 mm); body 14 x 14 x 2.2 mm SOT205-1



DIMENSIONS (mm are the original dimensions)

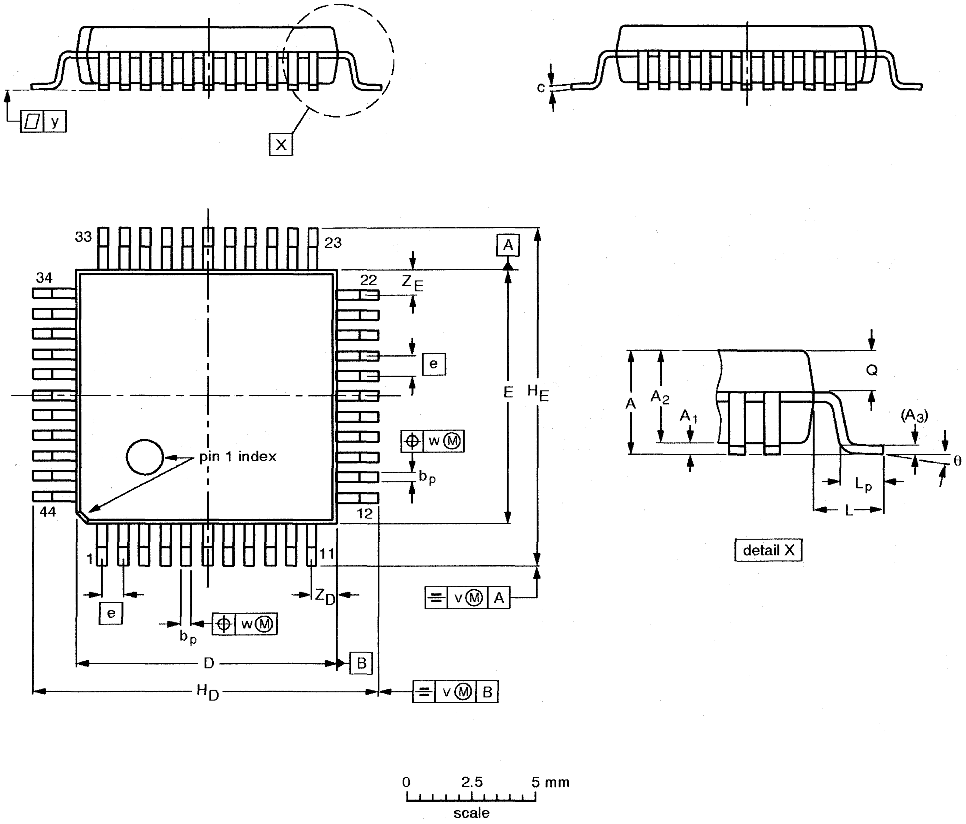
UNIT	A max.	A_1	A_2	A_3	b_p	c	$D^{(1)}$	$E^{(1)}$	e	H_D	H_E	L	L_p	Q	v	w	y	$Z_D^{(1)}$	$Z_E^{(1)}$	θ
mm	2.60	0.25 0.05	2.3 2.1	0.25	0.50 0.35	0.25 0.14	14.1 13.9	14.1 13.9	1	19.2 18.2	19.2 18.2	2.35	2.0 1.2	1.2 0.9	0.3	0.15	0.1	2.4 1.8	2.4 1.8	7° 0°

Note

1. Plastic or metal protrusions of 0.25 mm maximum per side are not included.

OUTLINE VERSION	REFERENCES				EUROPEAN PROJECTION	ISSUE DATE
	IEC	JEDEC	EIAJ			
SOT205-1	133E01A					~~92-11-17~~ 95-02-04

LQFP44: plastic low profile quad flat package; 44 leads; body 10 x 10 x 1.4 mm SOT389-1



DIMENSIONS (mm are the original dimensions)

UNIT	A max.	A_1	A_2	A_3	b_p	c	$D^{(1)}$	$E^{(1)}$	e	H_D	H_E	L	L_p	Q	v	w	y	$Z_D^{(1)}$	$Z_E^{(1)}$	θ
mm	1.60	0.15 0.05	1.45 1.35	0.25	0.45 0.30	0.20 0.12	10.10 9.90	10.10 9.90	0.80	12.15 11.85	12.15 11.85	1.0	0.75 0.45	0.70 0.57	0.20	0.20	0.10	1.14 0.85	1.14 0.85	7° 0°

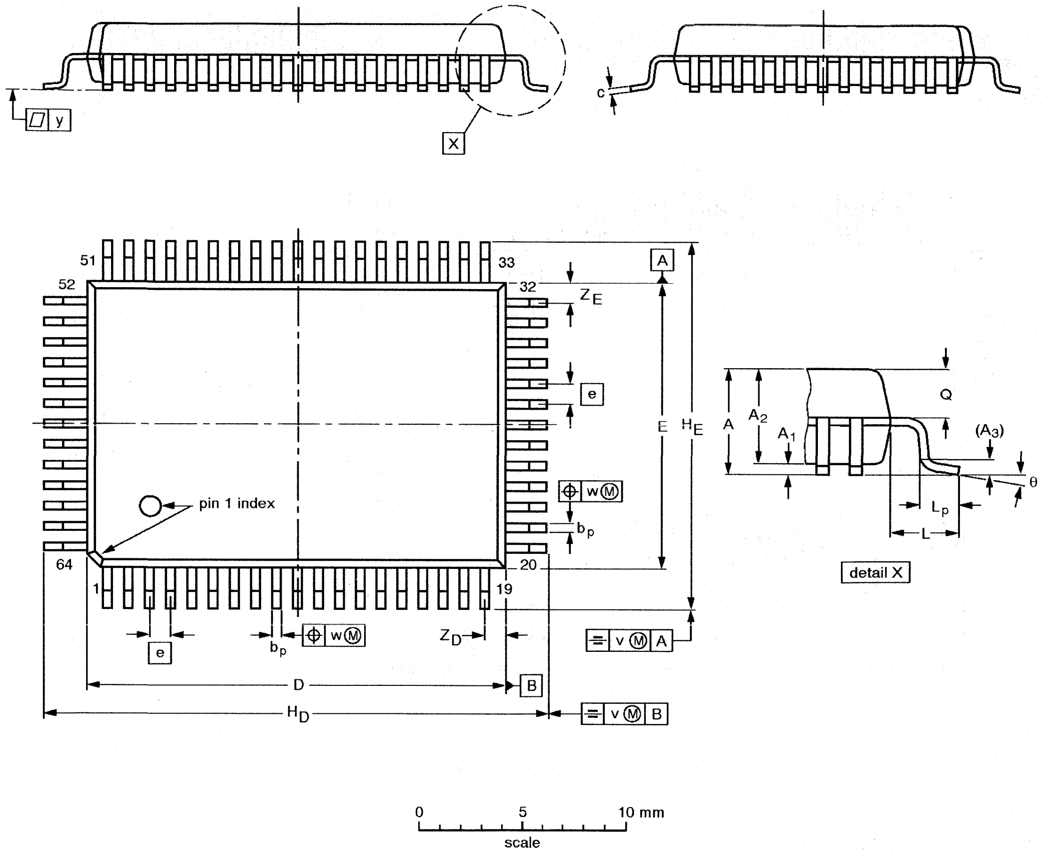
Note

1. Plastic or metal protrusions of 0.25 mm maximum per side are not included.

OUTLINE VERSION	REFERENCES				EUROPEAN PROJECTION	ISSUE DATE
	IEC	JEDEC	EIAJ			
SOT389-1						95-12-19

Package outlines

QFP64: plastic quad flat package; 64 leads (lead length 1.95mm); body 14 x 20 x 2.8 mm SOT319-2



DIMENSIONS (mm are the original dimensions)

UNIT	A max.	A_1	A_2	A_3	b_p	c	$D^{(1)}$	$E^{(1)}$	e	H_D	H_E	L	L_p	Q	v	w	y	$Z_D^{(1)}$	$Z_E^{(1)}$	θ
mm	3.20	0.25 0.05	2.90 2.65	0.25	0.50 0.35	0.25 0.14	20.1 19.9	14.1 13.9	1	24.2 23.6	18.2 17.6	1.95	1.0 0.6	1.4 1.2	0.2	0.2	0.1	1.2 0.8	1.2 0.8	7° 0°

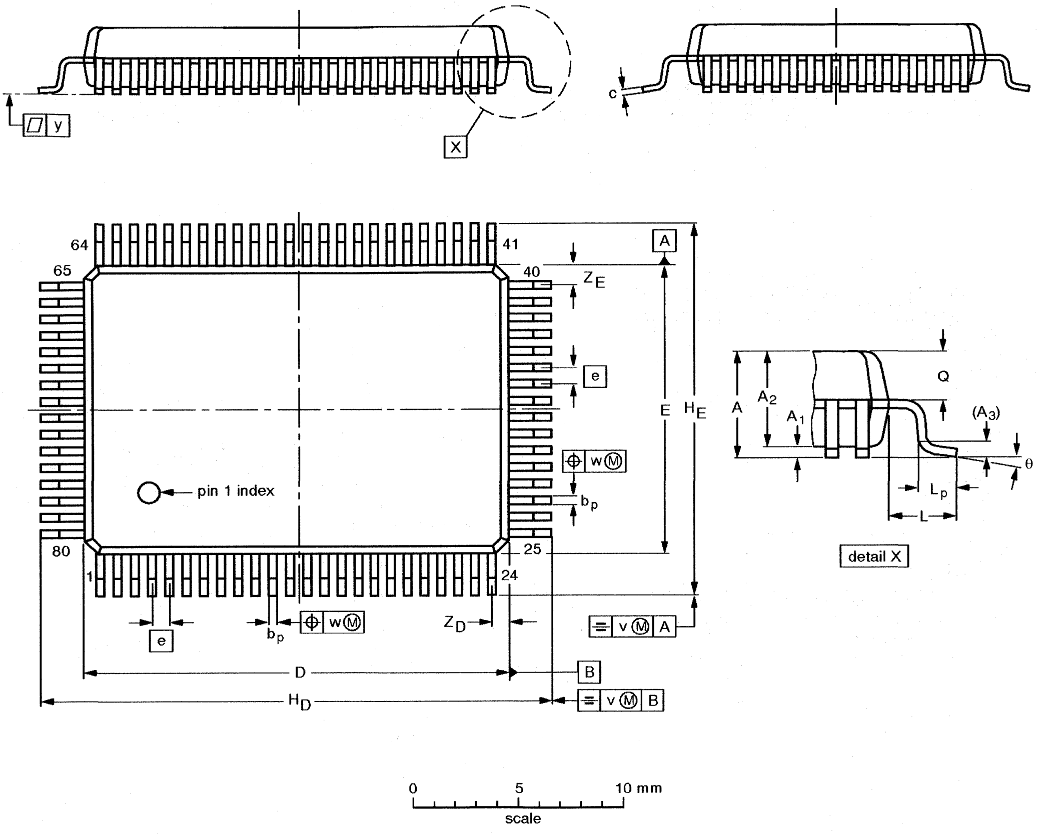
Note

1. Plastic or metal protrusions of 0.25 mm maximum per side are not included.

OUTLINE VERSION	REFERENCES				EUROPEAN PROJECTION	ISSUE DATE
	IEC	JEDEC	EIAJ			
SOT319-2						~~92-11-17~~ 95-02-04

Package outlines

QFP80: plastic quad flat package; 80 leads (lead length 1.95 mm); body 14 x 20 x 2.7 mm; high stand-off height SOT318-1



DIMENSIONS (mm are the original dimensions)

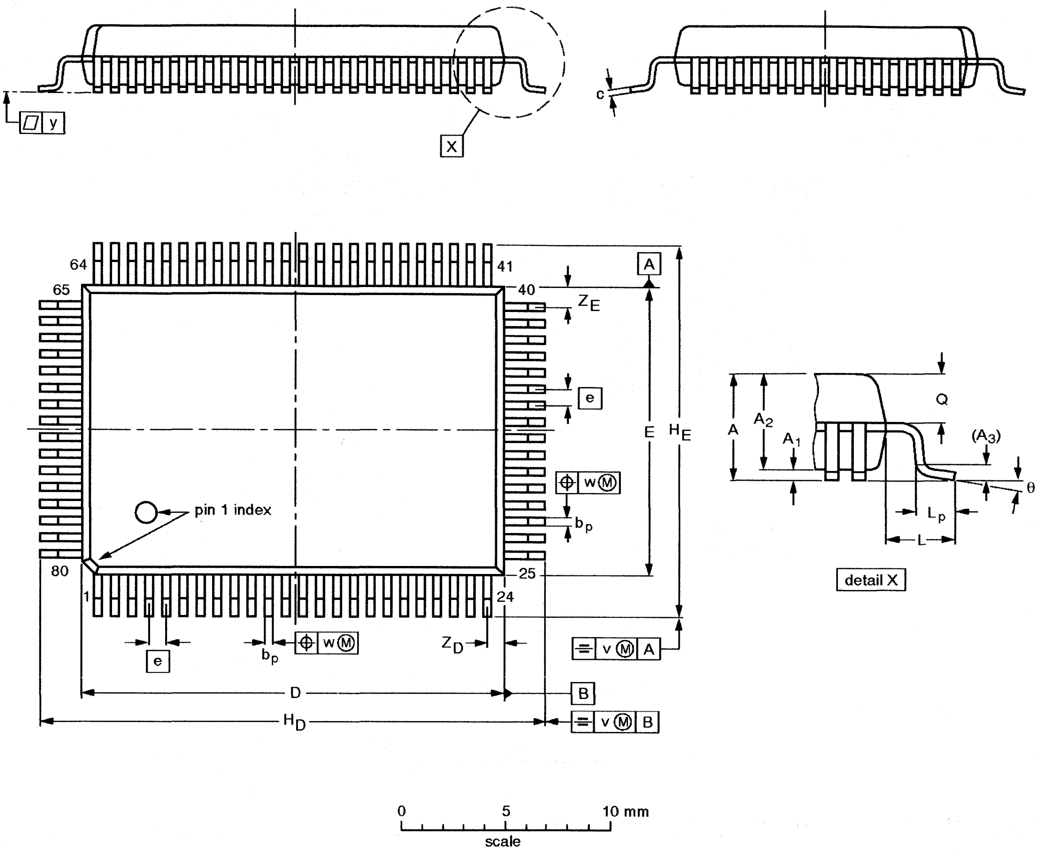
UNIT	A max.	A_1	A_2	A_3	b_p	c	$D^{(1)}$	$E^{(1)}$	e	H_D	H_E	L	L_p	Q	v	w	y	$Z_D{}^{(1)}$	$Z_E{}^{(1)}$	θ
mm	3.3	0.36 0.10	2.87 2.57	0.25	0.45 0.30	0.25 0.13	20.1 19.9	14.1 13.9	0.8	24.2 23.6	18.2 17.6	1.95	1.0 0.6	1.43 1.23	0.2	0.2	0.1	1.0 0.6	1.2 0.8	7° 0°

Note

1. Plastic or metal protrusions of 0.25 mm maximum per side are not included.

OUTLINE VERSION	REFERENCES				EUROPEAN PROJECTION	ISSUE DATE
	IEC	JEDEC	EIAJ			
SOT318-1						~~92-11-17~~ 95-02-04

QFP80: plastic quad flat package; 80 leads (lead length 1.95 mm); body 14 x 20 x 2.8 mm SOT318-2



DIMENSIONS (mm are the original dimensions)

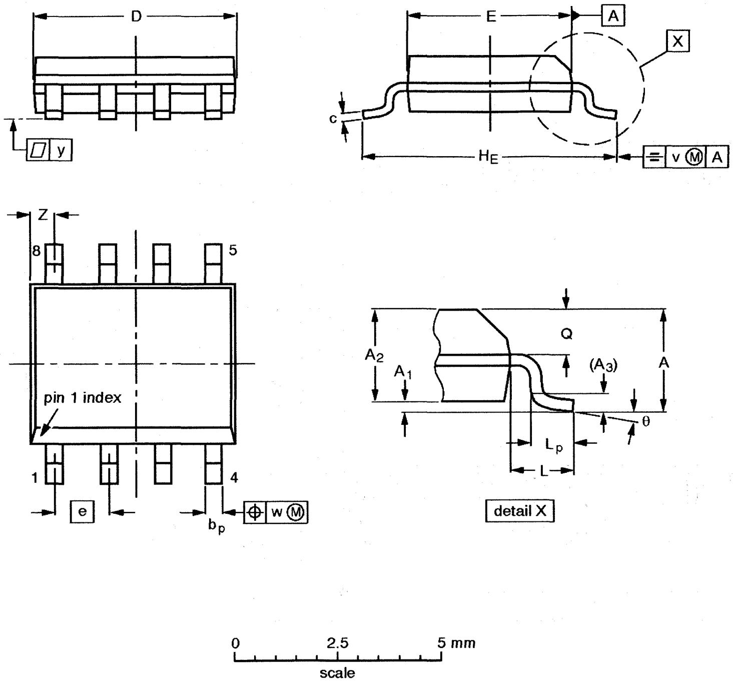
UNIT	A max.	A_1	A_2	A_3	b_p	c	D[1]	E[1]	e	H_D	H_E	L	L_p	Q	v	w	y	Z_D[1]	Z_E[1]	θ
mm	3.2	0.25 0.05	2.90 2.65	0.25	0.45 0.30	0.25 0.14	20.1 19.9	14.1 13.9	0.8	24.2 23.6	18.2 17.6	1.95	1.0 0.6	1.4 1.2	0.2	0.2	0.1	1.0 0.6	1.2 0.8	7° 0°

Note

1. Plastic or metal protrusions of 0.25 mm maximum per side are not included.

OUTLINE VERSION	REFERENCES				EUROPEAN PROJECTION	ISSUE DATE
	IEC	JEDEC	EIAJ			
SOT318-2						~~92-12-15~~ 95-02-04

SO8: plastic small outline package; 8 leads; body width 3.9mm SOT96-1



DIMENSIONS (inch dimensions are derived from the original mm dimensions)

UNIT	A max.	A_1	A_2	A_3	b_p	c	$D^{(1)}$	$E^{(2)}$	e	H_E	L	L_p	Q	v	w	y	$Z^{(1)}$	θ
mm	1.75	0.25 0.10	1.45 1.25	0.25	0.49 0.36	0.25 0.19	5.0 4.8	4.0 3.8	1.27	6.2 5.8	1.05	1.0 0.4	0.7 0.6	0.25	0.25	0.1	0.7 0.3	8° 0°
inches	0.069	0.0098 0.0039	0.057 0.049	0.01	0.019 0.014	0.0098 0.0075	0.20 0.19	0.16 0.15	0.050	0.24 0.23	0.041	0.039 0.016	0.028 0.024	0.01	0.01	0.004	0.028 0.012	

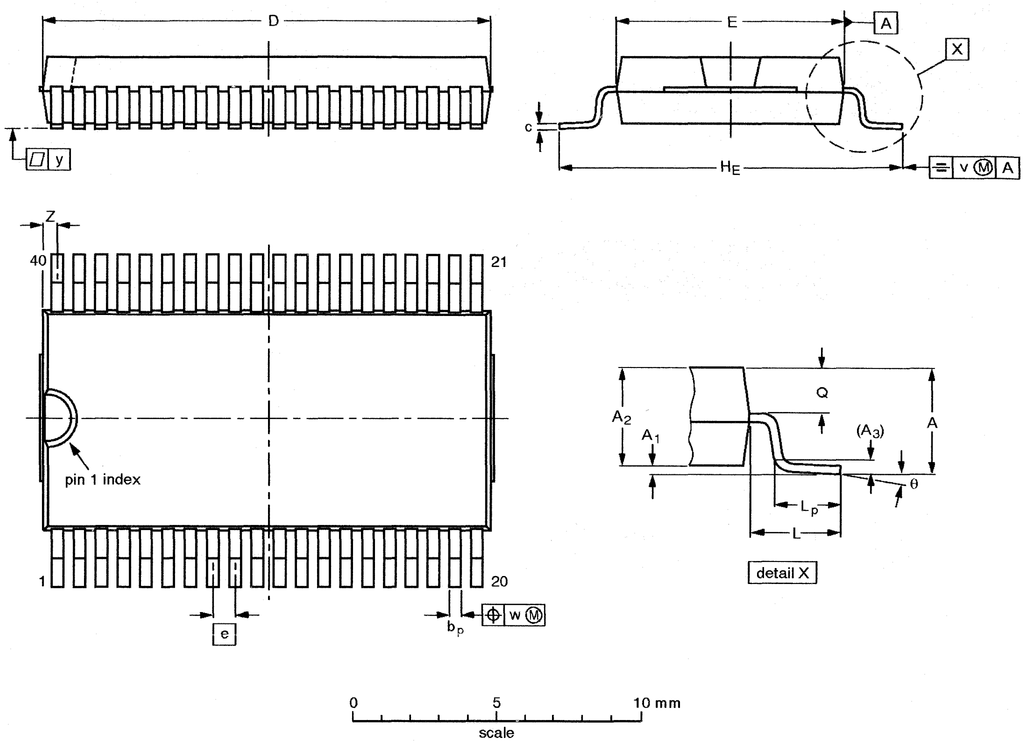
Notes

1. Plastic or metal protrusions of 0.15 mm maximum per side are not included.
2. Plastic or metal protrusions of 0.25 mm maximum per side are not included.

OUTLINE VERSION	REFERENCES				EUROPEAN PROJECTION	ISSUE DATE
	IEC	JEDEC	EIAJ			
SOT96-1	076E03S	MS-012AA				~~92-11-17~~ 95-02-04

Package outlines

VSO40: plastic very small outline package; 40 leads SOT158-1



DIMENSIONS (inch dimensions are derived from the original mm dimensions)

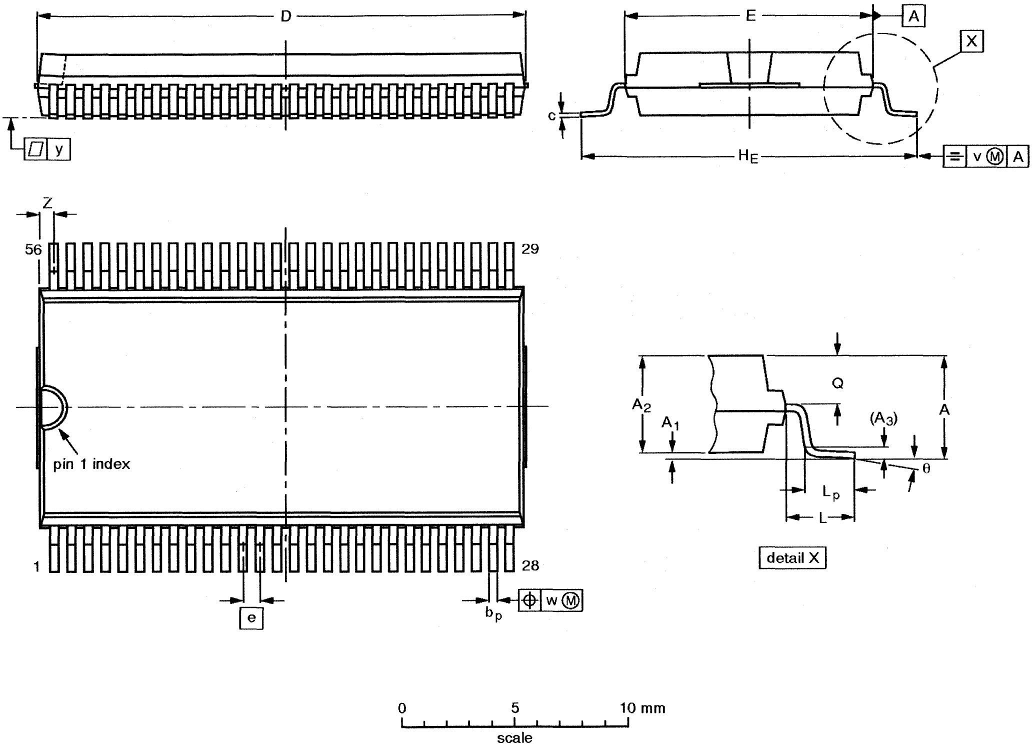
UNIT	A max.	A_1	A_2	A_3	b_p	c	$D^{(1)}$	$E^{(2)}$	e	H_E	L	L_p	Q	v	w	y	$Z^{(1)}$	θ
mm	2.70	0.3 0.1	2.45 2.25	0.25	0.42 0.30	0.22 0.14	15.6 15.2	7.6 7.5	0.762	12.3 11.8	2.25	1.7 1.5	1.15 1.05	0.2	0.1	0.1	0.6 0.3	7° 0°
inches	0.11	0.012 0.004	0.096 0.089	0.010	0.017 0.012	0.0087 0.0055	0.61 0.60	0.30 0.29	0.03	0.48 0.46	0.089	0.067 0.059	0.045 0.041	0.008	0.004	0.004	0.024 0.012	

Note

1. Plastic or metal protrusions of 0.4 mm maximum per side are not included.
2. Plastic interlead protrusions of 0.25 mm maximum per side are not included.

OUTLINE VERSION	REFERENCES				EUROPEAN PROJECTION	ISSUE DATE
	IEC	JEDEC	EIAJ			
SOT158-1						~~92-11-17~~ 95-01-24

VSO56: plastic very small outline package; 56 leads SOT190-1



DIMENSIONS (inch dimensions are derived from the original mm dimensions)

UNIT	A max.	A_1	A_2	A_3	b_p	c	$D^{(1)}$	$E^{(2)}$	e	H_E	L	L_p	Q	v	w	y	$Z^{(1)}$	θ
mm	3.3	0.3 0.1	3.0 2.8	0.25	0.42 0.30	0.22 0.14	21.65 21.35	11.1 11.0	0.75	15.8 15.2	2.25	1.6 1.4	1.45 1.30	0.2	0.1	0.1	0.90 0.55	7° 0°
inches	0.13	0.012 0.004	0.12 0.11	0.01	0.017 0.012	0.0087 0.0055	0.85 0.84	0.44 0.43	0.03	0.62 0.60	0.089	0.063 0.055	0.057 0.051	0.008	0.004	0.004	0.035 0.022	

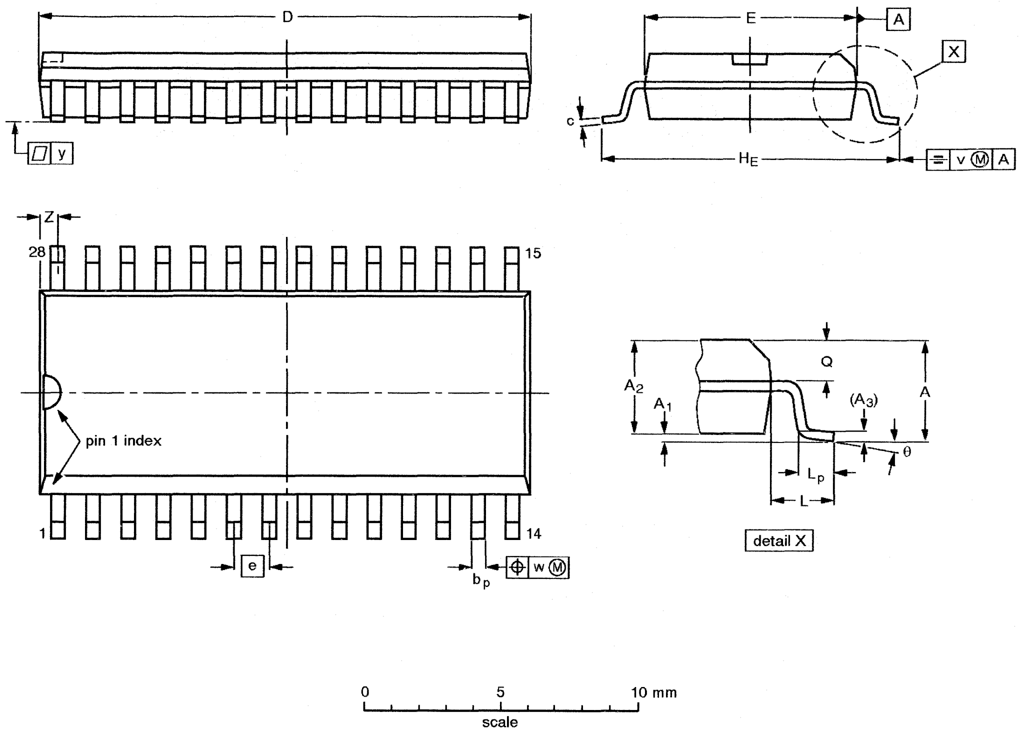
Note

1. Plastic or metal protrusions of 0.3 mm maximum per side are not included.
2. Plastic interlead protrusions of 0.25 mm maximum per side are not included.

OUTLINE VERSION	REFERENCES				EUROPEAN PROJECTION	ISSUE DATE
	IEC	JEDEC	EIAJ			
SOT190-1						~~92-11-17~~ 96-04-02

Package outlines

SO28: plastic small outline package; 28 leads; body width 7.5mm SOT136-1



DIMENSIONS (inch dimensions are derived from the original mm dimensions)

UNIT	A max.	A_1	A_2	A_3	b_p	c	D(1)	E(1)	e	H_E	L	L_p	Q	v	w	y	Z(1)	θ
mm	2.65	0.30 0.10	2.45 2.25	0.25	0.49 0.36	0.32 0.23	18.1 17.7	7.6 7.4	1.27	10.65 10.00	1.4	1.1 0.4	1.1 1.0	0.25	0.25	0.1	0.9 0.4	8° 0°
inches	0.10	0.012 0.004	0.096 0.089	0.01	0.019 0.014	0.013 0.009	0.71 0.69	0.30 0.29	0.050	0.42 0.39	0.055	0.043 0.016	0.043 0.039	0.01	0.01	0.004	0.035 0.016	

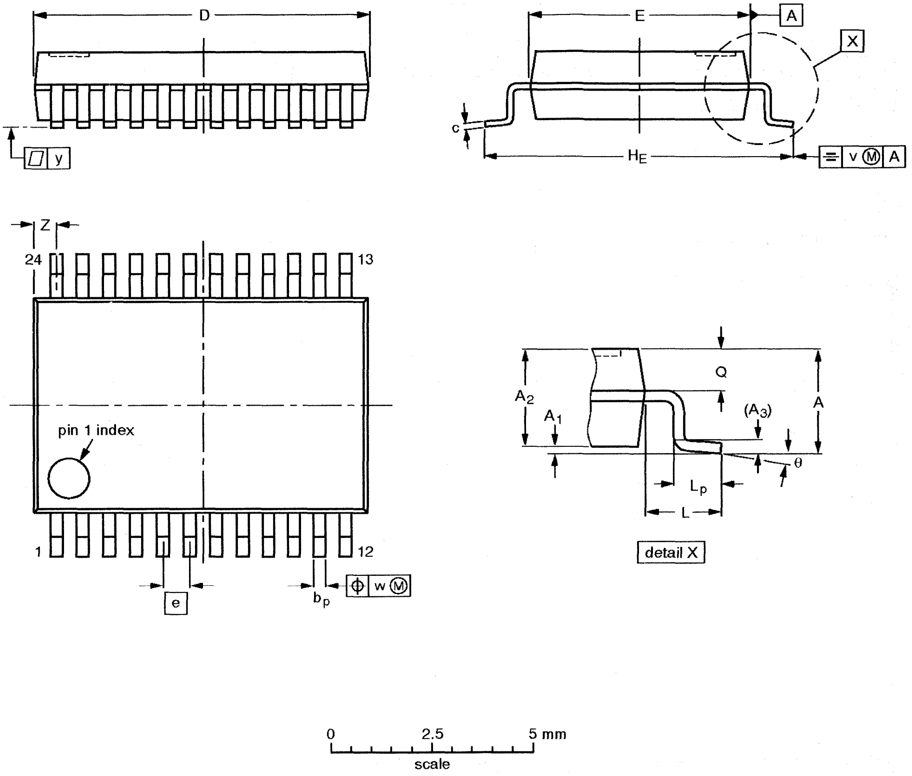
Note

1. Plastic or metal protrusions of 0.15 mm maximum per side are not included.

OUTLINE VERSION	REFERENCES				EUROPEAN PROJECTION	ISSUE DATE
	IEC	JEDEC	EIAJ			
SOT136-1	075E06	MS-013AE				~~91-08-13~~ 95-01-24

Package outlines

SSOP24: plastic shrink small outline package; 24 leads; body width 5.3 mm SOT340-1



DIMENSIONS (mm are the original dimensions)

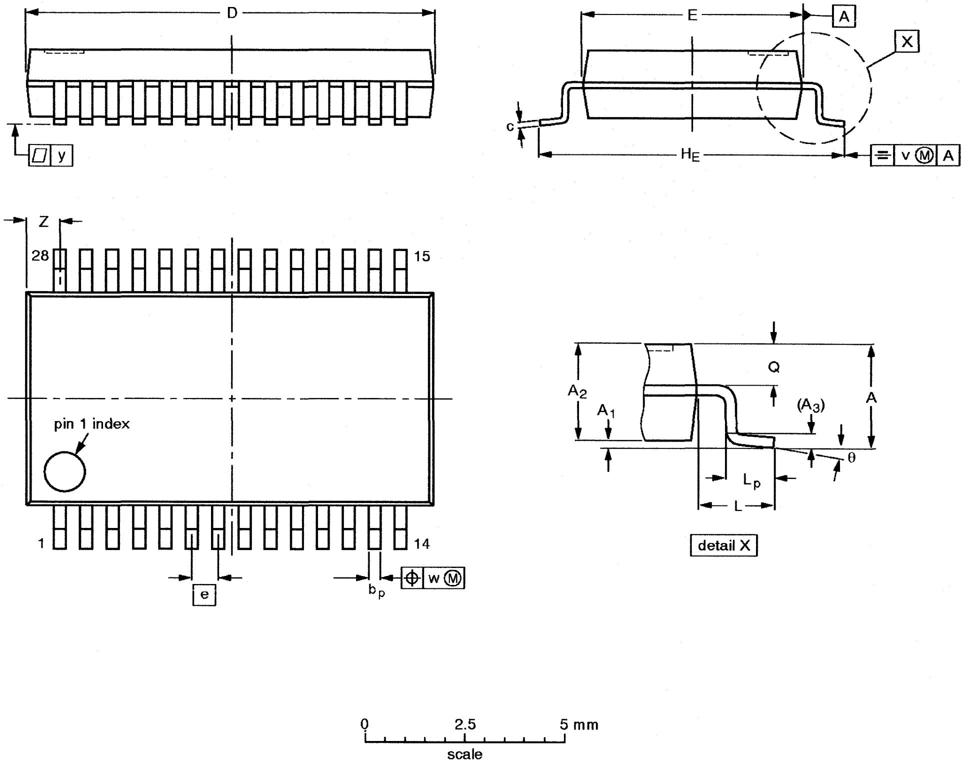
UNIT	A max.	A_1	A_2	A_3	b_p	c	$D^{(1)}$	$E^{(1)}$	e	H_E	L	L_p	Q	v	w	y	$Z^{(1)}$	θ
mm	2.0	0.21 0.05	1.80 1.65	0.25	0.38 0.25	0.20 0.09	8.4 8.0	5.4 5.2	0.65	7.9 7.6	1.25	1.03 0.63	0.9 0.7	0.2	0.13	0.1	0.8 0.4	8° 0°

Note

1. Plastic or metal protrusions of 0.20 mm maximum per side are not included.

OUTLINE VERSION	REFERENCES				EUROPEAN PROJECTION	ISSUE DATE
	IEC	JEDEC	EIAJ			
SOT340-1		MO-150AG				~~93-09-08~~ 95-02-04

SSOP28: plastic shrink small outline package; 28 leads; body width 5.3mm SOT341-1



DIMENSIONS (mm are the original dimensions)

UNIT	A max.	A_1	A_2	A_3	b_p	c	D(1)	E(1)	e	H_E	L	L_p	Q	v	w	y	Z(1)	θ
mm	2.0	0.21 0.05	1.80 1.65	0.25	0.38 0.25	0.20 0.09	10.4 10.0	5.4 5.2	0.65	7.9 7.6	1.25	1.03 0.63	0.9 0.7	0.2	0.13	0.1	1.1 0.7	8° 0°

Note

1. Plastic or metal protrusions of 0.20 mm maximum per side are not included.

OUTLINE VERSION	REFERENCES				EUROPEAN PROJECTION	ISSUE DATE
	IEC	JEDEC	EIAJ			
SOT341-1		MO-150AH				~~93-09-08~~ 95-02-04

Section 13
Data Handbook System

CONTENTS

Data handbook system 335

Data handbook system

DATA HANDBOOK SYSTEM

Philips Semiconductors data handbooks contain all pertinent data available at the time of publication and each is revised and reissued regularly.

Loose data sheets are sent to subscribers to keep them up-to-date on additions or alterations made during the lifetime of a data handbook.

Catalogs are available for selected product ranges (some catalogs are also on floppy discs).

Our data handbook titles are listed here.

Integrated Circuits

Book	*Title*
IC01	Semiconductors for Radio, Audio and CD/DVD Systems
IC02	Semiconductors for Television and Video Systems
IC03	Semiconductors for Wired Telecom Systems
IC04	HE4000B Logic Family CMOS
IC05	Advanced Low-power Schottky (ALS) Logic
IC06	High-speed CMOS Logic Family
IC11	General-purpose/Linear ICs
IC12	I^2C Peripherals
IC13	Programmable Logic Devices (PLD)
IC14	8048-based 8-bit Microcontrollers
IC15	FAST TTL Logic Series
IC16	CMOS ICs for Clocks and Watches
IC17	Semiconductors for Wireless Communications
IC18	Semiconductors for In-Car Electronics
IC19	ICs for Data Communications
IC20	80C51-based 8-bit Microcontrollers
IC22	Multimedia ICs
IC23	BiCMOS Bus Interface Logic
IC24	Low Voltage CMOS & BiCMOS Logic
IC25	16-bit 80C51XA Microcontrollers (eXtended Architecture)
IC26	Integrated Circuit Packages
IC27	Complex Programmable Logic Devices

Discrete Semiconductors

Book	*Title*
SC01	Small-signal and Medium-power Diodes
SC02	Power Diodes
SC03	Power Thyristors and Triacs
SC04	Small-signal Transistors
SC05	Video Transistors and Modules for Monitors
SC06	High-voltage and Switching NPN Power Transistors
SC07	Small-signal Field-effect Transistors
SC13a	Power MOS Transistors including TOPFETs and IGBTs
SC13b	Small-signal and Medium-power MOS Transistors
SC14	RF Wideband Transistors
SC16	Wideband Hybrid IC Modules
SC17	Semiconductor Sensors
SC18	Discrete Semiconductor Packages
SC19	RF & Microwave Power Transistors, RF Power Modules and Circulators/Isolators

MORE INFORMATION FROM PHILIPS SEMICONDUCTORS?
For more information about Philips Semiconductors data handbooks, catalogs and subscriptions, contact your nearest Philips Semiconductors national organization, select from the **address list on the back cover of this handbook**. Product specialists are at your service and inquiries are answered promptly.

OVERVIEW OF PHILIPS COMPONENTS DATA HANDBOOKS

Our sister product division, Philips Components, also has a comprehensive data handbook system to support their products. Their data handbook titles are listed here.

Display Components

Book	*Title*
DC01	Colour Television Tubes
DC02	Monochrome Monitor Tubes and Deflection Units
DC03	Television Tuners, Coaxial Aerial Input Assemblies
DC04	Colour Monitor and Multimedia Tubes
DC05	Wire Wound Components

Magnetic Products

MA01	Soft Ferrites
MA03	Piezoelectric Ceramics Specialty Ferrites
MA04	Dry-reed Switches

Passive Components

PA01	Electrolytic Capacitors
PA02	Varistors, Thermistors and Sensors
PA03	Potentiometers
PA04	Variable Capacitors
PA05	Film Capacitors
PA06	Ceramic Capacitors
PA06a	Surface Mounted Ceramic Multilayer Capacitors
PA06b	Leaded Ceramic Capacitors
PA08	Fixed Resistors
PA10	Quartz Crystals
PA11	Quartz Oscillators

MORE INFORMATION FROM PHILIPS COMPONENTS?

For more information contact your nearest Philips Components national organizaiton shown in the following list.

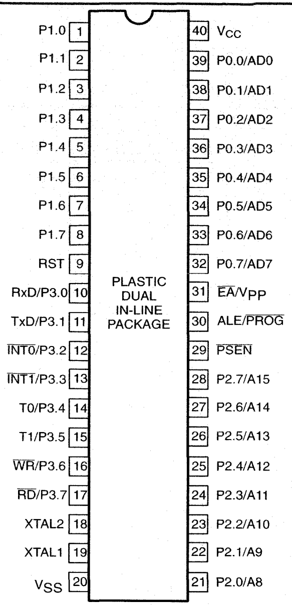
Australia: NORTH RYDE, Tel. (02) 9805 4455, Fax. (02) 9805 4466.
Austria: WIEN, Tel. (01) 601 01 12 41, Fax. (01) 60 101 12 11.
Belarus: MINSK, Tel. (5172) 200 924/733, Fax. (5172) 200 773.
Benelux: EINDHOVEN, Tel. (+31 40) 2783 749, Fax. (+31 40) 2788 399.
Brazil: SÃO PAULO, Tel. (011) 821 2333, Fax (011) 829 1849.
Canada: SCARBOROUGH, Tel. (0416) 292 5161, Fax. (0416) 754 6248.
China: SHANGHAI, Tel. (021) 6354 1088, Fax. (021) 6354 1060.
Denmark: COPENHAGEN, Tel. (32) 883 333, Fax. (31) 571 949.
Finland: ESPOO, Tel. 9 (0)-615 800, Fax. 9 (0)-615 80510.
France: SURESNES, Tel. (01) 4099 6161, Fax, (01) 4099 6493.
Germany: HAMBURG, Tel. (040) 2489-0, Fax. (040) 2489 1400.
Greece: TAVROS, Tel. (01) 4894 339/(01) 4894 239, Fax. (01) 4814 240.
Hong Kong: KOWLOON, Tel. 2784 3000, Fax. 2784 3003.
India: MUMBAI, Tel. (022) 4930 311, Fax. (022) 4930 966/4950 304.
Indonesia: JAKARTA, Tel. (021) 794 0040, Fax. (021) 794 0080.
Ireland: DUBLIN, Tel. (01) 76 40 203, Fax. (01) 76 40 210.
Israel: TEL AVIV, Tel (03) 6450 444, Fax. (03) 6491 007.
Italy: MILANO, Tel. (02) 6752 2531, Fax. (02) 6752 2557.
Japan: TOKYO, Tel. (0) 3 3740 5135, Fax. (0) 3 3740 5035.
Korea (Republic of)**:** SEOUL, Tel. (02) 709 1472, Fax. (02) 709 1480.
Malaysia: PULAU PINANG, Tel. (03) 750 5213, Fax. (03) 757 4880.
Mexico: EL PASO, Tel. (915) 772 4020, Fax. (915) 772 4332.
New Zealand: AUKLAND, Tel. (09) 815 4000, Fax. (09) 849 7811.
Norway: OSLO, Tel. (22) 74 8000, Fax (22) 74 8341.
Pakistan: KARACHI, Tel. (021) 587 4641-49, Fax. (021) 577 035/(021) 587 4546.
Philippines: MANILA, Tel. (02) 816 6345, Fax. (02) 817 3474.
Poland: WARSZAWA, Tel. (022) 612 2594, Fax. (022) 612 2327.
Portugal: LINDA-A-VELHA, Tel. (01) 416 3160/416 3333,
Fax. (01) 416 3174/416 3366.
Russia: MOSCOW, Tel (095) 755 6918, Fax. (095) 755 6919.
Singapore: SINGAPORE, Tel. 350 2000, Fax. 355 1758.
South Africa: JOHANNESBURG, Tel. (011) 470 5911, Fax. (011) 470 5494.
Spain: BARCELONA, Tel. (93) 301 63 12, Fax. (93) 301 42 43.
Sweden: STOCKHOLM, Tel. (+46) 8 632 2000, Fax. (+46) 8 632 2745.
Switzerland: ZÜRICH, Tel. (01) 488 22 11, Fax. (01) 481 77 30.
Taiwan: TAIPEI, Tel. (02) 2134 2900, Fax. (02) 2134 2929.
Thailand: BANGKOK, Tel. (02) 745 4090, Fax. (02) 398 0793.
Turkey: ISTANBUL, Tel. (0212) 279 2770, Fax. (0212) 282 6707.
United Kingdom: DORKING, Tel. (01306) 512 000, Fax. (01306) 512 345.
United States:

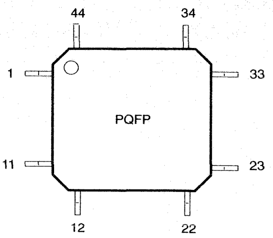
- ANN ARBOR, MI, Tel. (313) 996 9400, Fax. (313) 761 2776.
- SAUGERTIES, NY, Tel. (914) 246 2811, Fax (914) 246 0487
- SAN JOSE, CA.. Tel. (408) 570 5600, Fax, (408) 570 5700.

Yugoslavia (Federal Republic of)**:** BELGRADE, Tel. (0) 11 625 344/373,
Fax (0) 11 635 777.

For all other countries apply to:
Philips Components.
Marketing Communications,
P.O. Box 218,
5600 MD, EINDHOVEN, The Netherlands
Fax. +31–40–2724547.

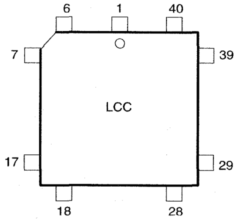
8XC51/80C31





Pin	Function	Pin	Function	Pin	Function
1	P1.5	16	V_{SS}	31	P0.6/AD6
2	P1.6	17	NIC*	32	P0.5/AD5
3	P1.7	18	P2.0/A8	33	P0.4/AD4
4	RST	19	P2.1/A9	34	P0.3/AD3
5	P3.0/RxD	20	P2.2/A10	35	P0.2/AD2
6	NIC*	21	P2.3/A11	36	P0.1/AD1
7	P3.1/TxD	22	P2.4/A12	37	P0.0/AD0
8	P3.2/$\overline{\text{INT0}}$	23	P2.5/A13	38	V_{CC}
9	P3.3/$\overline{\text{INT1}}$	24	P2.6/A14	39	NIC*
10	P3.4/T0	25	P2.7/A15	40	P1.0
11	P3.5/T1	26	$\overline{\text{PSEN}}$	41	P1.1
12	P3.6/$\overline{\text{WR}}$	27	ALE/$\overline{\text{PROG}}$	42	P1.2
13	P3.7/$\overline{\text{RD}}$	28	NIC*	43	P.13
14	XTAL2	29	$\overline{\text{EA}}$/V_{PP}	44	P1.4
15	XTAL1	30	P0.7/AD7		

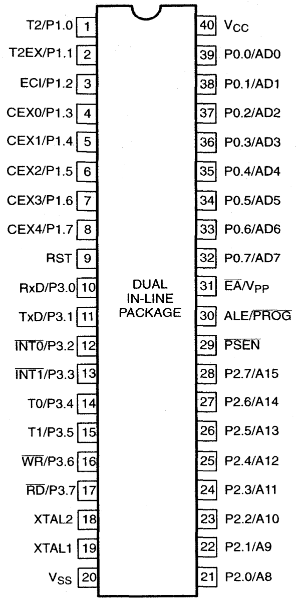
* NO INTERNAL CONNECTION

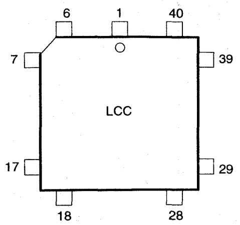


Pin	Function	Pin	Function	Pin	Function
1	NIC*	16	P3.4/T0	31	P2.7/A15
2	P1.0	17	P3.5/T1	32	$\overline{\text{PSEN}}$
3	P1.1	18	P3.6/$\overline{\text{WR}}$	33	ALE/$\overline{\text{PROG}}$
4	P1.2	19	P3.7/$\overline{\text{RD}}$	34	NIC*
5	P1.3	20	XTAL2	35	$\overline{\text{EA}}$/V_{PP}
6	P1.4	21	XTAL1	36	P0.7/AD7
7	P1.5	22	V_{SS}	37	P0.6/AD6
8	P1.6	23	NIC*	38	P0.5/AD5
9	P1.7	24	P2.0/A8	39	P0.4/AD4
10	RST	25	P2.1/A9	40	P0.3/AD3
11	P3.0/RxD	26	P2.2/A10	41	P0.2/AD2
12	NIC*	27	P2.3/A11	42	P0.1/AD1
13	P3.1/TxD	28	P2.4/A12	43	P0.0/AD0
14	P3.2/$\overline{\text{INT0}}$	29	P2.5/A13	44	V_{CC}
15	P3.3/$\overline{\text{INT1}}$	30	P2.6/A14		

* NO INTERNAL CONNECTION

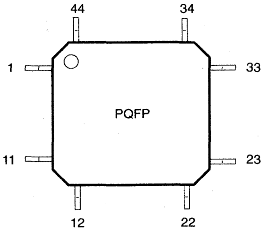
8XC52/54/58/80C32
8XC51FA/FB/FC/80C51FA
8XC51RA+/RB+/RC+/RD+/80C51RA+,
89C52/54/58/89C51RA+/RB+/RC+/RD+





Pin	Function	Pin	Function	Pin	Function
1	NIC*	16	P3.4/T0	31	P2.7/A15
2	P1.0/T2	17	P3.5/T1	32	$\overline{\text{PSEN}}$
3	P1.1/T2EX	18	P3.6/$\overline{\text{WR}}$	33	ALE/$\overline{\text{PROG}}$
4	P1.2/ECI	19	P3.7/$\overline{\text{RD}}$	34	NIC*
5	P1.3/CEX0	20	XTAL2	35	$\overline{\text{EA}}$/V_{PP}
6	P1.4/CEX1	21	XTAL1	36	P0.7/AD7
7	P1.5/CEX2	22	V_{SS}	37	P0.6/AD6
8	P1.6/CEX3	23	NIC*	38	P0.5/AD5
9	P1.7/CEX4	24	P2.0/A8	39	P0.4/AD4
10	RST	25	P2.1/A9	40	P0.3/AD3
11	P3.0/RxD	26	P2.2/A10	41	P0.2/AD2
12	NIC*	27	P2.3/A11	42	P0.1/AD1
13	P3.1/TxD	28	P2.4/A12	43	P0.0/AD0
14	P3.2/$\overline{\text{INT0}}$	29	P2.5/A13	44	V_{CC}
15	P3.3/$\overline{\text{INT1}}$	30	P2.6/A14		

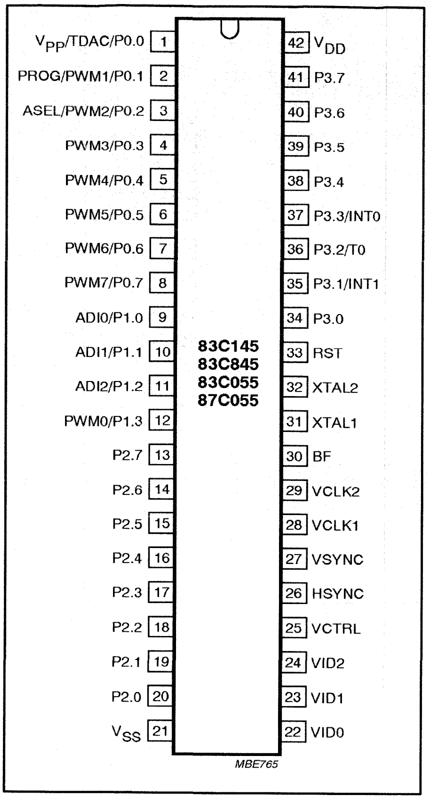
* NO INTERNAL CONNECTION



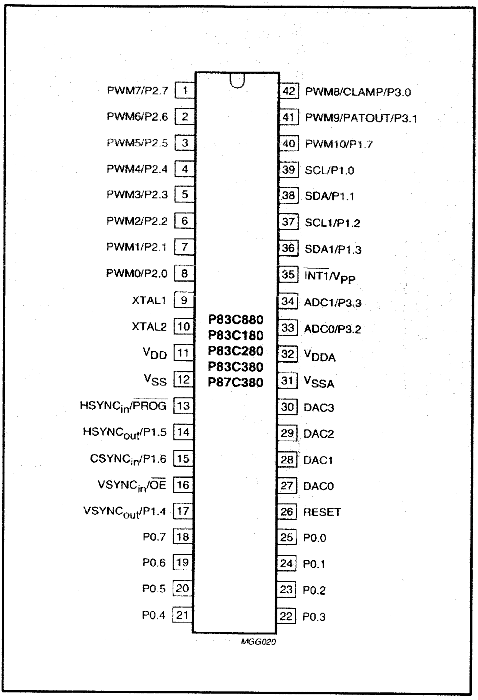
Pin	Function	Pin	Function	Pin	Function
1	P1.5/CEX2	16	V_{SS}	31	P0.6/AD6
2	P1.6/CEX3	17	NIC*	32	P0.5/AD5
3	P1.7/CEX4	18	P2.0/A8	33	P0.4/AD4
4	RST	19	P2.1/A9	34	P0.3/AD3
5	P3.0/RxD	20	P2.2/A10	35	P0.2/AD2
6	NIC*	21	P2.3/A11	36	P0.1/AD1
7	P3.1/TxD	22	P2.4/A12	37	P0.0/AD0
8	P3.2/$\overline{\text{INT0}}$	23	P2.5/A13	38	V_{CC}
9	P3.3/$\overline{\text{INT1}}$	24	P2.6/A14	39	NIC*
10	P3.4/T0	25	P2.7/A15	40	P1.0/T2
11	P3.5/T1	26	$\overline{\text{PSEN}}$	41	P1.1/T2EX
12	P3.6/$\overline{\text{WR}}$	27	ALE/$\overline{\text{PROG}}$	42	P1.2/ECI
13	P3.7/$\overline{\text{RD}}$	28	NIC*	43	P1.3/CEX0
14	XTAL2	29	$\overline{\text{EA}}$/V_{PP}	44	P1.4/CEX1
15	XTAL1	30	P0.7/AD7		

* NO INTERNAL CONNECTION

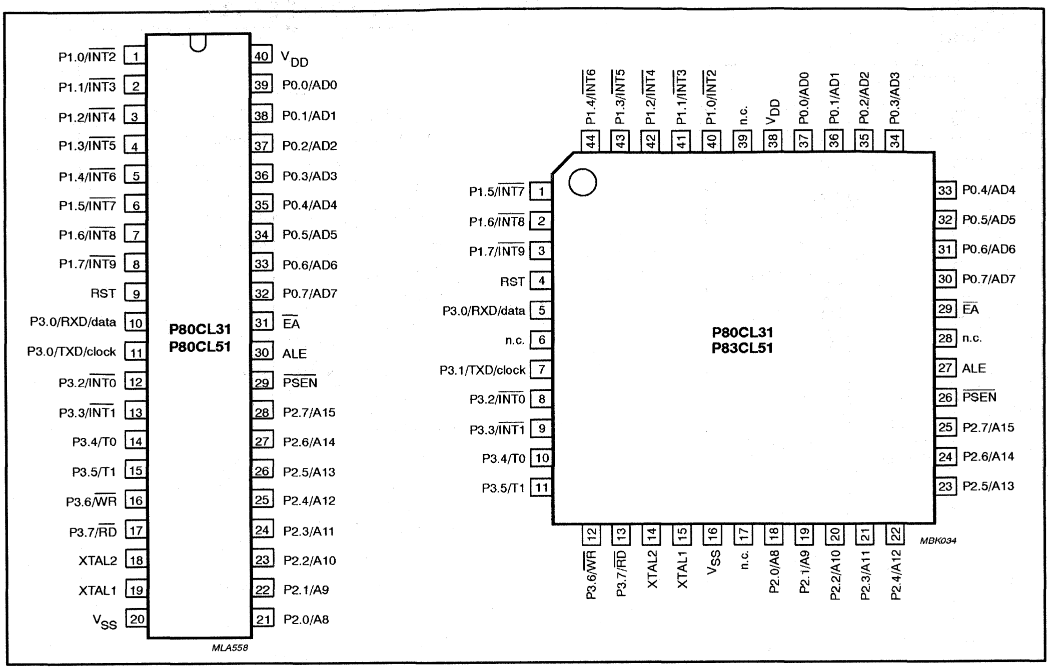
83C145; 83C845; 83C055; 87C055



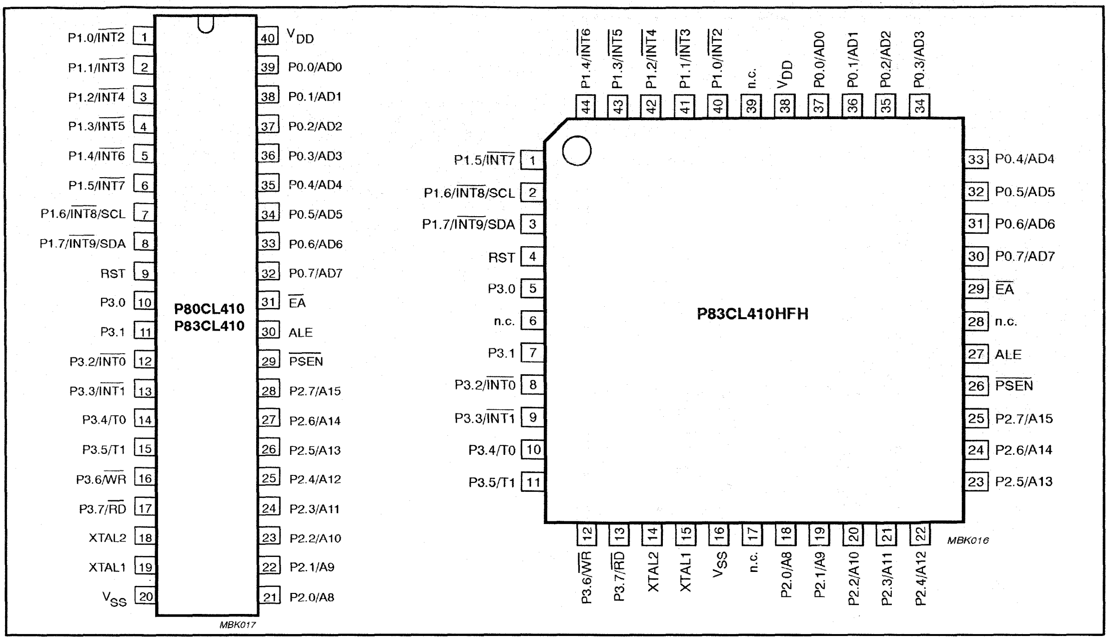
P83C880/180/280/380/P87C380



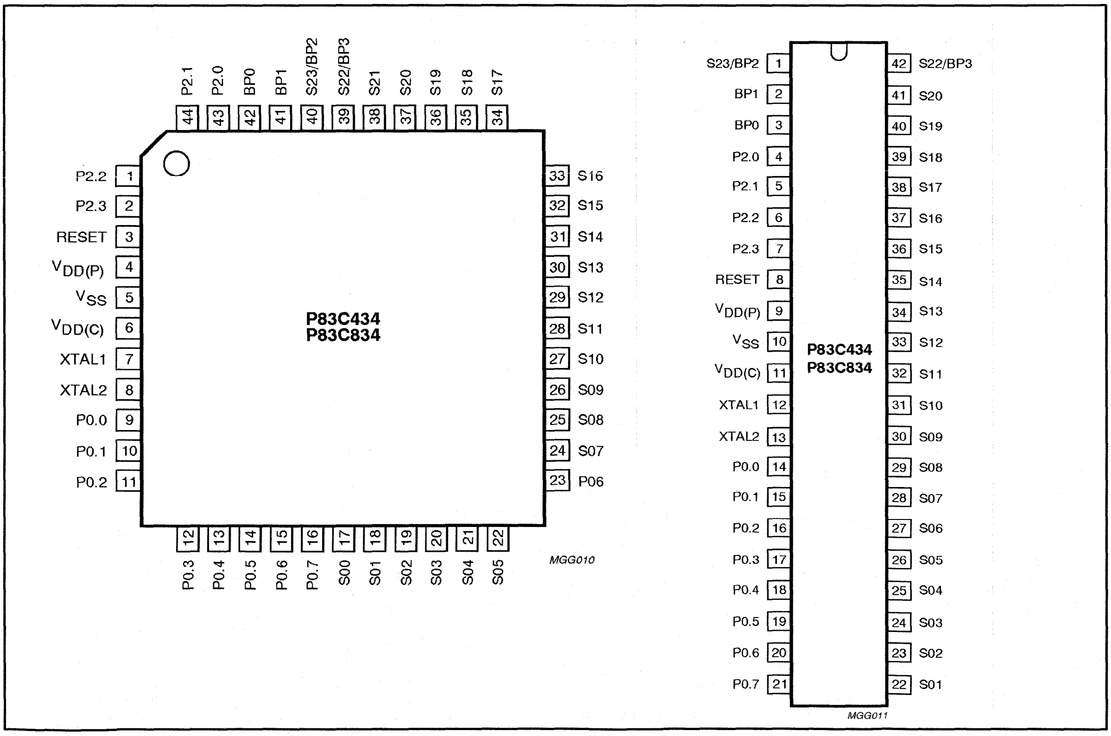
P80CL31; P80CL51



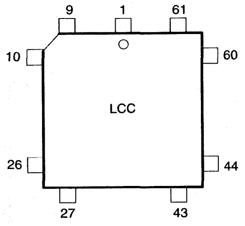
P80CL410; P83CL410



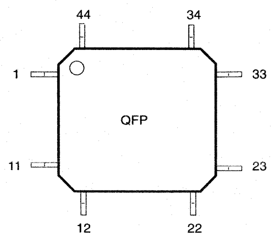
P83C434; P83C834



80C451/83C451/87C451

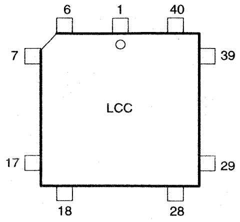


Pin	Function	Pin	Function	Pin	Function
1	$\overline{EA}/V_{PP}$	24	P4.2	47	P5.3
2	P2.0/A8	25	P4.1	48	P5.4
3	P2.1/A9	26	P4.0	49	P5.5
4	P2.2/A10	27	P1.0	50	P5.6
5	P2.3/A11	28	P1.1	51	P5.7
6	P2.4/A12	29	P1.2	52	XTAL2
7	P2.5/A13	30	P1.3	53	XTAL1
8	P2.6/A14	31	P1.4	54	V_{SS}
9	P2.7/A15	32	P1.5	55	$\overline{ODS}$
10	P0.7/AD7	33	P1.6	56	$\overline{IDS}$
11	P0.6/AD6	34	P1.7	57	BFLAG
12	P0.5/AD5	35	RST	58	AFLAG
13	P0.4/AD4	36	P3.0/RxD	59	P6.0
14	P0.3/AD3	37	P3.1/TxD	60	P6.1
15	P0.2/AD2	38	P3.2/$\overline{INT0}$	61	P6.2
16	P0.1/AD1	39	P3.3/$\overline{INT1}$	62	P6.3
17	P0.0/AD0	40	P3.4/T0	63	P6.4
18	V_{CC}	41	P3.5/T1	64	P6.5
19	P4.7	42	P3.6/$\overline{WR}$	65	P6.6
20	P4.6	43	P3.7/$\overline{RD}$	66	P6.7
21	P4.5	44	P5.0	67	$\overline{PSEN}$
22	P4.4	45	P5.1	68	ALE/$\overline{PROG}$
23	P4.3	46	P5.2		



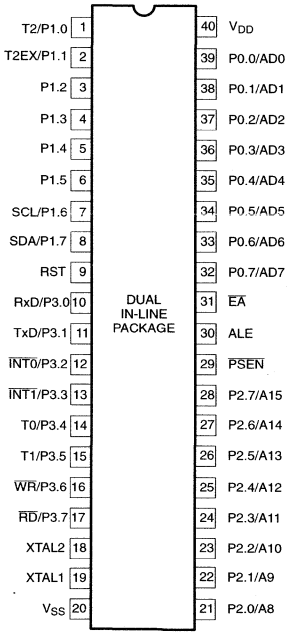
Pin	Function	Pin	Function
1	P1.5	23	P2.5/A13
2	P1.6/SCL	24	P2.6/A14
3	P1.7/SDA	25	P2.7/A15
4	RST	26	$\overline{\text{PSEN}}$
5	P3.0/RxD	27	ALE
6	NC*	28	NC*
7	P3.1/TxD	29	$\overline{\text{EA}}$
8	P3.2/$\overline{\text{INT0}}$	30	P0.7/AD7
9	P3.3/$\overline{\text{INT1}}$	31	P0.6/AD6
10	P3.4/T0	32	P0.5/AD5
11	P3.5/T1	33	P0.4/AD4
12	P3.6/$\overline{\text{WR}}$	34	P0.3/AD3
13	P3.7/$\overline{\text{RD}}$	35	P0.2/AD2
14	XTAL2	36	P0.1/AD1
15	XTAL1	37	P0.0/AD0
16	V_{SS}	38	V_{DD}
17	NC*	39	NC*
18	P2.0/A8	40	P1.0/T2
19	P2.1/A9	41	P1.1/T2EX
20	P2.2/A10	42	P1.2
21	P2.3/A11	43	P1.3
22	P2.4/A12	44	P1.4

* NO INTERNAL CONNECTION

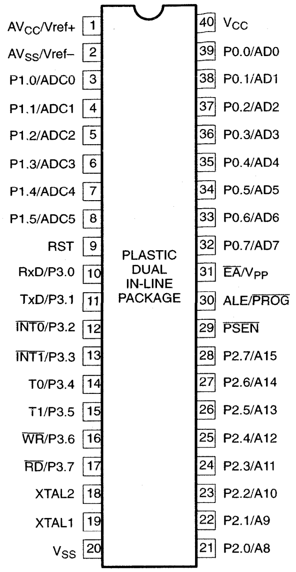


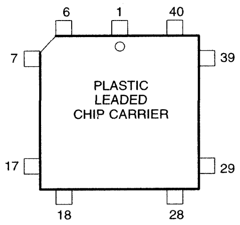
Pin	Function	Pin	Function
1	NC*	23	NC*
2	P1.0/T2	24	P2.0/A8
3	P1.1/T2EX	25	P2.1/A9
4	P1.2	26	P2.2/A10
5	P1.3	27	P2.3/A11
6	P1.4	28	P2.4/A12
7	P1.5	29	P2.5/A13
8	P1.6/SCL	30	P2.6/A14
9	P1.7/SDA	31	P2.7/A15
10	RST	32	$\overline{\text{PSEN}}$
11	P3.0/RxD	33	ALE
12	NC*	34	NC*
13	P3.1/TxD	35	$\overline{\text{EA}}$
14	P3.2/$\overline{\text{INT0}}$	36	P0.7/AD7
15	P3.3/$\overline{\text{INT1}}$	37	P0.6/AD6
16	P3.4/T0	38	P0.5/AD5
17	P3.5/T1	39	P0.4/AD4
18	P3.6/$\overline{\text{WR}}$	40	P0.3/AD3
19	P3.7/$\overline{\text{RD}}$	41	P0.2/AD2
20	XTAL2	42	P0.1/AD1
21	XTAL1	43	P0.0/AD0
22	V_{SS}	44	V_{DD}

* NO INTERNAL CONNECTIONS



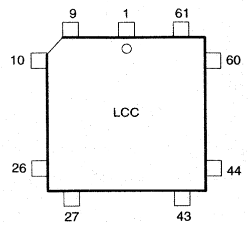
80C550/83C550/87C550





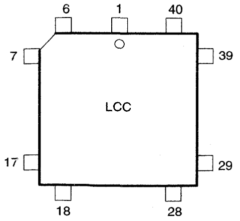
Pin	Function	Pin	Function	Pin	Function
1	AV_{CC}	16	P3.2/$\overline{\text{INT0}}$	31	P2.6/A14
2	Vref+	17	P3.3/$\overline{\text{INT1}}$	32	P2.7/A15
3	Vref–	18	P3.4/T0	33	$\overline{\text{PSEN}}$
4	AV_{SS}	19	P3.5/T1	34	ALE/$\overline{\text{PROG}}$
5	P1.0/ADC0	20	P3.6/$\overline{\text{WR}}$	35	$\overline{\text{EA}}$/V_{PP}
6	P1.1/ADC1	21	P3.7/$\overline{\text{RD}}$	36	P0.7/AD7
7	P1.2/ADC2	22	XTAL2	37	P0.6/AD6
8	P1.3/ADC3	23	XTAL1	38	P0.5/AD5
9	P1.4/ADC4	24	V_{SS}	39	P0.4/AD4
10	P1.5/ADC5	25	P2.0/A8	40	P0.3/AD3
11	P1.6/ADC6	26	P2.1/A9	41	P0.2/AD2
12	P1.7/ADC7	27	P2.2/A10	42	P0.1/AD1
13	RST	28	P2.3/A11	43	P0.0/AD0
14	P3.0/RxD	29	P2.4/A12	44	V_{CC}
15	P3.1/TxD	30	P2.5/A13		

83C453/87C453



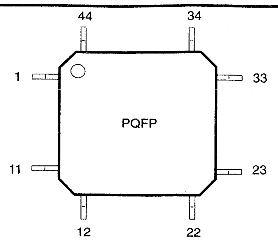
Pin	Function	Pin	Function	Pin	Function
1	$\overline{EA}/V_{PP}$	24	P4.2	47	P5.3
2	P2.0/A8	25	P4.1	48	P5.4
3	P2.1/A9	26	P4.0	49	P5.5
4	P2.2/A10	27	P1.0	50	P5.6
5	P2.3/A11	28	P1.1	51	P5.7
6	P2.4/A12	29	P1.2	52	XTAL2
7	P2.5/A13	30	P1.3	53	XTAL1
8	P2.6/A14	31	P1.4	54	V_{SS}
9	P2.7/A15	32	P1.5	55	$\overline{ODS}$
10	P0.7/AD7	33	P1.6	56	$\overline{IDS}$
11	P0.6/AD6	34	P1.7	57	BFLAG
12	P0.5/AD5	35	RST	58	AFLAG
13	P0.4/AD4	36	P3.0/RxD	59	P6.0
14	P0.3/AD3	37	P3.1/TxD	60	P6.1
15	P0.2/AD2	38	P3.2/$\overline{INT0}$	61	P6.2
16	P0.1/AD1	39	P3.3/$\overline{INT1}$	62	P6.3
17	P0.0/AD0	40	P3.4/T0	63	P6.4
18	V_{CC}	41	P3.5/T1	64	P6.5
19	P4.7	42	P3.6/$\overline{WR}$	65	P6.6
20	P4.6	43	P3.7/$\overline{RD}$	66	P6.7
21	P4.5	44	P5.0	67	$\overline{PSEN}$
22	P4.4	45	P5.1	68	ALE/$\overline{PROG}$
23	P4.3	46	P5.2		

87C524



Pin	Function	Pin	Function
1	NC*	23	NC*
2	P1.0/T2	24	P2.0/A8
3	P1.1/T2EX	25	P2.1/A9
4	P1.2	26	P2.2/A10
5	P1.3	27	P2.3/A11
6	P1.4	28	P2.4/A12
7	P1.5	29	P2.5/A13
8	P1.6/SCL	30	P2.6/A14
9	P1.7/SDA	31	P2.7/A15
10	RST	32	$\overline{\text{PSEN}}$
11	P3.0/RxD	33	ALE/$\overline{\text{PROG}}$
12	NC*	34	NC*
13	P3.1/TxD	35	$\overline{\text{EA}}$/V_{PP}
14	P3.2/$\overline{\text{INT0}}$	36	P0.7/AD7
15	P3.3/$\overline{\text{INT1}}$	37	P0.6/AD6
16	P3.4/T0	38	P0.5/AD5
17	P3.5/T1	39	P0.4/AD4
18	P3.6/$\overline{\text{WR}}$	40	P0.3/AD3
19	P3.7/$\overline{\text{RD}}$	41	P0.2/AD2
20	XTAL2	42	P0.1/AD1
21	XTAL1	43	P0.0/AD0
22	V_{SS}	44	V_{CC}

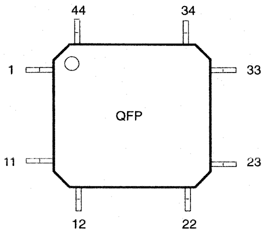
* NO INTERNAL CONNECTION



Pin	Function	Pin	Function
1	P1.5	23	P2.5/A13
2	P1.6/SCL	24	P2.6/A14
3	P1.7/SDA	25	P2.7/A15
4	RST	26	$\overline{\text{PSEN}}$
5	P3.0/RxD	27	ALE/$\overline{\text{PROG}}$
6	NC*	28	NC*
7	P3.1/TxD	29	$\overline{\text{EA}}$/V_{PP}
8	P3.2/$\overline{\text{INT0}}$	30	P0.7/AD7
9	P3.3/$\overline{\text{INT1}}$	31	P0.6/AD6
10	P3.4/T0	32	P0.5/AD5
11	P3.5/T1	33	P0.4/AD4
12	P3.6/$\overline{\text{WR}}$	34	P0.3/AD3
13	P3.7/$\overline{\text{RD}}$	35	P0.2/AD2
14	XTAL2	36	P0.1/AD1
15	XTAL1	37	P0.0/AD0
16	V_{SS}	38	V_{CC}
17	NC*	39	NC*
18	P2.0/A8	40	P1.0/T2
19	P2.1/A9	41	P1.1/T2EX
20	P2.2/A10	42	P1.2
21	P2.3/A11	43	P.13
22	P2.4/A12	44	P1.4

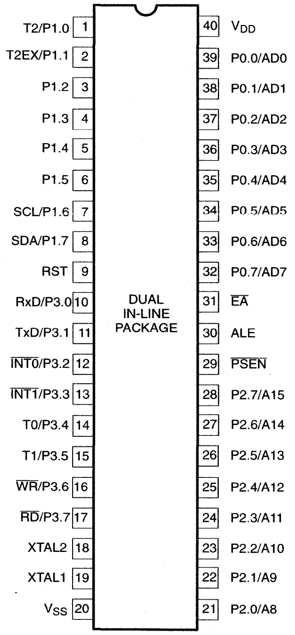
* NO INTERNAL CONNECTION

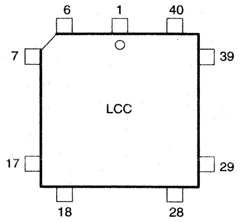
87C528, P8XC524/528



Pin	Function	Pin	Function
1	P1.5	23	P2.5/A13
2	P1.6/SCL	24	P2.6/A14
3	P1.7/SDA	25	P2.7/A15
4	RST	26	$\overline{PSEN}$
5	P3.0/RxD	27	ALE
6	NC*	28	NC*
7	P3.1/TxD	29	$\overline{EA}$
8	P3.2/$\overline{INT0}$	30	P0.7/AD7
9	P3.3/$\overline{INT1}$	31	P0.6/AD6
10	P3.4/T0	32	P0.5/AD5
11	P3.5/T1	33	P0.4/AD4
12	P3.6/$\overline{WR}$	34	P0.3/AD3
13	P3.7/$\overline{RD}$	35	P0.2/AD2
14	XTAL2	36	P0.1/AD1
15	XTAL1	37	P0.0/AD0
16	V_{SS}	38	V_{DD}
17	NC*	39	NC*
18	P2.0/A8	40	P1.0/T2
19	P2.1/A9	41	P1.1/T2EX
20	P2.2/A10	42	P1.2
21	P2.3/A11	43	P1.3
22	P2.4/A12	44	P1.4

* NO INTERNAL CONNECTION

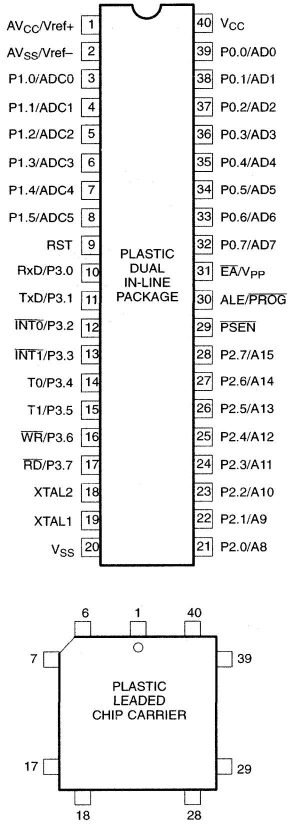




Pin	Function	Pin	Function
1	NC*	23	NC*
2	P1.0/T2	24	P2.0/A8
3	P1.1/T2EX	25	P2.1/A9
4	P1.2	26	P2.2/A10
5	P1.3	27	P2.3/A11
6	P1.4	28	P2.4/A12
7	P1.5	29	P2.5/A13
8	P1.6/SCL	30	P2.6/A14
9	P1.7/SDA	31	P2.7/A15
10	RST	32	$\overline{PSEN}$
11	P3.0/RxD	33	ALE
12	NC*	34	NC*
13	P3.1/TxD	35	$\overline{EA}$
14	P3.2/$\overline{INT0}$	36	P0.7/AD7
15	P3.3/$\overline{INT1}$	37	P0.6/AD6
16	P3.4/T0	38	P0.5/AD5
17	P3.5/T1	39	P0.4/AD4
18	P3.6/$\overline{WR}$	40	P0.3/AD3
19	P3.7/$\overline{RD}$	41	P0.2/AD2
20	XTAL2	42	P0.1/AD1
21	XTAL1	43	P0.0/AD0
22	V_{SS}	44	V_{DD}

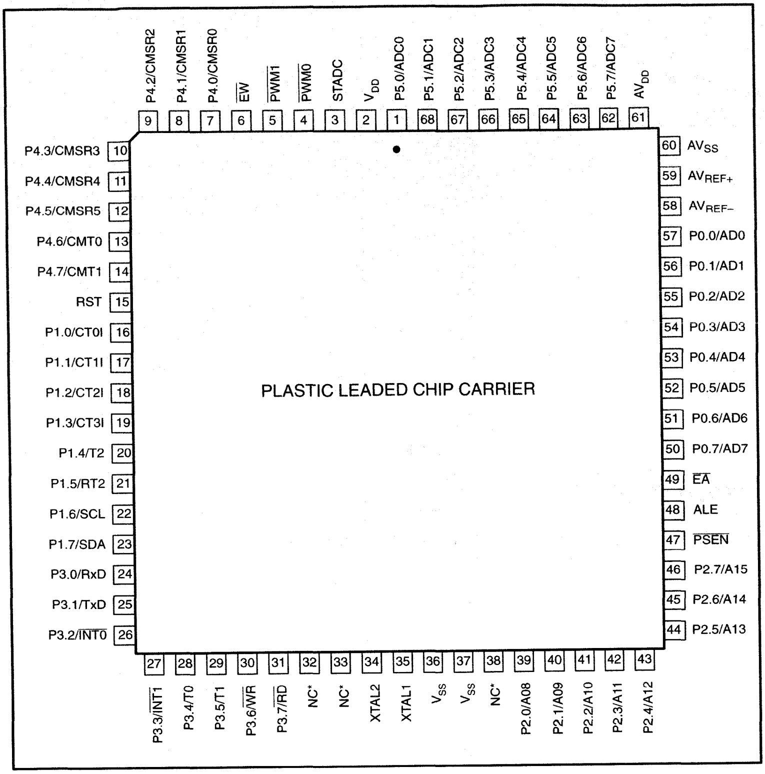
* NO INTERNAL CONNECTIONS

80C550/83C550/87C550

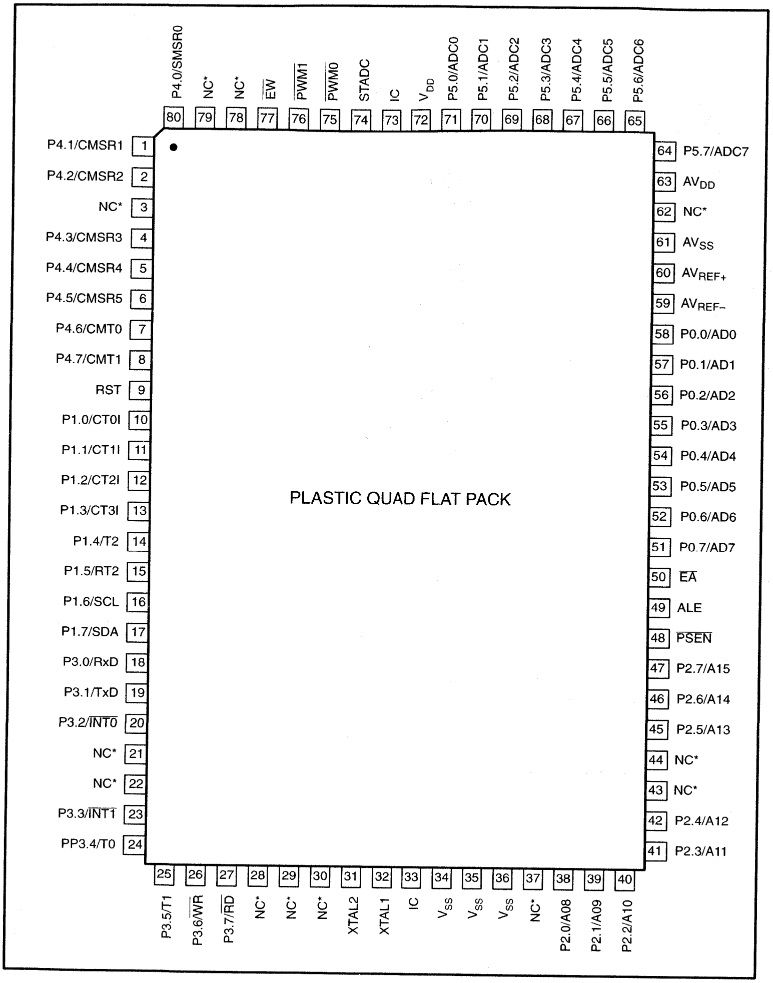


Pin	Function	Pin	Function	Pin	Function
1	AV$_{CC}$	16	P3.2/$\overline{\text{INT0}}$	31	P2.6/A14
2	Vref+	17	P3.3/$\overline{\text{INT1}}$	32	P2.7/A15
3	Vref–	18	P3.4/T0	33	$\overline{\text{PSEN}}$
4	AV$_{SS}$	19	P3.5/T1	34	ALE/$\overline{\text{PROG}}$
5	P1.0/ADC0	20	P3.6/$\overline{\text{WR}}$	35	$\overline{\text{EA}}$/V$_{PP}$
6	P1.1/ADC1	21	P3.7/$\overline{\text{RD}}$	36	P0.7/AD7
7	P1.2/ADC2	22	XTAL2	37	P0.6/AD6
8	P1.3/ADC3	23	XTAL1	38	P0.5/AD5
9	P1.4/ADC4	24	V$_{SS}$	39	P0.4/AD4
10	P1.5/ADC5	25	P2.0/A8	40	P0.3/AD3
11	P1.6/ADC6	26	P2.1/A9	41	P0.2/AD2
12	P1.7/ADC7	27	P2.2/A10	42	P0.1/AD1
13	RST	28	P2.3/A11	43	P0.0/AD0
14	P3.0/RxD	29	P2.4/A12	44	V$_{CC}$
15	P3.1/TxD	30	P2.5/A13		

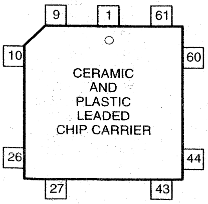
80C552/83C552



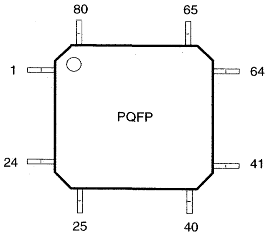
80C552/83C552 (Continued)



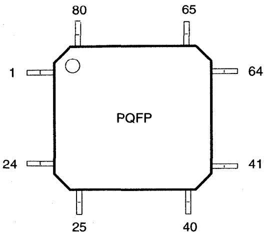
87C552



Pin	Function	Pin	Function
1	P5.0/ADC0	35	XTAL1
2	V_{DD}	36	V_{SS}
3	STADC	37	V_{SS}
4	$\overline{PWM0}$	38	NC
5	$\overline{PWM1}$	39	P2.0/A08
6	$\overline{EW}$	40	P2.1/A09
7	P4.0/CMSR0	41	P2.2/A10
8	P4.1/CMSR1	42	P2.3/A11
9	P4.2/CMSR2	43	P2.4/A12
10	P4.3/CMSR3	44	P2.5/A13
11	P4.4/CMSR4	45	P2.6/A14
12	P4.5/CMSR5	46	P2.7/A15
13	P4.6/CMT0	47	$\overline{PSEN}$
14	P4.7/CMT1	48	ALE/$\overline{PROG}$
15	RST	49	$\overline{EA}$/V_{PP}
16	P1.0/CT0I	50	P0.7/AD7
17	P1.1/CT1I	51	P0.6/AD6
18	P1.2/CT2I	52	P0.5/AD5
19	P1.3/CT3I	53	P0.4/AD4
20	P1.4/T2	54	P0.3/AD3
21	P1.5/RT2	55	P0.2/AD2
22	P1.6/SCL	56	P0.1/AD1
23	P1.7/SDA	57	P0.0/AD0
24	P3.0/RxD	58	AVref−
25	P3.1/TxD	59	AVref+
26	P3.2/$\overline{INT0}$	60	AV_{SS}
27	P3.3/$\overline{INT1}$	61	AV_{DD}
28	P3.4/T0	62	P5.7/ADC7
29	P3.5/T1	63	P5.6/ADC6
30	P3.6/$\overline{WR}$	64	P5.5/ADC5
31	P3.7/$\overline{RD}$	65	P5.4/ADC4
32	NC	66	P5.3/ADC3
33	NC	67	P5.2/ADC2
34	XTAL2	68	P5.1/ADC1



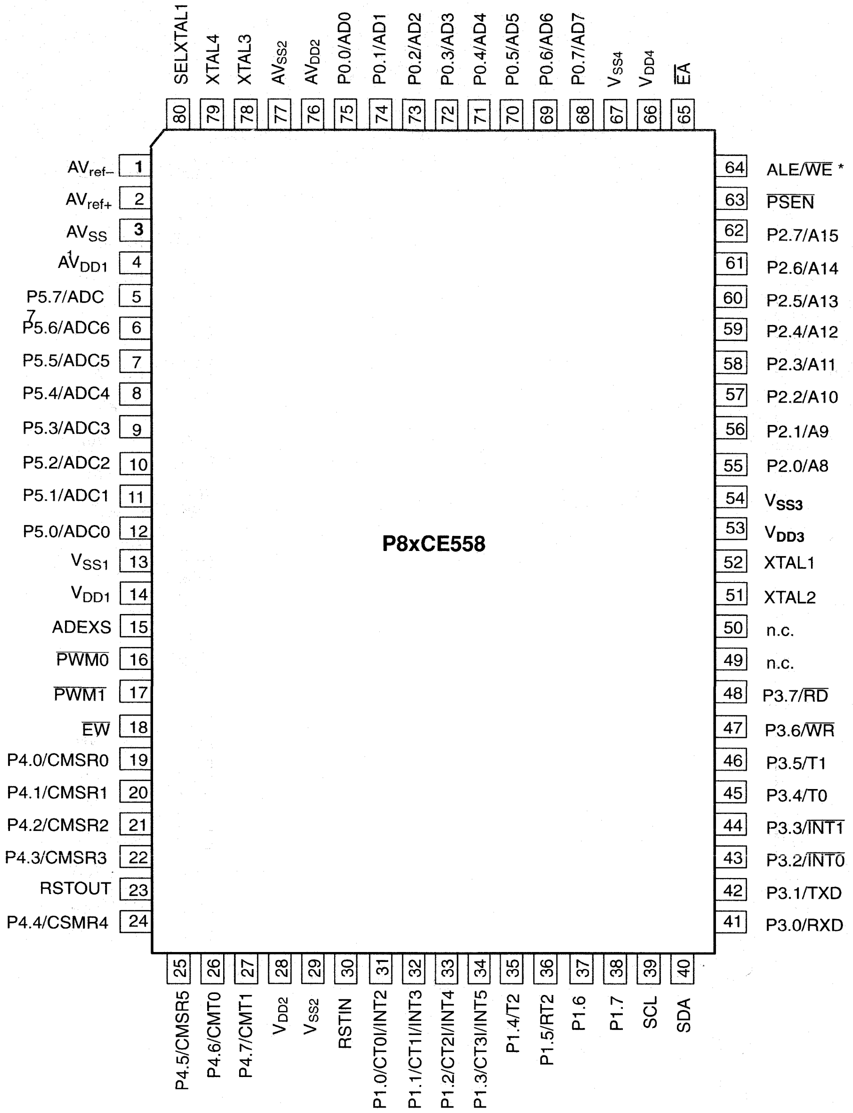
87C552 (Continued)



Pin	Function	Pin	Function	Pin	Function	Pin	Function
1	P4.1/CMSR1	21	NC	41	P2.3/A11	61	AV_{SS}
2	P4.2/CMSR2	22	NC	42	P2.4/A12	62	NC
3	NC	23	P3.3/$\overline{INT1}$	43	NC	63	AV_{DD}
4	P4.3/CMSR3	24	P3.4/T0	44	NC	64	P5.7/ADC7
5	P4.4/CMSR4	25	P3.5/T1	45	P2.5/A13	65	P5.6/ADC6
6	P4.5/CMSR5	26	P3.6/$\overline{WR}$	46	P2.6/A14	66	P5.5/ADC5
7	P4.6/CMT0	27	P3.7/$\overline{RD}$	47	P2.7/A15	67	P5.4/ADC4
8	P4.7/CMT1	28	NC	48	$\overline{PSEN}$	68	P5.3/ADC3
9	RST	29	NC	49	ALE/$\overline{PROG}$	69	P5.2/ADC2
10	P1.0/CT0I	30	NC	50	$\overline{EA}/V_{PP}$	70	P5.1/ADC1
11	P1.1/CT1I	31	XTAL2	51	P0.7/AD7	71	P5.0/ADC0
12	P1.2/CT2I	32	XTAL1	52	P0.6/AD6	72	V_{DD}
13	P1.3/CT3I	33	IC	53	P0.5/AD5	73	IC
14	P1.4/T2	34	V_{SS}	54	P0.4/AD4	74	STADC
15	P1.5/RT2	35	V_{SS}	55	P0.3/AD3	75	$\overline{PWM0}$
16	P1.6/SCL	36	V_{SS}	56	P0.2/AD2	76	$\overline{PWM1}$
17	P1.7/SDA	37	NC	57	P0.1/AD1	77	$\overline{EW}$
18	P3.0/RxD	38	P2.0/A08	58	P0.0/AD0	78	NC
19	P3.1/TxD	39	P2.1/A09	59	AVref–	79	NC
20	P3.2/$\overline{INT0}$	40	P2.2/A10	60	AVref+	80	P4.0/CMSR0

NC = Not Connected
IC = Internally Connected (do not use)

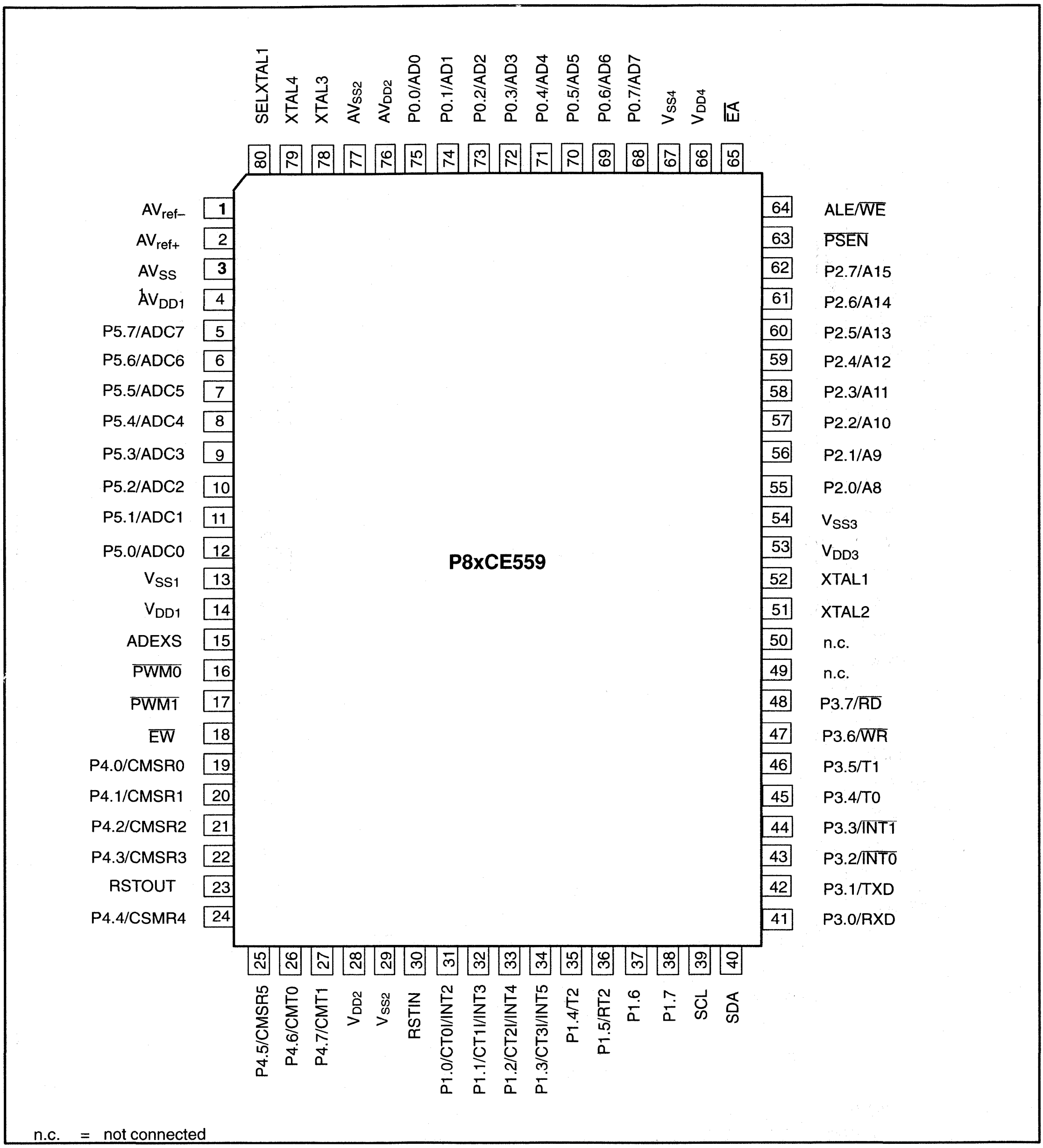
P80CE558/P83CE558/P89CE558



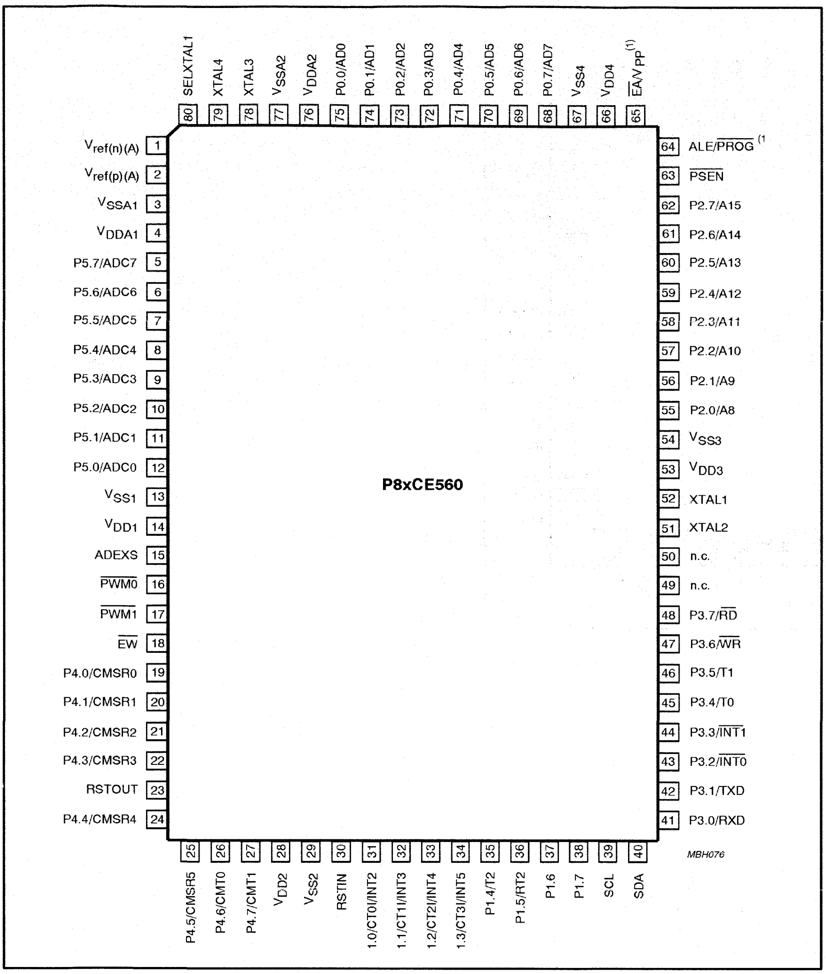
n.c. = not connected

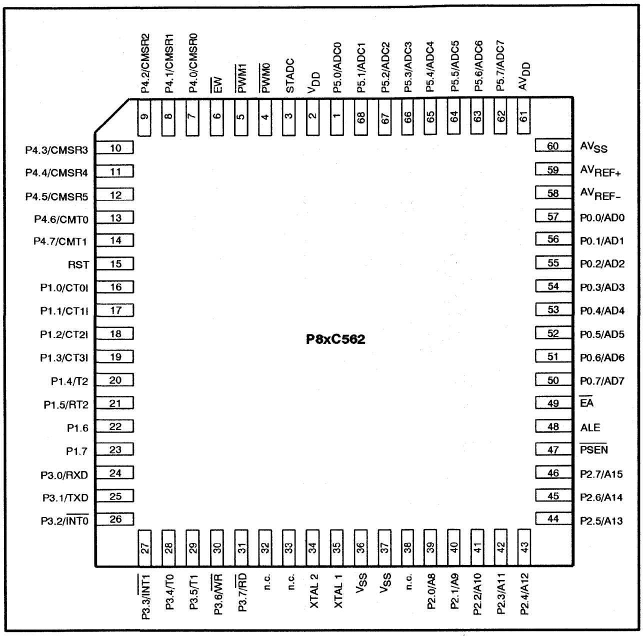
* = only P89CE558 with alternate function WE

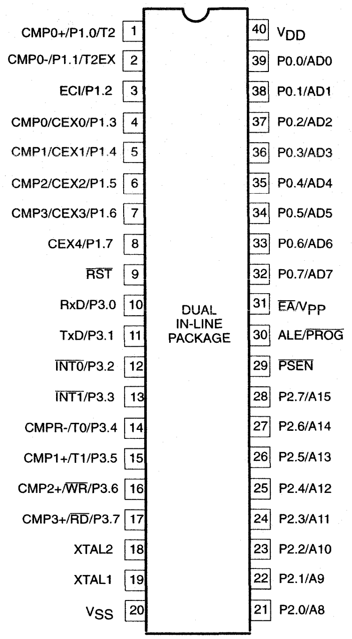
P83CE559/P80CE559

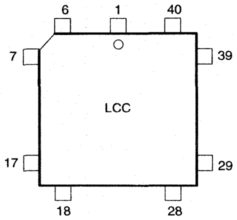


P8XCE560


P8xCE560
1 Vref(n)(A)
2 Vref(p)(A)
3 VSSA1
4 VDDA1
5 P5.7/ADC7
6 P5.6/ADC6
7 P5.5/ADC5
8 P5.4/ADC4
9 P5.3/ADC3
10 P5.2/ADC2
11 P5.1/ADC1
12 P5.0/ADC0
13 VSS1
14 VDD1
15 ADEXS
16 PWM0
17 PWM1
18 EW
19 P4.0/CMSR0
20 P4.1/CMSR1
21 P4.2/CMSR2
22 P4.3/CMSR3
23 RSTOUT
24 P4.4/CMSR4
25 P4.5/CMSR5
26 P4.6/CMT0
27 P4.7/CMT1
28 VDD2
29 VSS2
30 RSTIN
31 1.0/CT0I/INT2
32 1.1/CT1I/INT3
33 1.2/CT2I/INT4
34 1.3/CT3I/INT5
35 P1.4/T2
36 P1.5/RT2
37 P1.6
38 P1.7
39 SCL
40 SDA
41 P3.0/RXD
42 P3.1/TXD
43 P3.2/INT0
44 P3.3/INT1
45 P3.4/T0
46 P3.5/T1
47 P3.6/WR
48 P3.7/RD
49 n.c.
50 n.c.
51 XTAL2
52 XTAL1
53 VDD3
54 VSS3
55 P2.0/A8
56 P2.1/A9
57 P2.2/A10
58 P2.3/A11
59 P2.4/A12
60 P2.5/A13
61 P2.6/A14
62 P2.7/A15
63 PSEN
64 ALE/PROG (1
65 EA/VPP (1)
66 VDD4
67 VSS4
68 P0.7/AD7
69 P0.6/AD6
70 P0.5/AD5
71 P0.4/AD4
72 P0.3/AD3
73 P0.2/AD2
74 P0.1/AD1
75 P0.0/AD0
76 VDDA2
77 VSSA2
78 XTAL3
79 XTAL4
80 SELXTAL1
MBH076


P8xC562
9 P4.2/CMSR2
8 P4.1/CMSR1
7 P4.0/CMSR0
6 EW
5 PWM1
4 PWM0
3 STADC
2 VDD
1 P5.0/ADC0
68 P5.1/ADC1
67 P5.2/ADC2
66 P5.3/ADC3
65 P5.4/ADC4
64 P5.5/ADC5
63 P5.6/ADC6
62 P5.7/ADC7
61 AVDD
10 P4.3/CMSR3
11 P4.4/CMSR4
12 P4.5/CMSR5
13 P4.6/CMT0
14 P4.7/CMT1
15 RST
16 P1.0/CT0I
17 P1.1/CT1I
18 P1.2/CT2I
19 P1.3/CT3I
20 P1.4/T2
21 P1.5/RT2
22 P1.6
23 P1.7
24 P3.0/RXD
25 P3.1/TXD
26 P3.2/INT0
60 AVSS
59 AVREF+
58 AVREF-
57 P0.0/AD0
56 P0.1/AD1
55 P0.2/AD2
54 P0.3/AD3
53 P0.4/AD4
52 P0.5/AD5
51 P0.6/AD6
50 P0.7/AD7
49 EA
48 ALE
47 PSEN
46 P2.7/A15
45 P2.6/A14
44 P2.5/A13
27 P3.3/INT1
28 P3.4/T0
29 P3.5/T1
30 P3.6/WR
31 P3.7/RD
32 n.c.
33 n.c.
34 XTAL 2
35 XTAL 1
36 VSS
37 VSS
38 n.c.
39 P2.0/A8
40 P2.1/A9
41 P2.2/A10
42 P2.3/A11
43 P2.4/A12

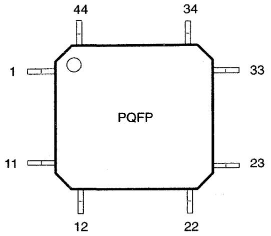




Pin	Function	Pin	Function
1	NC*	23	NC*
2	T2/P1.0/CMP0+	24	P2.0/A8
3	T2EX/P1.1/CMP0–	25	P2.1/A9
4	P1.2/ECI	26	P2.2/A10
5	P1.3/CMP0/CEX0	27	P2.3/A11
6	P1.4/CMP1/CEX1	28	P2.4/A12
7	P1.5/CMP2/CEX2	29	P2.5/A13
8	P1.6/CMP3/CEX3	30	P2.6/A14
9	P1.7/CEX4	31	P2.7/A15
10	$\overline{\text{RST}}$	32	$\overline{\text{PSEN}}$
11	RxD/P3.0	33	ALE/$\overline{\text{PROG}}$
12	NC*	34	NC*
13	TxD/P3.1	35	$\overline{\text{EA}}$/V_{PP}
14	$\overline{\text{INT0}}$/P3.2	36	P0.7/AD7
15	$\overline{\text{INT1}}$/P3.3	37	P0.6/AD6
16	T0/P3.4/CMPR–	38	P0.5/AD5
17	T1/P3.5/CMP1+	39	P0.4/AD4
18	$\overline{\text{WR}}$/P3.6/CMP2+	40	P0.3/AD3
19	$\overline{\text{RD}}$/P3.7/CMP3+	41	P0.2/AD2
20	XTAL2	42	P0.1/AD1
21	XTAL1	43	P0.0/AD0
22	V_{SS}	44	V_{CC}

* NO INTERNAL CONNECTION

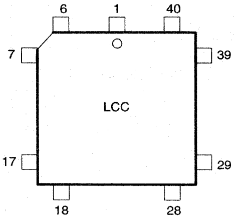
80C575/83C575/87C575 (Continued)



Pin	Function	Pin	Function
1	P1.5/CMP2/CEX2	23	P2.5/A13
2	P1.6/CMP3/CEX3	24	P2.6/A14
3	P1.7/CEX4	25	P2.7/A15
4	$\overline{\text{RST}}$	26	$\overline{\text{PSEN}}$
5	RxD/P3.0	27	ALE/$\overline{\text{PROG}}$
6	NC*	28	NC*
7	TxD/P3.1	29	$\overline{\text{EA}}$/V_{PP}
8	$\overline{\text{INT}}$0/P3.2	30	P0.7/AD7
9	$\overline{\text{INT}}$1/P3.3	31	P0.6/AD6
10	T0/P3.4/CMPR–	32	P0.5/AD5
11	T1/P3.5/CMP1+	33	P0.4/AD4
12	$\overline{\text{WR}}$/P3.6/CMP2+	34	P0.3/AD3
13	$\overline{\text{RD}}$/P3.7CMP3+	35	P0.2/AD2
14	XTAL2	36	P0.1/AD1
15	XTAL1	37	P0.0/AD0
16	V_{SS}	38	V_{CC}
17	NC*	39	NC*
18	P2.0/A8	40	T2/P1.0/CMP0+
19	P2.1/A9	41	T2EX/P1.1/CMP0–
20	P2.2/A10	42	P1.2/ECI
21	P2.3/A11	43	P1.3/CMP0/CEX0
22	P2.4/A12	44	P1.4/CMP1/CEX1

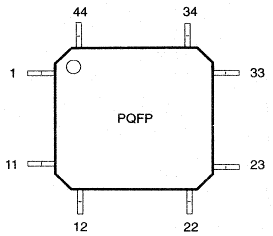
* NO INTERNAL CONNECTION

83C576/87C576



Pin	Function	Pin	Function	Pin	Function
1	NC*	16	T0/P3.4/CMP1+	31	P2.7/A15/PWM1/ECI
2	$+V_{REF}/AV_{CC}$	17	T1/P3.5/CMPR–	32	$\overline{PSEN}$
3	$-V_{REF}/AV_{SS}$	18	$\overline{WR}$/P3.6/CMP0+	33	ALE/$\overline{PROG}$
4	ADIN0/P1.0	19	$\overline{RD}$/P3.7/CMP0–	34	NC*
5	ADIN1/P1.1	20	XTAL2	35	$\overline{EA}/V_{PP}$
6	ADIN2/P1.2	21	XTAL1	36	P0.7/AD7/DB7
7	ADIN3/P1.3	22	V_{SS}	37	P0.6/AD6/DB6
8	ADIN4/P1.4	23	NC*	38	P0.5/AD5/DB5
9	ADIN5/P1.5	24	P2.0/A8/CEX0/CMP0	39	P0.4/AD4/DB4
10	$\overline{RST}$	25	P2.1/A9/CEX1/CMP1	40	P0.3/AD3/DB3
11	RxD/P3.0	26	P2.2/A10/CEX2/CMP2	41	P0.2/AD2/DB2
12	NC*	27	P2.3/A11/CEX3/CMP3	42	P0.1/AD1/DB1
13	TxD/P3.1	28	P2.4/A12/T2EX/A0	43	P0.0/AD0/DB0
14	$\overline{INT}$0/P3.2/CMP3+	29	P2.5/A13/T2/$\overline{CS}$	44	V_{CC}
15	$\overline{INT}$1/P3.3/CMP2+	30	P2.6/A14/CEX4/PWM0		

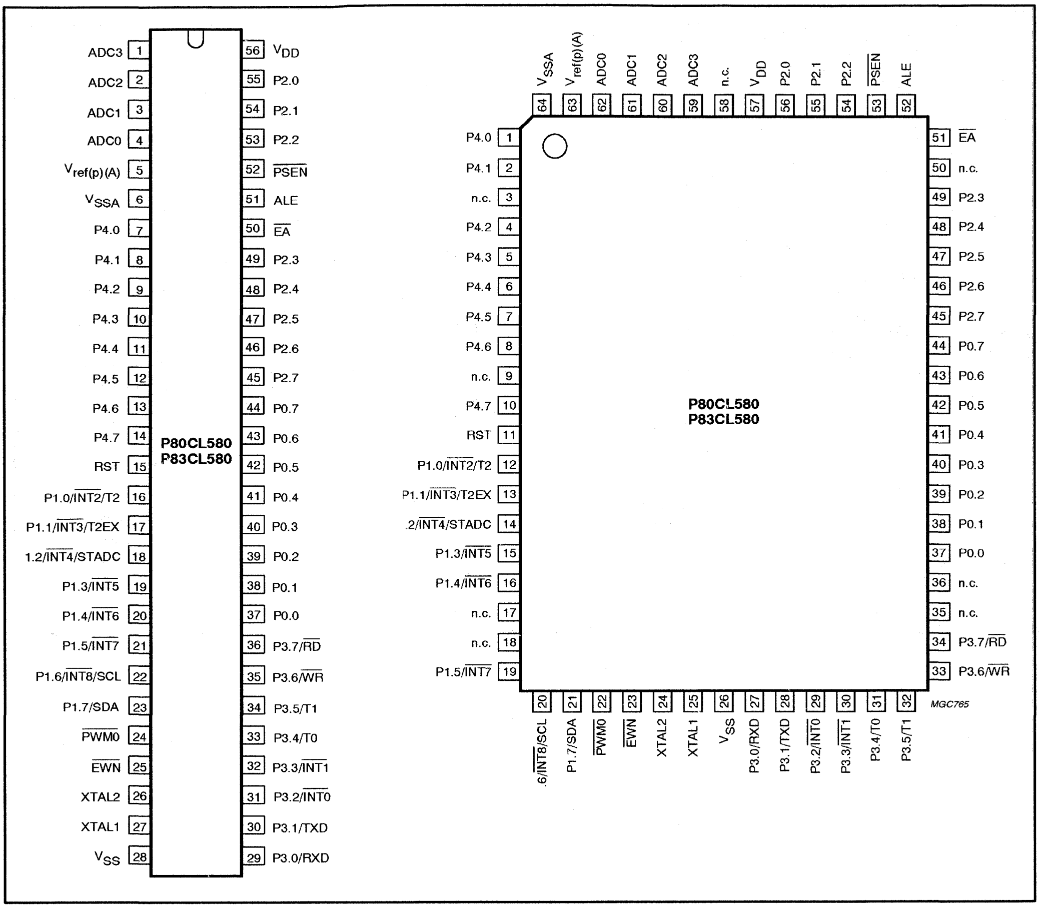
* NO INTERNAL CONNECTION



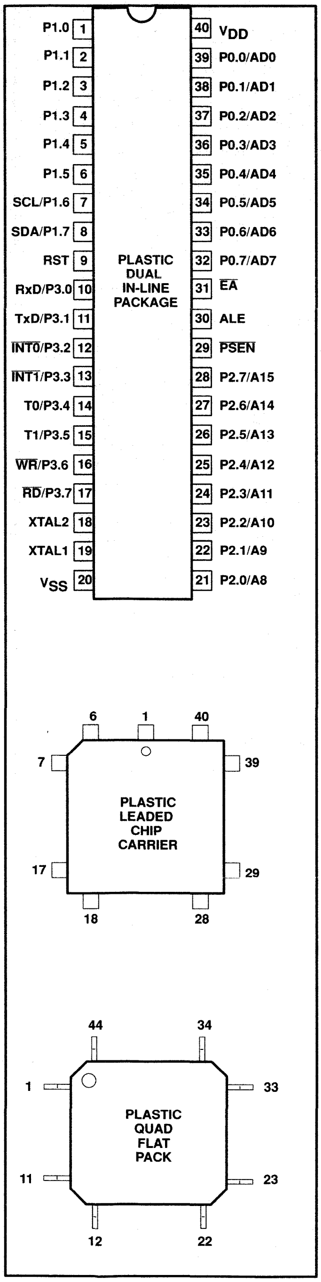
Pin	Function	Pin	Function	Pin	Function
1	ADIN3/P1.3	16	V_{SS}	31	P0.6/AD6/DB6
2	ADIN4/P1.4	17	NC*	32	P0.5/AD5/DB5
3	ADIN5/P1.5	18	P2.0/A8/CEX0/CMP0	33	P0.4/AD4/DB4
4	$\overline{RST}$	19	P2.1/A9/CEX1/CMP1	34	P0.3/AD3/DB3
5	RxD/P3.0	20	P2.2/A10/CEX2/CMP2	35	P0.2/AD2/DB2
6	NC*	21	P2.3/A11/CEX3/CMP3	36	P0.1/AD1/DB1
7	TXD/P3.1	22	P2.4/A12/T2EX/A0	37	P0.0/AD0/DB0
8	$\overline{INT}$0/P3.2/CMP3+	23	P2.5/A13/T2/$\overline{CS}$	38	V_{CC}
9	$\overline{INT}$1/P3.3/CMP2+	24	P2.6/A14/CEX4/PWM0	39	NC*
10	T0/P3.4/CMP1+	25	P2.7/A15/PWM1/ECI	40	$+V_{REF}/AV_{CC}$
11	T1/P3.5/CMPR–	26	$\overline{PSEN}$	41	$-V_{REF}/AV_{SS}$
12	$\overline{WR}$/P3.6/CMP0+	27	ALE/$\overline{PROG}$	42	ADIN0/P1.0
13	$\overline{RD}$/P3.7CMP0–	28	NC*	43	ADIN1/P1.1
14	XTAL2	29	$\overline{EA}/V_{PP}$	44	ADIN2/P1.2
15	XTAL1	30	P0.7/AD7/DB7		

* NO INTERNAL CONNECTION

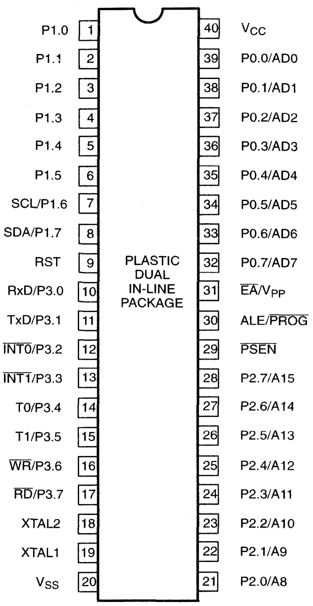
8CL580

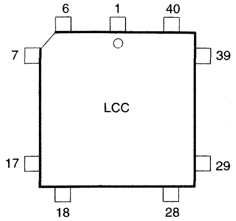

ADC3 1
ADC2 2
ADC1 3
ADC0 4
Vref(p)(A) 5
VSSA 6
P4.0 7
P4.1 8
P4.2 9
P4.3 10
P4.4 11
P4.5 12
P4.6 13
P4.7 14
RST 15
P1.0/INT2/T2 16
P1.1/INT3/T2EX 17
1.2/INT4/STADC 18
P1.3/INT5 19
P1.4/INT6 20
P1.5/INT7 21
P1.6/INT8/SCL 22
P1.7/SDA 23
PWM0 24
EWN 25
XTAL2 26
XTAL1 27
VSS 28
56 VDD
55 P2.0
54 P2.1
53 P2.2
52 PSEN
51 ALE
50 EA
49 P2.3
48 P2.4
47 P2.5
46 P2.6
45 P2.7
44 P0.7
43 P0.6
42 P0.5
41 P0.4
40 P0.3
39 P0.2
38 P0.1
37 P0.0
36 P3.7/RD
35 P3.6/WR
34 P3.5/T1
33 P3.4/T0
32 P3.3/INT1
31 P3.2/INT0
30 P3.1/TXD
29 P3.0/RXD
P80CL580
P83CL580
VSSA 64
Vref(p)(A) 63
ADC0 62
ADC1 61
ADC2 60
ADC3 59
n.c. 58
VDD 57
P2.0 56
P2.1 55
P2.2 54
PSEN 53
ALE 52
P4.0 1
P4.1 2
n.c. 3
P4.2 4
P4.3 5
P4.4 6
P4.5 7
P4.6 8
n.c. 9
P4.7 10
RST 11
P1.0/INT2/T2 12
P1.1/INT3/T2EX 13
.2/INT4/STADC 14
P1.3/INT5 15
P1.4/INT6 16
n.c. 17
n.c. 18
P1.5/INT7 19
51 EA
50 n.c.
49 P2.3
48 P2.4
47 P2.5
46 P2.6
45 P2.7
44 P0.7
43 P0.6
42 P0.5
41 P0.4
40 P0.3
39 P0.2
38 P0.1
37 P0.0
36 n.c.
35 n.c.
34 P3.7/RD
33 P3.6/WR
P80CL580
P83CL580
20 .6/INT8/SCL
21 P1.7/SDA
22 PWM0
23 EWN
24 XTAL2
25 XTAL1
26 VSS
27 P3.0/RXD
28 P3.1/TXD
29 P3.2/INT0
30 P3.3/INT1
31 P3.4/T0
32 P3.5/T1
MGC765

80C652/83C652



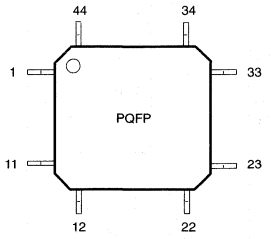
87C652, 83C654





Pin	Function	Pin	Function
1	NC*	23	NC*
2	P1.0	24	P2.0/A8
3	P1.1	25	P2.1/A9
4	P1.2	26	P2.2/A10
5	P1.3	27	P2.3/A11
6	P1.4	28	P2.4/A12
7	P1.5	29	P2.5/A13
8	P1.6/SCL	30	P2.6/A14
9	P1.7/SDA	31	P2.7/A15
10	RST	32	$\overline{\text{PSEN}}$
11	P3.0/RxD	33	ALE/$\overline{\text{PROG}}$
12	NC*	34	NC*
13	P3.1/TxD	35	$\overline{\text{EA}}$/V_{PP}
14	P3.2/$\overline{\text{INT0}}$	36	P0.7/AD7
15	P3.3/$\overline{\text{INT1}}$	37	P0.6/AD6
16	P3.4/T0	38	P0.5/AD5
17	P3.5/T1	39	P0.4/AD4
18	P3.6/$\overline{\text{WR}}$	40	P0.3/AD3
19	P3.7/$\overline{\text{RD}}$	41	P0.2/AD2
20	XTAL2	42	P0.1/AD1
21	XTAL1	43	P0.0/AD0
22	V_{SS}	44	V_{CC}

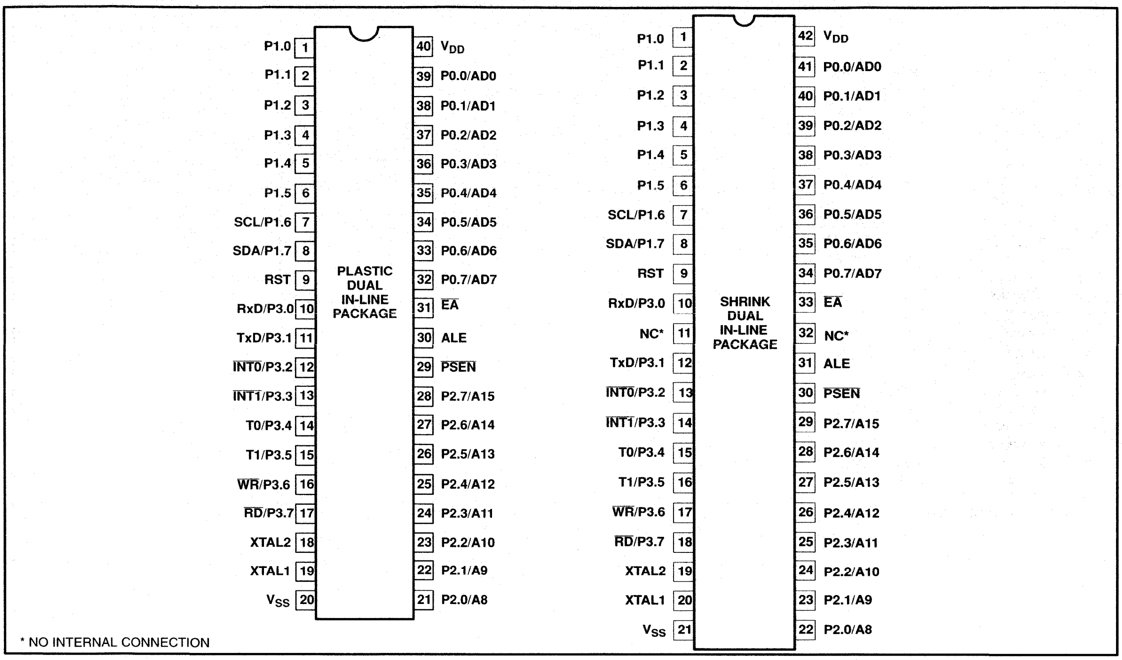
* NO INTERNAL CONNECTION



Pin	Function	Pin	Function
1	P1.5	23	P2.5/A13
2	P1.6/SCL	24	P2.6/A14
3	P1.7/SDA	25	P2.7/A15
4	RST	26	$\overline{\text{PSEN}}$
5	P3.0/RxD	27	ALE/$\overline{\text{PROG}}$
6	NC*	28	NC*
7	P3.1/TxD	29	$\overline{\text{EA}}$/V_{PP}
8	P3.2/$\overline{\text{INT0}}$	30	P0.7/AD7
9	P3.3/$\overline{\text{INT1}}$	31	P0.6/AD6
10	P3.4/T0	32	P0.5/AD5
11	P3.5/T1	33	P0.4/AD4
12	P3.6/$\overline{\text{WR}}$	34	P0.3/AD3
13	P3.7/$\overline{\text{RD}}$	35	P0.2/AD2
14	XTAL2	36	P0.1/AD1
15	XTAL1	37	P0.0/AD0
16	V_{SS}	38	V_{CC}
17	NC*	39	NC*
18	P2.0/A8	40	P1.0
19	P2.1/A9	41	P1.1
20	P2.2/A10	42	P1.2
21	P2.3/A11	43	P.13
22	P2.4/A12	44	P1.4

* NO INTERNAL CONNECTION

83C654

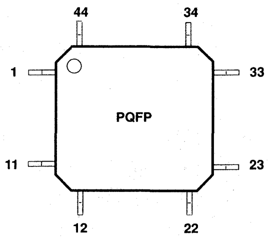


* NO INTERNAL CONNECTION

83C654 (Continued)

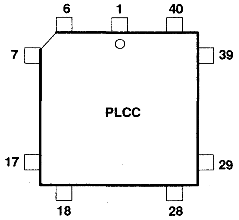
Pin	Function	Pin	Function
1	P1.5	23	P2.5/A13
2	P1.6/SCL	24	P2.6/A14
3	P1.7/SDA	25	P2.7/A15
4	RST	26	$\overline{\text{PSEN}}$
5	P3.0/RxD	27	ALE
6	V_{SS4}	28	V_{SS2}
7	P3.1/TxD	29	$\overline{\text{EA}}$/V_{PP}
8	P3.2/$\overline{\text{INT0}}$	30	P0.7/AD7
9	P3.3/$\overline{\text{INT1}}$	31	P0.6/AD6
10	P3.4/T0	32	P0.5/AD5
11	P3.5/T1	33	P0.4/AD4
12	P3.6/$\overline{\text{WR}}$	34	P0.3/AD3
13	P3.7/$\overline{\text{RD}}$	35	P0.2/AD2
14	XTAL2	36	P0.1/AD1
15	XTAL1	37	P0.0/AD0
16	V_{SS1}	38	V_{DD}
17	NC*	39	V_{SS3}
18	P2.0/A8	40	P1.0
19	P2.1/A9	41	P1.1
20	P2.2/A10	42	P1.2
21	P2.3/A11	43	P1.3
22	P2.4/A12	44	P1.4

* NO INTERNAL CONNECTION



Pin	Function	Pin	Function
1	NC*	23	NC*
2	P1.0	24	P2.0/A8
3	P1.1	25	P2.1/A9
4	P1.2	26	P2.2/A10
5	P1.3	27	P2.3/A11
6	P1.4	28	P2.4/A12
7	P1.5	29	P2.5/A13
8	P1.6/SCL	30	P2.6/A14
9	P1.7/SDA	31	P2.7/A15
10	RST	32	$\overline{\text{PSEN}}$
11	P3.0/RxD	33	ALE
12	NC*	34	NC*
13	P3.1/TxD	35	$\overline{\text{EA}}$
14	P3.2/$\overline{\text{INT0}}$	36	P0.7/AD7
15	P3.3/$\overline{\text{INT1}}$	37	P0.6/AD6
16	P3.4/T0	38	P0.5/AD5
17	P3.5/T1	39	P0.4/AD4
18	P3.6/$\overline{\text{WR}}$	40	P0.3/AD3
19	P3.7/$\overline{\text{RD}}$	41	P0.2/AD2
20	XTAL2	42	P0.1/AD1
21	XTAL1	43	P0.0/AD0
22	V_{SS}	44	V_{DD}

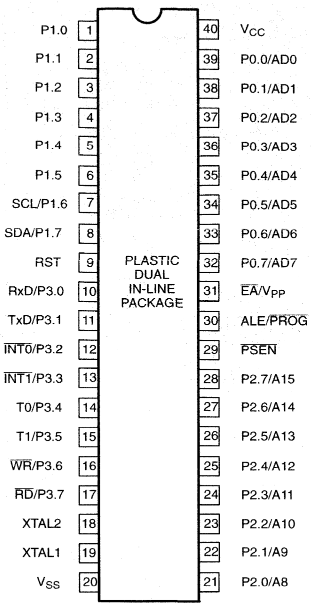
* NO INTERNAL CONNECTION

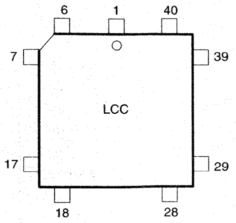


NOTES TO QFP ONLY:

1. Due to EMC improvements, all V_{SS} pins (6, 16, 28, 39) must be connected to V_{SS} on the 80C652/83C654.

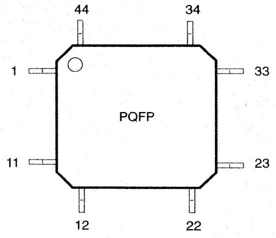
87C654





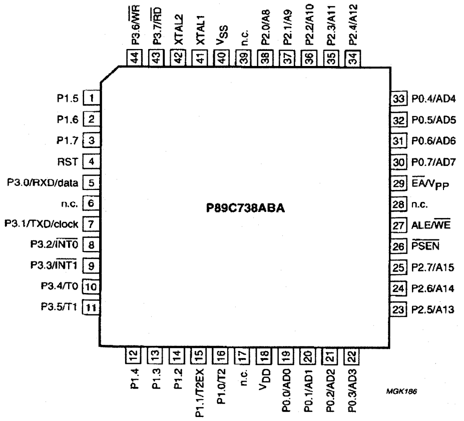
Pin	Function	Pin	Function
1	NC*	23	NC*
2	P1.0	24	P2.0/A8
3	P1.1	25	P2.1/A9
4	P1.2	26	P2.2/A10
5	P1.3	27	P2.3/A11
6	P1.4	28	P2.4/A12
7	P1.5	29	P2.5/A13
8	P1.6/SCL	30	P2.6/A14
9	P1.7/SDA	31	P2.7/A15
10	RST	32	$\overline{\text{PSEN}}$
11	P3.0/RxD	33	ALE/$\overline{\text{PROG}}$
12	NC*	34	NC*
13	P3.1/TxD	35	$\overline{\text{EA}}$/V_{PP}
14	P3.2/$\overline{\text{INT0}}$	36	P0.7/AD7
15	P3.3/$\overline{\text{INT1}}$	37	P0.6/AD6
16	P3.4/T0	38	P0.5/AD5
17	P3.5/T1	39	P0.4/AD4
18	P3.6/$\overline{\text{WR}}$	40	P0.3/AD3
19	P3.7/$\overline{\text{RD}}$	41	P0.2/AD2
20	XTAL2	42	P0.1/AD1
21	XTAL1	43	P0.0/AD0
22	V_{SS}	44	V_{CC}

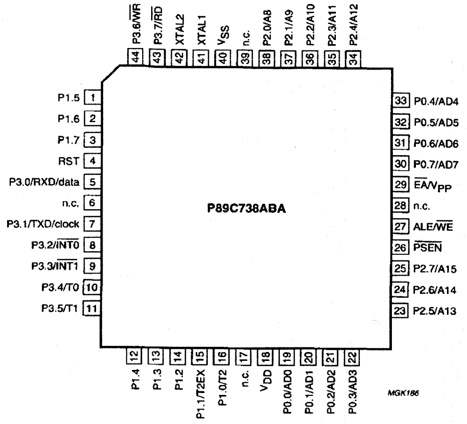
* NO INTERNAL CONNECTION

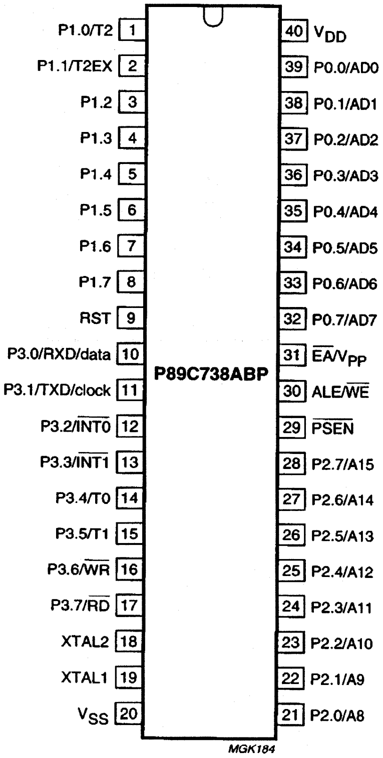


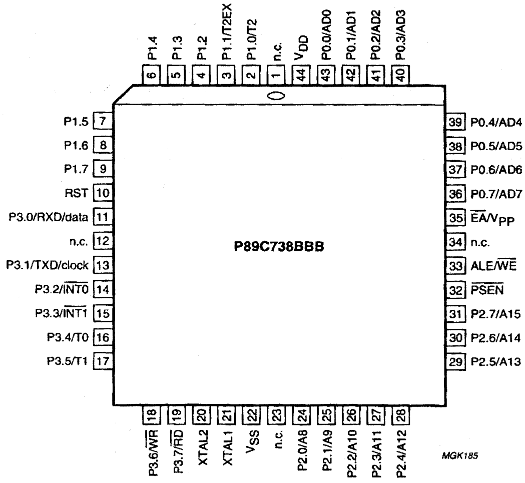
Pin	Function	Pin	Function
1	P1.5	23	P2.5/A13
2	P1.6/SCL	24	P2.6/A14
3	P1.7/SDA	25	P2.7/A15
4	RST	26	$\overline{\text{PSEN}}$
5	P3.0/RxD	27	ALE/$\overline{\text{PROG}}$
6	NC*	28	NC*
7	P3.1/TxD	29	$\overline{\text{EA}}$/V_{PP}
8	P3.2/$\overline{\text{INT0}}$	30	P0.7/AD7
9	P3.3/$\overline{\text{INT1}}$	31	P0.6/AD6
10	P3.4/T0	32	P0.5/AD5
11	P3.5/T1	33	P0.4/AD4
12	P3.6/$\overline{\text{WR}}$	34	P0.3/AD3
13	P3.7/$\overline{\text{RD}}$	35	P0.2/AD2
14	XTAL2	36	P0.1/AD1
15	XTAL1	37	P0.0/AD0
16	V_{SS}	38	V_{CC}
17	NC*	39	NC*
18	P2.0/A8	40	P1.0
19	P2.1/A9	41	P1.1
20	P2.2/A10	42	P1.2
21	P2.3/A11	43	P.13
22	P2.4/A12	44	P1.4

* NO INTERNAL CONNECTION

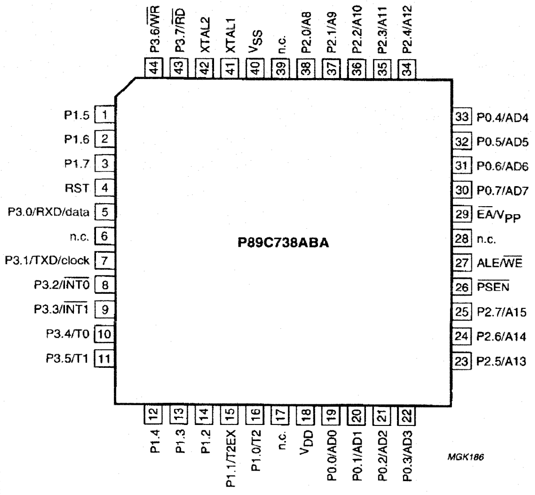

P3.6/WR 44
P3.7/RD 43
XTAL2 42
XTAL1 41
VSS 40
n.c. 39
P2.0/A8 38
P2.1/A9 37
P2.2/A10 36
P2.3/A11 35
P2.4/A12 34
P1.5 1
P1.6 2
P1.7 3
RST 4
P3.0/RXD/data 5
n.c. 6
P3.1/TXD/clock 7
P3.2/INT0 8
P3.3/INT1 9
P3.4/T0 10
P3.5/T1 11
P89C738ABA
33 P0.4/AD4
32 P0.5/AD5
31 P0.6/AD6
30 P0.7/AD7
29 EA/VPP
28 n.c.
27 ALE/WE
26 PSEN
25 P2.7/A15
24 P2.6/A14
23 P2.5/A13
12 P1.4
13 P1.3
14 P1.2
15 P1.1/T2EX
16 P1.0/T2
17 n.c.
18 VDD
19 P0.0/AD0
20 P0.1/AD1
21 P0.2/AD2
22 P0.3/AD3
MGK186

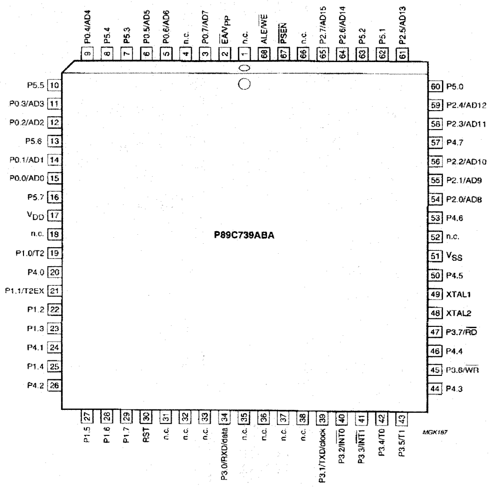

P3.6/WR 44
P3.7/RD 43
XTAL2 42
XTAL1 41
VSS 40
n.c. 39
P2.0/A8 38
P2.1/A9 37
P2.2/A10 36
P2.3/A11 35
P2.4/A12 34
P1.5 1
P1.6 2
P1.7 3
RST 4
P3.0/RXD/data 5
n.c. 6
P3.1/TXD/clock 7
P3.2/INT0 8
P3.3/INT1 9
P3.4/T0 10
P3.5/T1 11
P89C738ABA
33 P0.4/AD4
32 P0.5/AD5
31 P0.6/AD6
30 P0.7/AD7
29 EA/VPP
28 n.c.
27 ALE/WE
26 PSEN
25 P2.7/A15
24 P2.6/A14
23 P2.5/A13
12 P1.4
13 P1.3
14 P1.2
15 P1.1/T2EX
16 P1.0/T2
17 n.c.
18 VDD
19 P0.0/AD0
20 P0.1/AD1
21 P0.2/AD2
22 P0.3/AD3
MGK186

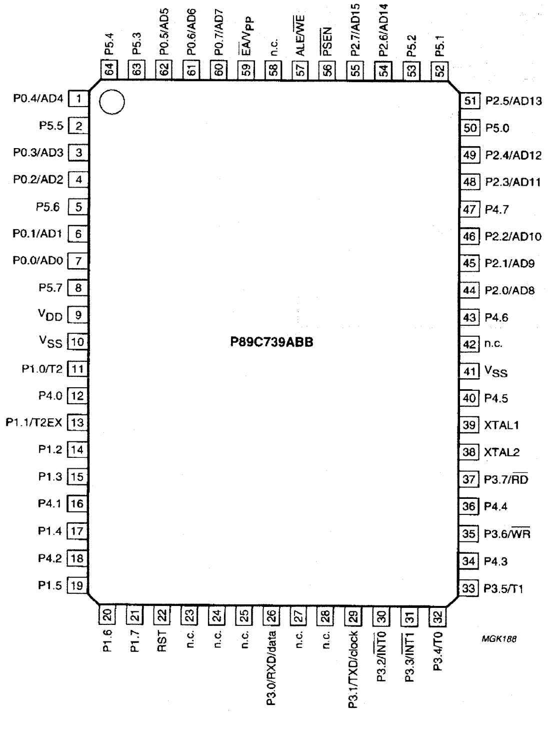

P1.0/T2 1
P1.1/T2EX 2
P1.2 3
P1.3 4
P1.4 5
P1.5 6
P1.6 7
P1.7 8
RST 9
P3.0/RXD/data 10
P3.1/TXD/clock 11
P3.2/INT0 12
P3.3/INT1 13
P3.4/T0 14
P3.5/T1 15
P3.6/WR 16
P3.7/RD 17
XTAL2 18
XTAL1 19
VSS 20
P89C738ABP
40 VDD
39 P0.0/AD0
38 P0.1/AD1
37 P0.2/AD2
36 P0.3/AD3
35 P0.4/AD4
34 P0.5/AD5
33 P0.6/AD6
32 P0.7/AD7
31 EA/VPP
30 ALE/WE
29 PSEN
28 P2.7/A15
27 P2.6/A14
26 P2.5/A13
25 P2.4/A12
24 P2.3/A11
23 P2.2/A10
22 P2.1/A9
21 P2.0/A8
MGK184


P1.4 6
P1.3 5
P1.2 4
P1.1/T2EX 3
P1.0/T2 2
n.c. 1
VDD 44
P0.0/AD0 43
P0.1/AD1 42
P0.2/AD2 41
P0.3/AD3 40
P1.5 7
P1.6 8
P1.7 9
RST 10
P3.0/RXD/data 11
n.c. 12
P3.1/TXD/clock 13
P3.2/INT0 14
P3.3/INT1 15
P3.4/T0 16
P3.5/T1 17
P89C738BBB
39 P0.4/AD4
38 P0.5/AD5
37 P0.6/AD6
36 P0.7/AD7
35 EA/VPP
34 n.c.
33 ALE/WE
32 PSEN
31 P2.7/A15
30 P2.6/A14
29 P2.5/A13
18 P3.6/WR
19 P3.7/RD
20 XTAL2
21 XTAL1
22 VSS
23 n.c.
24 P2.0/A8
25 P2.1/A9
26 P2.2/A10
27 P2.3/A11
28 P2.4/A12
MGK185

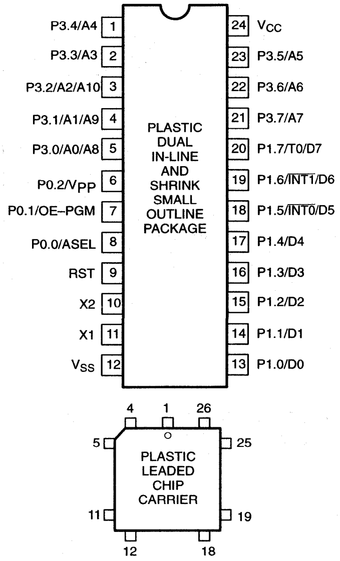
P89C738; P89C739 (Continued)







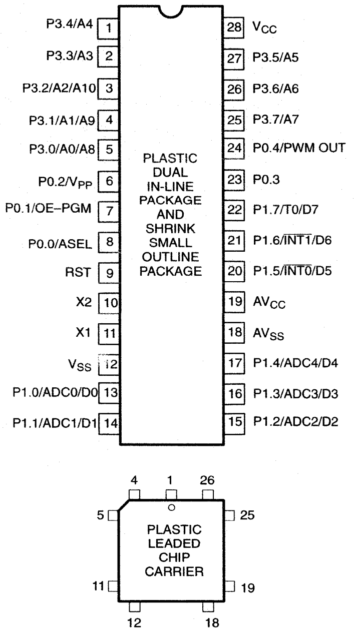
83C748/87C748



Pin	Function	Pin	Function
1	P3.4/A4	15	P1.0/D0
2	P3.3/A3	16	P1.1/D1
3	P3.2/A2/A10	17	P1.2/D2
4	P3.1/A1/A9	18	P1.3/D3
5	NC*	19	P1.4/D4
6	P3.0/A0/A8	20	P1.5/$\overline{\text{INT0}}$/D5
7	P0.2/V_{PP}	21	NC*
8	P0.1/OE-PGM	22	NC*
9	P0.0/ASEL	23	P1.6/$\overline{\text{INT1}}$/D6
10	NC*	24	P1.7/T0/D7
11	RST	25	P3.7/A7
12	X2	26	P3.6/A6
13	X1	27	P3.5/A5
14	V_{SS}	28	V_{CC}

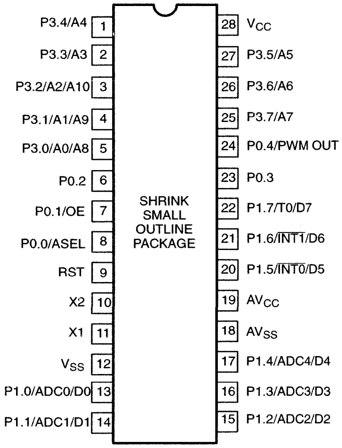
* NO INTERNAL CONNECTION

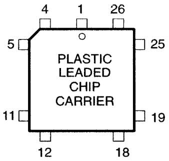
83C749/87C749



Pin	Function	Pin	Function
1	P3.4/A4	15	P1.2/ADC2/D2
2	P3.3/A3	16	P1.3/ADC3/D3
3	P3.2/A2/A10	17	P1.4/ADC4/D4
4	P3.1/A1/A9	18	AV_{SS}
5	P3.0/A0/A8	19	AV_{CC}
6	P0.2/V_{PP}	20	P1.5/$\overline{\text{INT0}}$/D5
7	P0.1/OE-PGM	21	P1.6/$\overline{\text{INT1}}$/D6
8	P0.0/ASEL	22	P1.7/T0/D7
9	RST	23	P0.3
10	X2	24	P0.4/PWM OUT
11	X1	25	P3.7/A7
12	V_{SS}	26	P3.6/A6
13	P1.0/ADC0/D0	27	P3.5/A5
14	P1.1/ADC1/D1	28	V_{CC}

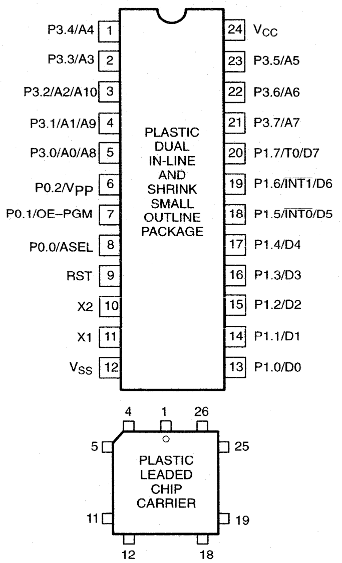
TPM749





Pin	Function	Pin	Function
1	P3.4/A4	15	P1.2/ADC2/D2
2	P3.3/A3	16	P1.3/ADC3/D3
3	P3.2/A2/A10	17	P1.4/ADC4/D4
4	P3.1/A1/A9	18	AV_{SS}
5	P3.0/A0/A8	19	AV_{CC}
6	P0.2	20	P1.5/$\overline{INT0}$/D5
7	P0.1/OE	21	P1.6/$\overline{INT1}$/D6
8	P0.0/ASEL	22	P1.7/T0/D7
9	RST	23	P0.3
10	X2	24	P0.4/PWM OUT
11	X1	25	P3.7/A7
12	V_{SS}	26	P3.6/A6
13	P1.0/ADC0/D0	27	P3.5/A5
14	P1.1/ADC1/D1	28	V_{CC}

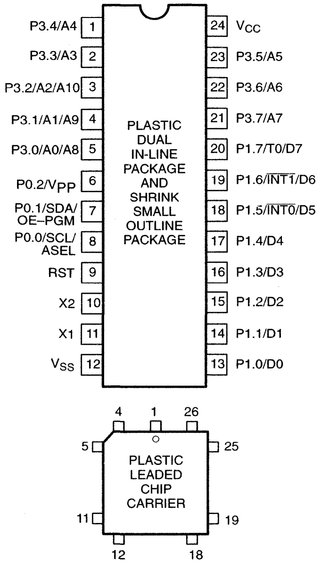
83C750/87C750



Pin	Function	Pin	Function
1	P3.4/A4	15	P1.0/D0
2	P3.3/A3	16	P1.1/D1
3	P3.2/A2/A10	17	P1.2/D2
4	P3.1/A1/A9	18	P1.3/D3
5	NC*	19	P1.4/D4
6	P3.0/A0/A8	20	P1.5/$\overline{\text{INT0}}$/D5
7	P0.2/V_{PP}	21	NC*
8	P0.1/OE-PGM	22	NC*
9	P0.0/ASEL	23	P1.6/$\overline{\text{INT1}}$/D6
10	NC*	24	P1.7/T0/D7
11	RST	25	P3.7/A7
12	X2	26	P3.6/A6
13	X1	27	P3.5/A5
14	V_{SS}	28	V_{CC}

* NO INTERNAL CONNECTION

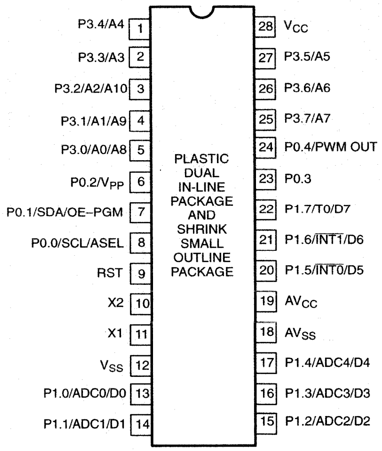
83C751/87C751

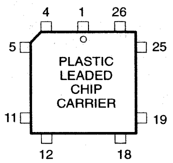


Pin	Function	Pin	Function
1	P3.4/A4	15	P1.0/D0
2	P3.3/A3	16	P1.1/D1
3	P3.2/A2/A10	17	P1.2/D2
4	P3.1/A1/A9	18	P1.3/D3
5	NC*	19	P1.4/D4
6	P3.0/A0/A8	20	P1.5/$\overline{\text{INT0}}$/D5
7	P0.2/V_{PP}	21	NC*
8	P0.1/SDA/OE-PGM	22	NC*
9	P0.0//SCLASEL	23	P1.6/$\overline{\text{INT1}}$/D6
10	NC*	24	P1.7/T0/D7
11	RST	25	P3.7/A7
12	X2	26	P3.6/A6
13	X1	27	P3.5/A5
14	V_{SS}	28	V_{CC}

* NO INTERNAL CONNECTION

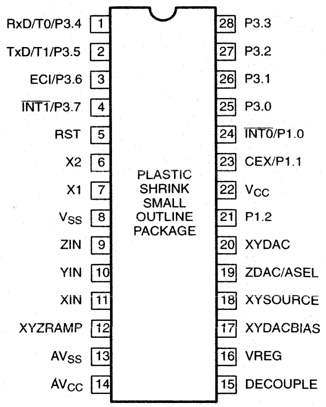
83C751/87C752



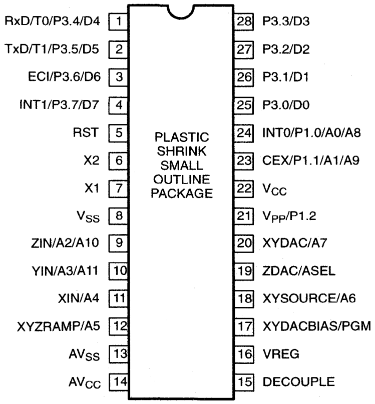


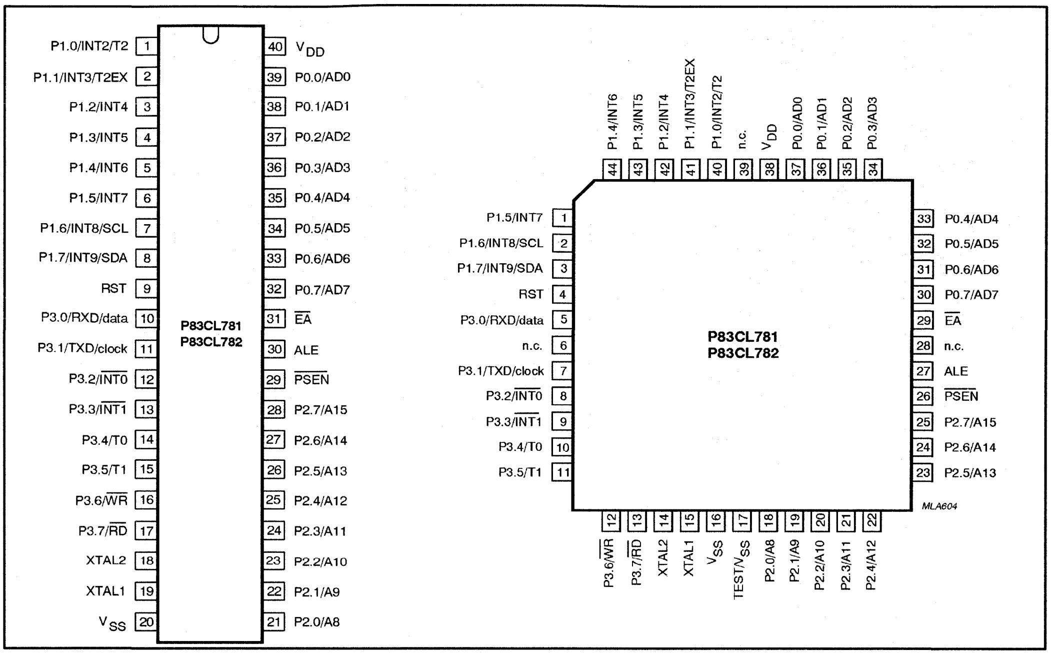
Pin	Function	Pin	Function
1	P3.4/A4	15	P1.2/ADC2/D2
2	P3.3/A3	16	P1.3/ADC3/D3
3	P3.2/A2/A10	17	P1.4/ADC4/D4
4	P3.1/A1/A9	18	AV$_{SS}$
5	P3.0/A0/A8	19	AV$_{CC}$
6	P0.2/V$_{PP}$	20	P1.5/$\overline{\text{INT0}}$/D5
7	P0.1/SDA/OE-PGM	21	P1.6/$\overline{\text{INT1}}$/D6
8	P0.0/SCL/ASEL	22	P1.7/T0/D7
9	RST	23	P0.3
10	X2	24	P0.4/PWM OUT
11	X1	25	P3.7/A7
12	V$_{SS}$	26	P3.6/A6
13	P1.0/ADC0/D0	27	P3.5/A5
14	P1.1/ADC1/D1	28	V$_{CC}$

TPM754

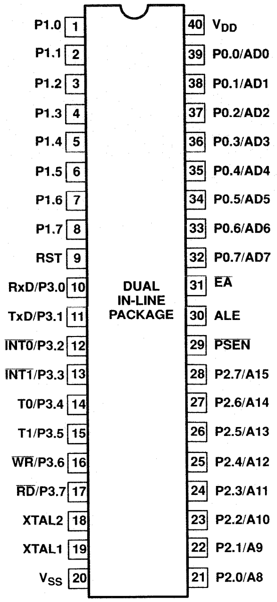


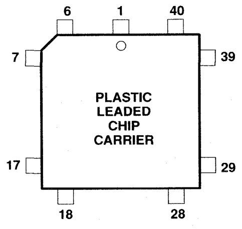
83C754/87C754

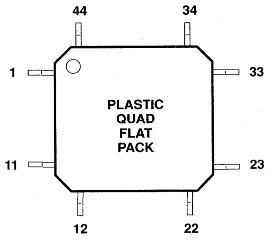


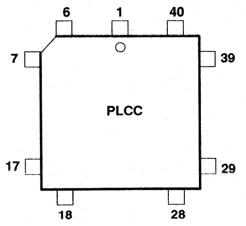

P1.0/INT2/T2 1
P1.1/INT3/T2EX 2
P1.2/INT4 3
P1.3/INT5 4
P1.4/INT6 5
P1.5/INT7 6
P1.6/INT8/SCL 7
P1.7/INT9/SDA 8
RST 9
P3.0/RXD/data 10
P3.1/TXD/clock 11
P3.2/$\overline{INT0}$ 12
P3.3/$\overline{INT1}$ 13
P3.4/T0 14
P3.5/T1 15
P3.6/$\overline{WR}$ 16
P3.7/$\overline{RD}$ 17
XTAL2 18
XTAL1 19
V_{SS} 20
40 V_{DD}
39 P0.0/AD0
38 P0.1/AD1
37 P0.2/AD2
36 P0.3/AD3
35 P0.4/AD4
34 P0.5/AD5
33 P0.6/AD6
32 P0.7/AD7
31 $\overline{EA}$
30 ALE
29 $\overline{PSEN}$
28 P2.7/A15
27 P2.6/A14
26 P2.5/A13
25 P2.4/A12
24 P2.3/A11
23 P2.2/A10
22 P2.1/A9
21 P2.0/A8
P83CL781
P83CL782
P1.4/INT6 44
P1.3/INT5 43
P1.2/INT4 42
P1.1/INT3/T2EX 41
P1.0/INT2/T2 40
n.c. 39
V_{DD} 38
P0.0/AD0 37
P0.1/AD1 36
P0.2/AD2 35
P0.3/AD3 34
P1.5/INT7 1
P1.6/INT8/SCL 2
P1.7/INT9/SDA 3
RST 4
P3.0/RXD/data 5
n.c. 6
P3.1/TXD/clock 7
P3.2/$\overline{INT0}$ 8
P3.3/$\overline{INT1}$ 9
P3.4/T0 10
P3.5/T1 11
33 P0.4/AD4
32 P0.5/AD5
31 P0.6/AD6
30 P0.7/AD7
29 $\overline{EA}$
28 n.c.
27 ALE
26 $\overline{PSEN}$
25 P2.7/A15
24 P2.6/A14
23 P2.5/A13
P83CL781
P83CL782
12 P3.6/$\overline{WR}$
13 P3.7/$\overline{RD}$
14 XTAL2
15 XTAL1
16 V_{SS}
17 TEST/V_{SS}
18 P2.0/A8
19 P2.1/A9
20 P2.2/A10
21 P2.3/A11
22 P2.4/A12
MLA604

80C851/83C851





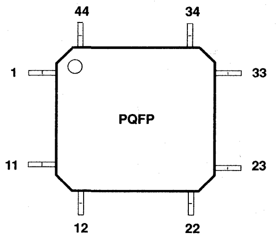




Pin	Function	Pin	Function
1	NC*	23	NC*
2	P1.0	24	P2.0/A8
3	P1.1	25	P2.1/A9
4	P1.2	26	P2.2/A10
5	P1.3	27	P2.3/A11
6	P1.4	28	P2.4/A12
7	P1.5	29	P2.5/A13
8	P1.6	30	P2.6/A14
9	P1.7	31	P2.7/A15
10	RST	32	$\overline{\text{PSEN}}$
11	P3.0/RxD	33	ALE
12	NC*	34	NC*
13	P3.1/TxD	35	$\overline{\text{EA}}$
14	P3.2/$\overline{\text{INT0}}$	36	P0.7/AD7
15	P3.3/$\overline{\text{INT1}}$	37	P0.6/AD6
16	P3.4/T0	38	P0.5/AD5
17	P3.5/T1	39	P0.4/AD4
18	P3.6/$\overline{\text{WR}}$	40	P0.3/AD3
19	P3.7/$\overline{\text{RD}}$	41	P0.2/AD2
20	XTAL2	42	P0.1/AD1
21	XTAL1	43	P0.0/AD0
22	V_{SS}	44	V_{DD}

* NO INTERNAL CONNECTION

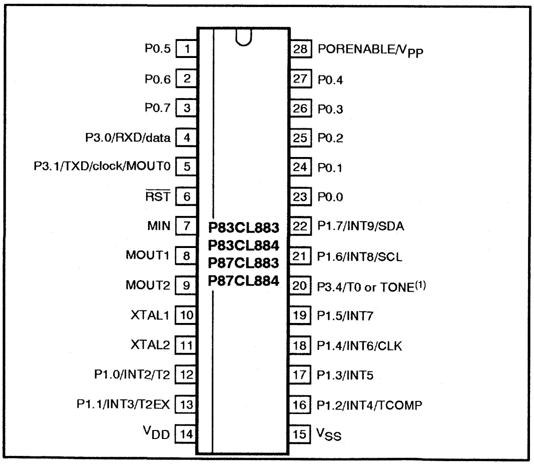
80C851/83C851 (Continued)



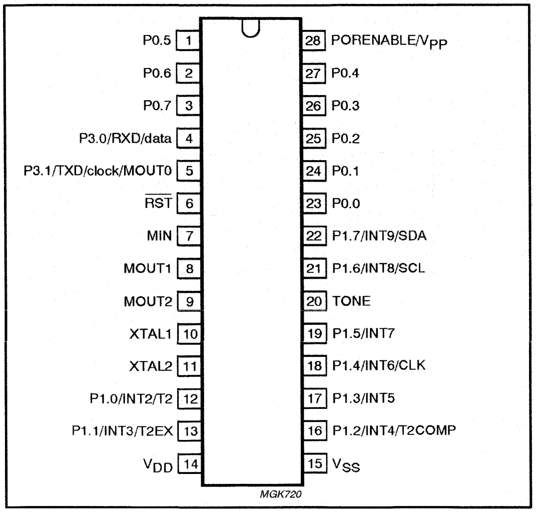
Pin	Function	Pin	Function
1	P1.5	23	P2.5/A13
2	P1.6	24	P2.6/A14
3	P1.7	25	P2.7/A15
4	RST	26	$\overline{\text{PSEN}}$
5	P3.0/RxD	27	ALE
6	NC*	28	NC*
7	P3.1/TxD	29	$\overline{\text{EA}}$
8	P3.2/$\overline{\text{INT0}}$	30	P0.7/AD7
9	P3.3/$\overline{\text{INT1}}$	31	P0.6/AD6
10	P3.4/T0	32	P0.5/AD5
11	P3.5/T1	33	P0.4/AD4
12	P3.6/$\overline{\text{WR}}$	34	P0.3/AD3
13	P3.7$\overline{\text{RD}}$	35	P0.2/AD2
14	XTAL2	36	P0.1/AD1
15	XTAL1	37	P0.0/AD0
16	V_{SS}	38	V_{DD}
17	NC*	39	V_{SS}
18	P2.0/A8	40	P1.0
19	P2.1/A9	41	P1.1
20	P2.2/A10	42	P1.2
21	P2.3/A11	43	P1.3
22	P2.4/A12	44	P1.4

* NO INTERNAL CONNECTION

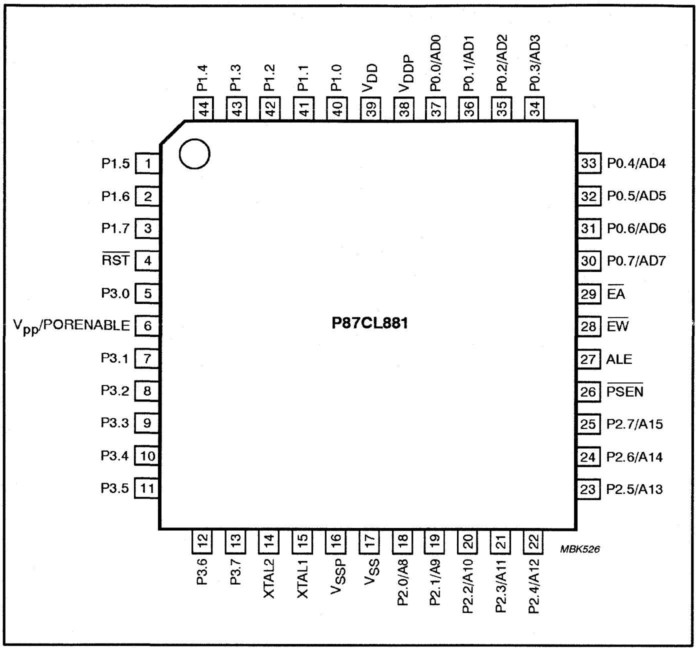
P8CL883_P8L3CL884/P87CL883_P87CL884



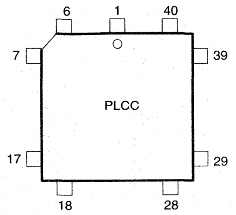
P83CL886_P87CL886, P83CL887_P87CL887



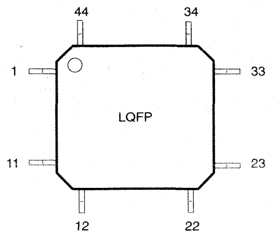
P87CL881


P1.4
P1.3
P1.2
P1.1
P1.0
V_DD
V_DDP
P0.0/AD0
P0.1/AD1
P0.2/AD2
P0.3/AD3
44
43
42
41
40
39
38
37
36
35
34
P1.5 1
P1.6 2
P1.7 3
RST 4
P3.0 5
Vpp/PORENABLE 6
P3.1 7
P3.2 8
P3.3 9
P3.4 10
P3.5 11
33 P0.4/AD4
32 P0.5/AD5
31 P0.6/AD6
30 P0.7/AD7
29 EA
28 EW
27 ALE
26 PSEN
25 P2.7/A15
24 P2.6/A14
23 P2.5/A13
P87CL881
12
13
14
15
16
17
18
19
20
21
22
P3.6
P3.7
XTAL2
XTAL1
V_SSP
V_SS
P2.0/A8
P2.1/A9
P2.2/A10
P2.3/A11
P2.4/A12
MBK526

XA–G1, XA–G2, XA–G3

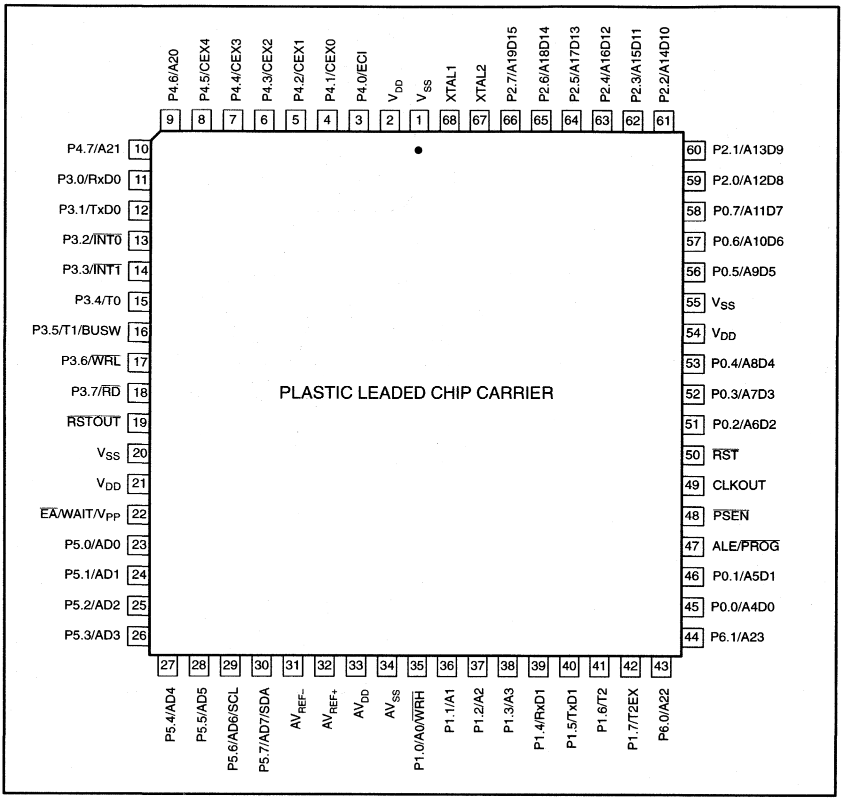


Pin	Function	Pin	Function
1	V_{SS}	23	V_{DD}
2	P1.0/A0/$\overline{\text{WRH}}$	24	P2.0/A12D8
3	P1.1/A1	25	P2.1/A13D9
4	P1.2/A2	26	P2.2/A14D10
5	P1.3/A3	27	P2.3/A15D11
6	P1.4/RxD1	28	P2.4/A16D12
7	P1.5/TxD1	29	P2.5/A17D13
8	P1.6/T2	30	P2.6/A18D14
9	P1.7/T2EX	31	P2.7/A19D15
10	$\overline{\text{RST}}$	32	$\overline{\text{PSEN}}$
11	P3.0/RxD0	33	ALE/$\overline{\text{PROG}}$
12	NC	34	NC
13	P3.1/TxD0	35	$\overline{\text{EA}}$/V_{PP}/WAIT
14	P3.2/$\overline{\text{INT0}}$	36	P0.7/A11D7
15	P3.3/$\overline{\text{INT1}}$	37	P0.6/A10D6
16	P3.4/T0	38	P0.5/A9D5
17	P3.5/T1/BUSW	39	P0.4/A8D4
18	P3.6/$\overline{\text{WRL}}$	40	P0.3/A7D3
19	P3.7/$\overline{\text{RD}}$	41	P0.2/A6D2
20	XTAL2	42	P0.1/A5D1
21	XTAL1	43	P0.0/A4D0
22	V_{SS}	44	V_{DD}

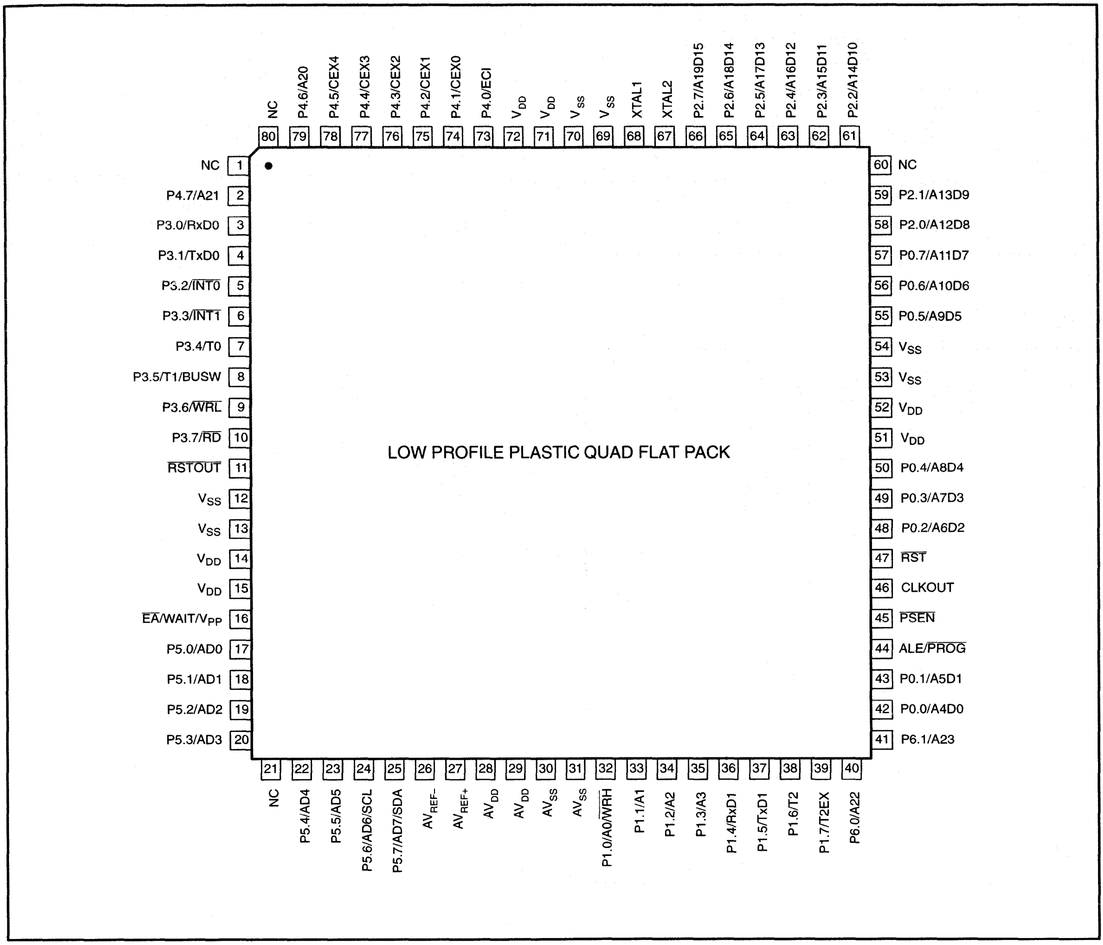


Pin	Function	Pin	Function
1	P1.5/TxD1	23	P2.5/A17D13
2	P1.6/T2	24	P2.6/A18D14
3	P1.7/T2EX	25	P2.7/A19D15
4	$\overline{\text{RST}}$	26	$\overline{\text{PSEN}}$
5	P3.0/RxD0	27	ALE/$\overline{\text{PROG}}$
6	NC	28	NC
7	P3.1/TxD0	29	$\overline{\text{EA}}$/V_{PP}/WAIT
8	P3.2/$\overline{\text{INT0}}$	30	P0.7/A11D7
9	P3.3/$\overline{\text{INT1}}$	31	P0.6/A10D6
10	P3.4/T0	32	P0.5/A9D5
11	P3.5/T1/BUSW	33	P0.4/A8D4
12	P3.6/$\overline{\text{WRL}}$	34	P0.3/A7D3
13	P3.7/$\overline{\text{RD}}$	35	P0.2/A6D2
14	XTAL2	36	P0.1/A5D1
15	XTAL1	37	P0.0/A4D0
16	V_{SS}	38	V_{DD}
17	V_{DD}	39	V_{SS}
18	P2.0/A12D8	40	P1.0/A0/$\overline{\text{WRH}}$
19	P2.1/A13D9	41	P1.1/A1
20	P2.2/A14D10	42	P1.2/A2
21	P2.3/A15D11	43	P1.3/A3
22	P2.4/A16/D12	44	P1.4/RxD1

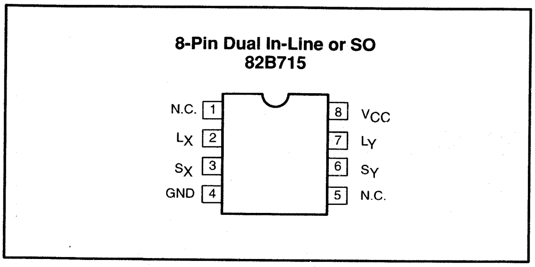
XA–S3


P4.6/A20
P4.5/CEX4
P4.4/CEX3
P4.3/CEX2
P4.2/CEX1
P4.1/CEX0
P4.0/ECI
V_{DD}
V_{SS}
XTAL1
XTAL2
P2.7/A19D15
P2.6/A18D14
P2.5/A17D13
P2.4/A16D12
P2.3/A15D11
P2.2/A14D10
9
8
7
6
5
4
3
2
1
68
67
66
65
64
63
62
61
P4.7/A21 10
P3.0/RxD0 11
P3.1/TxD0 12
P3.2/$\overline{INT0}$ 13
P3.3/$\overline{INT1}$ 14
P3.4/T0 15
P3.5/T1/BUSW 16
P3.6/$\overline{WRL}$ 17
P3.7/$\overline{RD}$ 18
$\overline{RSTOUT}$ 19
V_{SS} 20
V_{DD} 21
$\overline{EA}$/WAIT/V_{PP} 22
P5.0/AD0 23
P5.1/AD1 24
P5.2/AD2 25
P5.3/AD3 26
60 P2.1/A13D9
59 P2.0/A12D8
58 P0.7/A11D7
57 P0.6/A10D6
56 P0.5/A9D5
55 V_{SS}
54 V_{DD}
53 P0.4/A8D4
52 P0.3/A7D3
51 P0.2/A6D2
50 $\overline{RST}$
49 CLKOUT
48 $\overline{PSEN}$
47 ALE/$\overline{PROG}$
46 P0.1/A5D1
45 P0.0/A4D0
44 P6.1/A23
PLASTIC LEADED CHIP CARRIER
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
P5.4/AD4
P5.5/AD5
P5.6/AD6/SCL
P5.7/AD7/SDA
AV_{REF-}
AV_{REF+}
AV_{DD}
AV_{SS}
P1.0/A0/$\overline{WRH}$
P1.1/A1
P1.2/A2
P1.3/A3
P1.4/RxD1
P1.5/TxD1
P1.6/T2
P1.7/T2EX
P6.0/A22

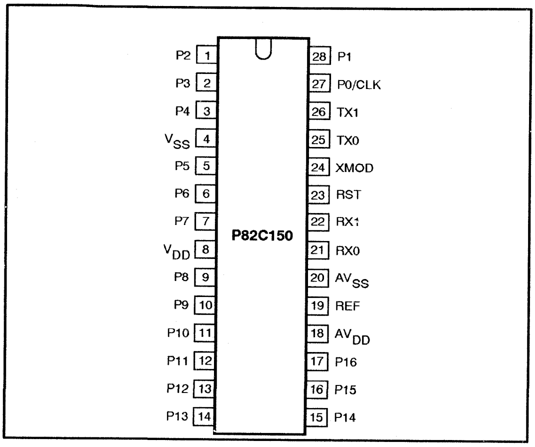
XA–S3 (Continued)



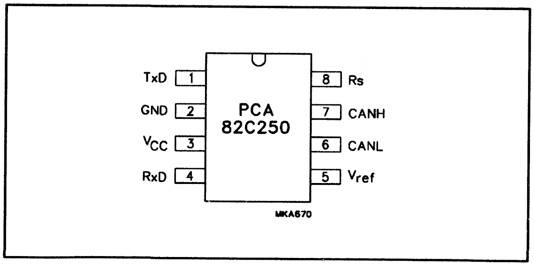
U82B715



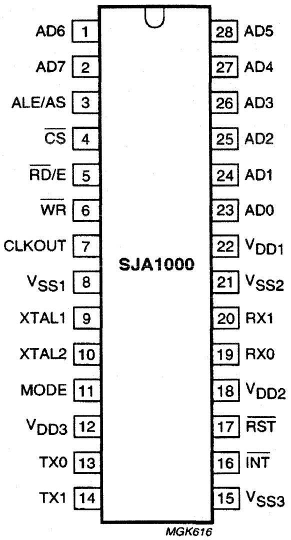
P82C150

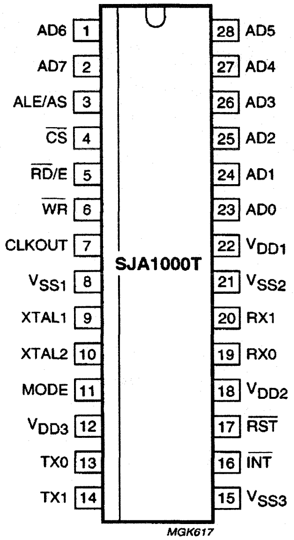


PCA82C250

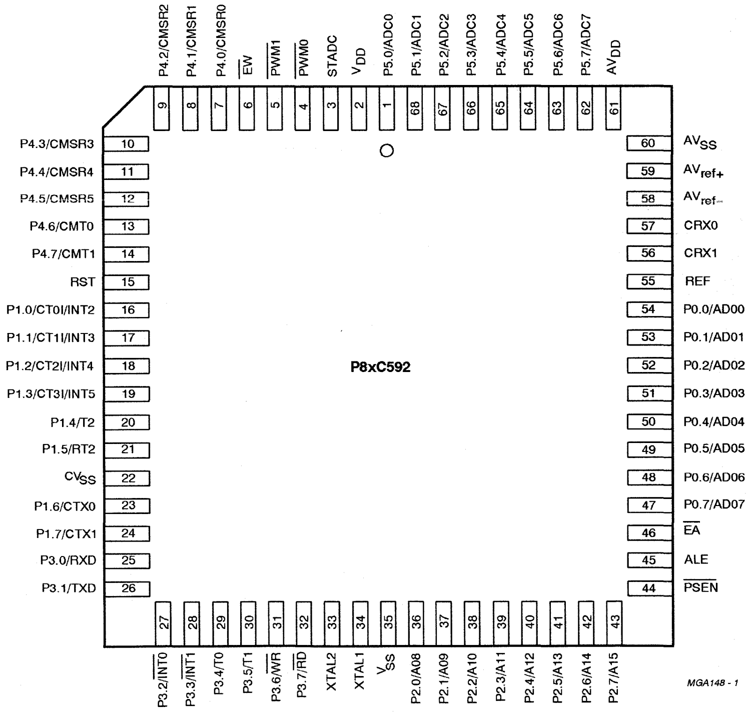


SJA1000

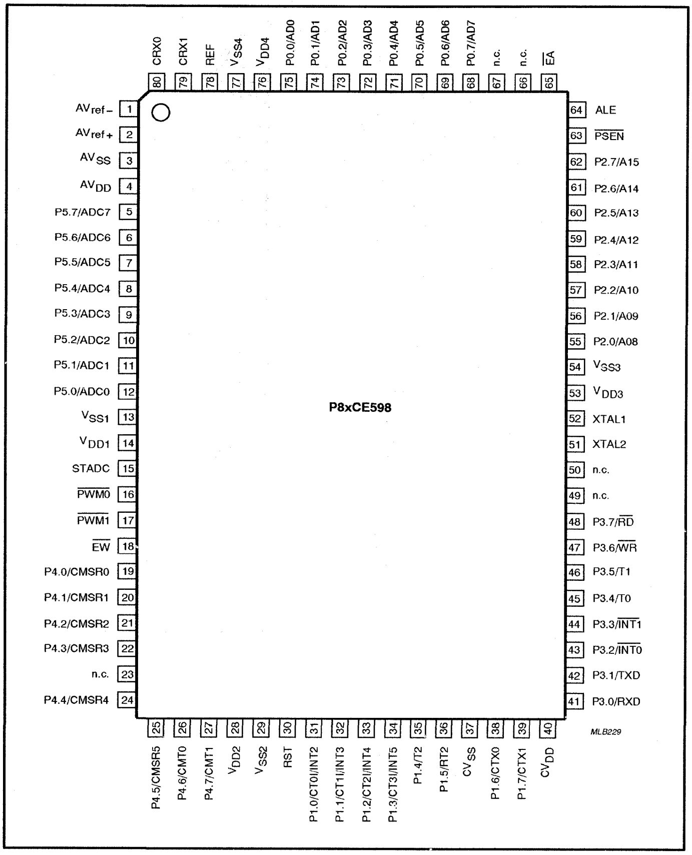




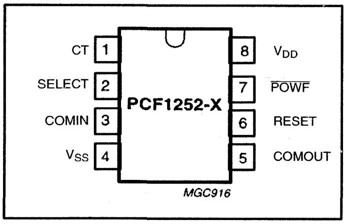
P8XC592



P8XCE598



PCF1252–X FAMILY



North American Sales Offices, Representatives and Distributors

IILIPS :MICONDUCTORS
1 East Arques Avenue
). Box 3409
nnyvale, CA 94088-3409

.ABAMA
ntsville
Philips Semiconductors
Phone: (205) 464-0111
(205) 464-9101

Elcom, Inc.
Phone: (205) 830-4001

ᛁIZONA
ottsdale
Thom Luke Sales, Inc.
Phone: (602) 451-5400

mpe
Philips Semiconductors
Phone: (602) 820-2225

\LIFORNIA
ılabasas
Philips Semiconductors
Phone: (818) 880-6304

Centaur Corporation
Phone: (818) 878-5800

'ine
Philips Semiconductors
Phone: (714) 453-0770

Centaur Corporation
Phone: (714) 261-2123

ıomis
B.A.E. Sales, Inc.
Phone: (916) 652-6777

ın Diego
Philips Semiconductors
Phone: (619) 560-0242

ın Jose
B.A.E. Sales, Inc.
Phone: (408) 452-8133

ınnyvale
Philips Semiconductors
Phone: (408) 991-3737

OLORADO
ıglewood
Philips Semiconductors
Phone: (303) 792-9011

Thom Luke Sales, Inc.
Phone: (303) 649-9717

ONNECTICUT
allingford
JEBCO
Phone: (203) 265-1318

LORIDA
learwater
Conley and Assoc., Inc.
Phone: (813) 572-8895

viedo
Conley and Assoc., Inc.
Phone: (407) 365-3283

EORGIA
orcross
Elcom, Inc.
Phone: (770) 447-8200

ILLINOIS
Itasca
Philips Semiconductors
Phone: (630) 250-0050

Schaumburg
Micro-Tex, Inc.
Phone: (708) 885-8200

INDIANA
Indianapolis
Mohrfield Marketing, Inc.
Phone: (317) 546-6969

Kokomo
Philips Semiconductors
Phone: (765) 459-5355

MARYLAND
Columbia
Third Wave Solutions, Inc.
Phone: (410) 290-5990

MASSACHUSETTS
Chelmsford
JEBCO
Phone: (508) 256-5800

Westford
Philips Semiconductors
Phone: (508) 692-6211

MICHIGAN
Farmington Hills
Philips Semiconductors
Phone: (248) 848-7600

Novi
Mohrfield Marketing, Inc.
Phone: (810) 380-8100

MINNESOTA
Bloomington
High Technology Sales
Phone: (612) 844-9933

MISSOURI
Bridgeton
Centech, Inc.
Phone: (314) 291-4230

Raytown
Centech, Inc.
Phone: (816) 358-8100

NEW JERSEY
Toms River
Philips Semiconductors
Phone: (732) 505-1200
(732) 240-1479

NEW YORK
Ithaca
Bob Dean, Inc.
Phone: (607) 257-0007

Rockville Centre
S-J Associates
Phone: (516) 536-4242

Wappingers Falls
Bob Dean, Inc.
Phone: (914) 297-6406

NORTH CAROLINA
Cary
Philips Semiconductors
Phone: (919) 462-1332

Charlotte
Elcom, Inc.
Phone: (704) 543-1229

Raleigh
Elcom, Inc.
Phone: (919) 743-5200

OHIO
Columbus
Great Lakes Group, Inc.
Phone: (614) 885-6700

Kettering
Great Lakes Group, Inc.
Phone: (513) 298-7322

Solon
Great Lakes Group, Inc.
Phone: (216) 349-2700

OREGON
Beaverton
Philips Semiconductors
Phone: (503) 627-0110

Western Technical Sales
Phone: (503) 644-8860

PENNSYLVANIA
Erie
S-J Associates, Inc.
Phone: (216) 888-7004

Hatboro
Delta Technical Sales, Inc.
Phone: (215) 957-0600

Pittsburgh
S-J Associates, Inc.
Phone: (216) 349-2700

TENNESSEE
Dandridge
Philips Semiconductors
Phone: (615) 397-5053

TEXAS
Austin
OM Associates
Phone: (512) 794-9971

Houston
Philips Semiconductors
Phone: (281) 999-1316

OM Associates
Phone: (713) 376-6400

Richardson
Philips Semiconductors
Phone: (972) 644-1610
(972) 705-9555

OM Associates
Phone: (972) 690-6746

UTAH
Salt Lake City
Electrodyne
Phone: (801) 264-8050

WASHINGTON
Bellevue
Western Technical Sales
Phone: (425) 641-3900

Spokane
Western Technical Sales
Phone: (509) 922-7600

WISCONSIN
Waukesha
Micro-Tex, Inc.
Phone: (414) 542-5352

CANADA PHILIPS SEMICONDUCTORS CANADA, LTD.
Calgary, Alberta
Philips Semiconductors/
Components, Inc.
Phone: (403) 735-6233

Tech-Trek, Ltd.
Phone: (403) 241-1719

Kanata, Ontario
Philips Semiconductors
Phone: (613) 599-8720

Tech-Trek, Ltd.
Phone: (613) 599-8787

Montreal, Quebec
Philips Semiconductors/
Components, Inc.
Phone: (514) 956-2134

Mississauga, Ontario
Tech-Trek, Ltd.
Phone: (416) 238-0366

Richmond, B.C.
Tech-Trek, Ltd.
Phone: (604) 276-8735

Scarborough, Ontario
Philips Semiconductors/
Components, Ltd.
(416) 292-5161

Ville St. Laurent, Quebec
Tech-Trek, Ltd.
Phone: (514) 337-7540

MEXICO
Anzures Section
Philips Components
Phone: +9–5 (800) 234-7381

El Paso, TX
Philips Components
Phone: (915) 775-4020

PUERTO RICO
Caguas
Mectron Group
Phone: (809) 746-3522

DISTRIBUTORS
Contact one of our local distributors:
Allied Electronics
Anthem Electronics
Arrow/Schweber Electronics
Future Electronics
Hamilton Hallmark
Marshall Industries
Newark Electronics
Penstock
Richardson Electronics
Zeus Electronics

Philips Semiconductors – a worldwide company

Argentina: see South America

Australia: 34 Waterloo Road, NORTH RYDE, NSW 2113,
Tel. +61 2 9805 4455, Fax. +61 2 9805 4466

Austria: Computerstr. 6, A-1101 WIEN, P.O. Box 213, Tel. +43 160 1010,
Fax. +43 160 101 1210

Belarus: Hotel Minsk Business Center, Bld. 3, r. 1211, Volodarski Str. 6,
220050 MINSK, Tel. +375 172 200 733, Fax. +375 172 200 773

Belgium: see The Netherlands

Brazil: see South America

Bulgaria: Philips Bulgaria Ltd., Energoproject, 15th floor,
51 James Bourchier Blvd., 1407 SOFIA,
Tel. +359 2 689 211, Fax. +359 2 689 102

Canada: PHILIPS SEMICONDUCTORS/COMPONENTS, Tel. +1 800 234 7381

China/Hong Kong: 501 Hong Kong Industrial Technology Centre,
72 Tat Chee Avenue, Kowloon Tong, HONG KONG,
Tel. +852 2319 7888, Fax. +852 2319 7700

Colombia: see South America

Czech Republic: see Austria

Denmark: Prags Boulevard 80, PB 1919, DK-2300 COPENHAGEN S,
Tel. +45 32 88 2636, Fax. +45 31 57 0044

Finland: Sinikalliontie 3, FIN-02630 ESPOO,
Tel. +358 9 615800, Fax. +358 9 61580920

France: 51 Rue Carnot, BP317, 92156 SURESNES Cedex,
Tel. +33 1 40 99 6161, Fax. +33 1 40 99 6427

Germany: Hammerbrookstraße 69, D-20097 HAMBURG,
Tel. +49 40 23 53 60, Fax. +49 40 23 536 300

Greece: No. 15, 25th March Street, GR 17778 TAVROS/ATHENS,
Tel. +30 1 4894 339/239, Fax. +30 1 4814 240

Hungary: see Austria

India: Philips INDIA Ltd, Band Box Building, 2nd floor,
254-D, Dr. Annie Besant Road, Worli, MUMBAI 400 025,
Tel. +91 22 493 8541, Fax. +91 22 493 0966

Indonesia: see Singapore

Ireland: Newstead, Clonskeagh, DUBLIN 14,
Tel. +353 1 7640 000, Fax. +353 1 7640 200

Israel: RAPAC Electronics, 7 Kehilat Saloniki St, PO Box 18053,
TEL AVIV 61180, Tel. +972 3 645 0444, Fax. +972 3 649 1007

Italy: PHILIPS SEMICONDUCTORS, Piazza IV Novembre 3, 20124 MILANO,
Tel. +39 2 6752 2531, Fax. +39 2 6752 2557

Japan: Philips Bldg 13-37, Kohnan 2-chome, Minato-ku, TOKYO 108,
Tel. +81 3 3740 5130, Fax. +81 3 3740 5077

Korea: Philips House, 260-199 Itaewon-dong, Yongsan-ku, SEOUL,
Tel. +82 2 709 1412, Fax. +82 2 709 1415

Malaysia: No. 76 Jalan Universiti, 46200 PETALING JAYA, SELANGOR,
Tel. +60 3 750 5214, Fax. +60 3 757 4880

Mexico: 5900 Gateway East, Suite 200, EL PASO, TEXAS 79905,
Tel. +9-5 800 234 7381

Middle East: see Italy

Netherlands: Postbus 90050, 5600 PB EINDHOVEN, Bldg. VB,
Tel. +31 40 27 82785, Fax. +31 40 27 88399

New Zealand: 2 Wagener Place, C.P.O. Box 1041, AUCKLAND,
Tel. +64 9 849 4160, Fax. +64 9 849 7811

Norway: Box 1, Manglerud 0612, OSLO,
Tel. +47 22 74 8000, Fax. +47 22 74 8341

Philippines: Philips Semiconductors Philippines Inc.,
106 Valero St. Salcedo Village, P.O. Box 2108 MCC, MAKATI, Metro MANILA,
Tel. +63 2 816 6380, Fax. +63 2 817 3474

Poland: Ul. Lukiska 10, PL 04-123 WARSZAWA,
Tel. +48 22 612 2831, Fax. +48 22 612 2327

Portugal: see Spain

Romania: see Italy

Russia: Philips Russia, Ul. Usatcheva 35A, 119048 MOSCOW,
Tel. +7 095 755 6918, Fax. +7 095 755 6919

Singapore: Lorong 1, Toa Payoh, SINGAPORE 1231,
Tel. +65 350 2538, Fax. +65 251 6500

Slovakia: see Austria

Slovenia: see Italy

South Africa: S.A. PHILIPS Pty Ltd., 195-215 Main Road Martindale,
2092 JOHANNESBURG, P.O. Box 7430 Johannesburg 2000,
Tel. +27 11 470 5911, Fax. +27 11 470 5494

South America: Al. Vicente Pinzon, 173, 6th floor, 04547-130 SÃO PAULO, SP,
Brazil, Tel. +55 11 821 2333, Fax. +55 11 821 2382

Spain: Balmes 22, 08007 BARCELONA,
Tel. +34 3 301 6312, Fax. +34 3 301 4107

Sweden: Kottbygatan 7, Akalla, S-16485 STOCKHOLM,
Tel. +46 8 632 2000, Fax. +46 8 632 2745

Switzerland: Allmendstrasse 140, CH-8027 ZÜRICH,
Tel. +41 1 488 2686, Fax. +41 1 488 3263

Taiwan: Philips Semiconductors, 6F, No. 96, Chien Kuo N. Rd., Sec. 1, TAIPEI,
Taiwan Tel. +886 2 2134 2865, Fax. +886 2 2134 2874

Thailand: PHILIPS ELECTRONICS (THAILAND) Ltd.,
209/2 Sanpavuth-Bangna Road Prakanong, BANGKOK 10260,
Tel. +66 2 745 4090, Fax. +66 2 398 0793

Turkey: Talatpasa Cad. No. 5, 80640 GÜLTEPE/ISTANBUL,
Tel. +90 212 279 2770, Fax. +90 212 282 6707

Ukraine: PHILIPS UKRAINE, 4 Patrice Lumumba str., Building B, Floor 7,
252042 KIEV, Tel. +380 44 264 2776, Fax. +380 44 268 0461

United Kingdom: Philips Semiconductors Ltd., 276 Bath Road, Hayes,
MIDDLESEX UB3 5BX, Tel. +44 181 730 5000, Fax. +44 181 754 8421

United States: 811 East Arques Avenue, SUNNYVALE, CA 94088-3409,
Tel. +1 800 234 7381

Uruguay: see South America

Vietnam: see Singapore

Yugoslavia: PHILIPS, Trg N. Pasica 5/v, 11000 BEOGRAD,
Tel. +381 11 625 344, Fax.+381 11 635 777

For all other countries apply to: Philips Semiconductors, International Marketing & Sales
Communications, Building BE-p, P.O. Box 218, 5600 MD EINDHOVEN, The Netherlands, Fax. +31 40 27 24825

Internet: http://www.semiconductors.philips.com

SCH57

The information presented in this document does not form part of any quotation or contract, is believed to be accurate and reliable and may be changed without notice. No liability will be accepted by the publisher for any consequence of its use. Publication thereof does not convey nor imply any license under patent- or other industrial or intellectual property rights.

Printed in USA

453801/51.3M/06/pp382

Date of release: January 1998

Document order number: 9397 750 03105

PHILIP

Let's make things bett