# 8048-based 8-bit Microcontrollers

# DATA HANDBOOK

Philips Semiconductors

PHILIPS

BOOK IC14 1994

## 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.

# 8048-based 8-bit Microcontrollers

## CONTENTS

# PREFACE

**Microcontrollers from Philips Semiconductors**

Philips Semiconductors 8 and 16-bit microcontrollers are based on the widely-accepted 8048, 8051 and 68000 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"*.
- 68000 compatible 'industry standard' architecture types in *"Data Handbook IC21"*.

The Low Power 80CL51 family of derivatives can be found in *"Data Handbook IC20"*. These devices operate over the wide voltage range of 1.8 to 6.0 V 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. We also offer the Philips/Digital Equipment Corp, ACCESS.bus, a new Standard Desktop bus. And 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 68000 architecture. While these are called 16-bit microcontrollers, the 68000 CPU core architecture is a 32-bit. This offers the user a great deal more processing power when the need arises in a design to move from an 8-bit to a 16-bit microcontroller.

Philips Semiconductors 16-bit microcontrollers are software compatible with existing 68000 code. Future developments include the introduction of SPARC and Trimedia devices.

Philips Semiconductors is developing a family of 16-bit microcontrollers based on the 8051 'XA' (eXtended Architecture). This family of microcontrollers will offer advanced performance for those applications that are computation and memory intensive in an embedded control environment.

## DEFINITIONS

| **Data sheet status** | |
|---|---|
| Objective specification | This data sheet contains target or goal specifications for product development. |
| Preliminary specification | This data sheet contains preliminary data; supplementary data may be published later. |
| Product specification | This data sheet contains final product specifications. |
| **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** | |
| Where application information is given, it is advisory and does not form part of the specification. | |

## 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 customers using or selling these products for use in such applications do so at their own risk and agree to fully indemnify Philips for any damages resulting from such improper use or sale.

## 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 Q1 and IBM MDQ, and the CECC system of conformity. Our factories are certified to ISO 9000 and CECC by external inspectorates.

### Partnerships with customers

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

### 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 OEM (original equipment manufacturer) equipment, component reliability must be extremely high. Our research laboratories and development departments study the failure mechanisms of semiconductors. Their studies have resulted in design rules and process optimization for the highest built-in product reliability. Highly accelerated tests are applied to the products 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.

## LIMITING VALUES (RATINGS)

The rating systems recommended by the International Electrotechnical Commission (IEC) in publication IEC 134 are reproduced below. The limiting values (or ratings) given in the data sheets are defined in accordance with the Absolute Maximum Rating System.

### Definitions of terms used

ELECTRONIC DEVICE

An electronic tube or valve, transistor or other semiconductor device. This definition excludes inductors, capacitors, resistors and similar components.

CHARACTERISTIC

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

BOGEY ELECTRONIC DEVICE

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

RATING

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

RATING SYSTEM

The set of principles upon which ratings are established and which determine their interpretation. The rating system indicates the division of responsibility between the device manufacturer and the circuit designer, with the object of ensuring that the working conditions do not exceed the ratings.

### Absolute maximum rating system

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

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

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

### Design maximum rating system

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

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

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

**Design centre rating system**

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

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

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

## ELECTROSTATIC CHARGES

Electrostatic charges can exist in many things; for example, man-made-fibre clothing, moving machinery, objects with air blowing across them, plastic storage bins, sheets of paper stored in plastic envelopes, paper from electrostatic copying machines, and people. The charges are caused by friction between two surfaces, at least one of which is non-conductive. The magnitude and polarity of the charges depend on the different affinities for electrons of the two materials rubbing together, the friction force and the humidity of the surrounding air.

Electrostatic discharge is the transfer of an electrostatic charge between bodies at different potentials and occurs with direct contact or when induced by an electrostatic field. All of our MOS devices are internally protected against electrostatic discharge but they **can** be damaged if the following precautions are not taken.

## WORK STATION

Figure 1 shows a working area suitable for safely handling electrostatic sensitive devices. It has a work bench, the surface of which is conductive or covered by an antistatic sheet. Typical resistivity for the bench surface is between 1 and 500 kΩ per $cm^2$. The floor should also be covered with antistatic material. The following precautions should be observed:

- Persons at a work bench should be earthed via a wrist strap and a resistor
- All mains-powered electrical equipment should be connected via an earth leakage switch
- Equipment cases should be earthed
- Relative humidity should be maintained between 50 and 65%
- An ionizer should be used to neutralize objects with immobile static charges.

## RECEIPT AND STORAGE

MOS devices are packed for dispatch in antistatic/conductive containers, usually boxes, tubes or blister tape. The fact that the contents are sensitive to electrostatic discharge is shown by warning labels on both primary and secondary packaging.

The devices should be kept in their original packaging whilst in storage. If a bulk container is partially unpacked, the unpacking should be performed at a protected work station. Any MOS devices that are stored temporarily should be packed in conductive or antistatic packaging or carriers.

## ASSEMBLY

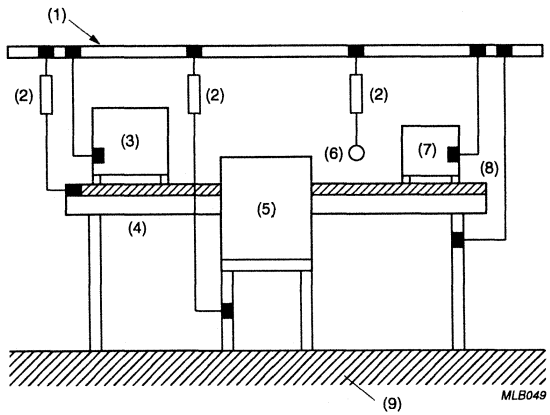
MOS devices must be removed from their protective packaging with earthed component pincers or short-circuit clips. Short-circuit clips must remain in place during mounting, soldering and cleansing/drying processes. Do not remove more devices from the storage packaging than are needed at any one time. Production/assembly documents should state that the product contains electrostatic sensitive devices and that special precautions need to be taken.

During assembly, ensure that the MOS devices are the last of the components to be mounted and that this is done at a protected work station.

All tools used during assembly, including soldering tools and solder baths, must be earthed. All hand tools should be of conductive or antistatic material and, where possible, should not be insulated.

Measuring and testing of completed circuit boards must be done at a protected work station. Place the soldered side of the circuit board on conductive or antistatic foam and remove the short-circuit clips. Remove the circuit board from the foam, holding the board only at the edges. Make sure the circuit board does not touch the conductive surface of the work bench. After testing, replace the circuit board on the conductive foam to await packaging.

Assembled circuit boards containing MOS devices should be handled in the same way as unmounted MOS devices. They should also carry warning labels and be packed in conductive or antistatic packaging.




(1) Earthing rail.
(2) Resistor (500 kΩ ±10%, 0.5 W).
(3) Ionizer.
(4) Work bench.
(5) Chair.
(6) Wrist strap.
(7) Electrical equipment.
(8) Conductive surface/antistatic sheet.
(9) Antistatic floor covering.

Fig.1 Protected work station.

# SECTION 1 INTRODUCTION

## INTRODUCTION

IC14 is one of three books detailing the Philips product line of microcontrollers and microprocessors. This book deals only with the 8-bit microcontrollers that use the 8048 instruction set and derivative microcontrollers based on the 8048 architecture and instruction set. The other two books are pocket books ideal as a reference:

- **"Microcontrollers, Microprocessors and Support overview"** (12NC 9398 706 37011)
- **"Products & Support for Tele/Datacom Terminals"** (12NC 9398 706 36011).

### Single-chip 8-bit microcontrollers with 8048 based instruction set

The microcontrollers presented in this book are based on the 8048 industry standard 8-bit microcontroller. This section provides full technical details on the 8048 and derivatives.

The NMOS 84XX family of derivatives are optimized for cost-effective applications. The I/O and interrupt capabilities have been enhanced, and the pin count has been reduced in comparison with the 8048. This family is detailed in Section 2.

Section 3, deals with the CMOS 84CXXX family of derivatives. The 84CXXX microcontroller family is essentially a CMOS enhancement of the NMOS 84XX family and contains a wide variety of more application specific derivatives; most of the derivatives feature a supply voltage range of 2.5 to 5.5 V. These devices are ideally suited for low-power, low-voltage applications, especially in the telecommunication and consumer areas.

The CMOS 33XX family is detailed in Section 4. These devices are optimized for telephony applications. Several members of the 33XX family feature a supply voltage range of 1.8 to 6.0 V The wide supply voltage range plus on-chip application- specific hardware, make the 84XX, 84CXXX and 33XX microcontroller families unique in the microcontroller market.

The packaging, soldering and mechanical data are detailed in Section 5 of this handbook.

### Type numbering

Figure 1 gives detailed examples of the type numbering system used throughout this book.

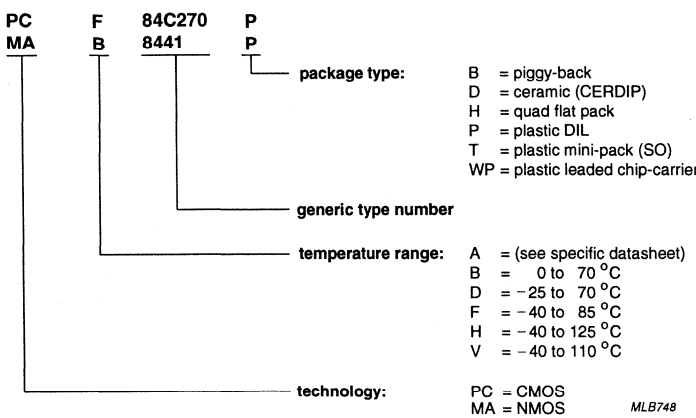



Fig.1 Type numbering system.

**A range of microcontrollers with enhanced I²C-bus serial bus interface**

To reduce interconnect complexity, the two wire I²C serial bus interface hardware has been incorporated in many derivatives (see Fig.2). More than 100 microcontrollers and other devices support the I²C-bus.

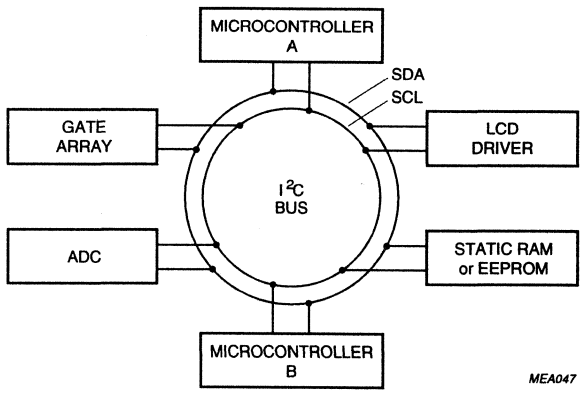



Fig.2 I²C-bus overview.

**PURCHASE OF PHILIPS I²C COMPONENTS**

Purchase of Philips I²C components conveys a license under the Philips' I²C patent to use the components in the I²C system provided the system conforms to the I²C specification defined by Philips. This specification can be ordered using the code 9398 393 40011.

# MAB84XX
# Single-chip 8-bit microcontroller family specification

**CONTENTS**

Section

# Single-chip 8-bit microcontroller family specification

# MAB84XX

## 1 INTRODUCTION

**This family data sheet describes the microcontroller core which is common for all members of the MAB84XX family. For complete information of a particular microcontroller, consult both the specific microcontroller data sheet and this family data sheet.**

## 2 FEATURES

- 8-bit CPU
- Up to 8 K bytes ROM
- Up to 128 bytes RAM
- Over 80 instructions, all instructions 1 or 2 cycles
- Up to 20 quasi bi-directional I/O port lines
- 8-bit programmable timer/event counter
- 3 single-level vectored interrupts (external, timer/counter and serial I/O)
- 2 Test inputs: T0 (may also be used as an interrupt) and T1 (may also be used as an input to an 8-bit counter)
- Serial I/O interface
- On-chip oscillator

The family is well supported with:

- Cross assemblers
- In-circuit emulation tools
- Window debugger
- Piggy-back versions for prototyping.

## 3 GENERAL DESCRIPTION

The MAB84XX single-chip 8-bit microcontroller family is fabricated in NMOS. Members contain on-chip mask programmable program ROM (up to 6 K bytes, except the ROM-less version which can address up to 8 K bytes of external ROM), up to 128 bytes RAM, an interrupt input, a test input (directly testable), a serial I/O and general purpose I/O lines. The instruction set is based on that of the MAB8048. A number of the MAB84XX family members have similar CMOS counterparts in the PCF84CXXX microcontroller family.

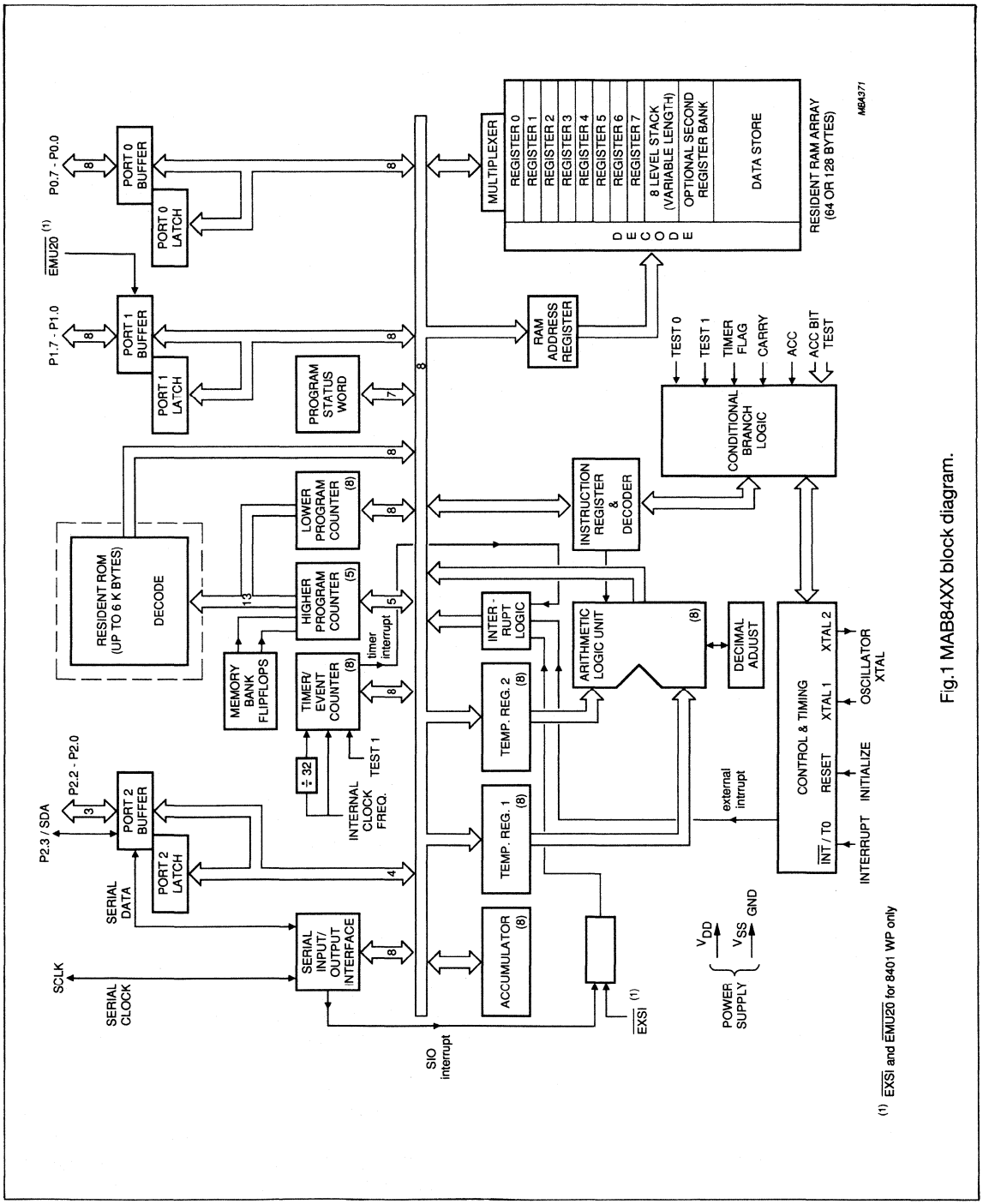



Fig.1 MAB84XX block diagram.

(1) EXSI and EMU20 for 8401 WP only

## 4 CENTRAL PROCESSING UNIT (CPU)

The CPU performs arithmetic, logical and program control functions. The CPU consists of:

- **ALU** (arithmetic and logic unit) which performs the following operations:
  arithmetic operations: ADD, INCREMENT, DECREMENT, ROTATE. The DA A, SWAP A, and XCHD instructions and a half carry flag simplify BCD arithmetic and the handling of nibbles.
  logical operations: OR, AND, EXCLUSIVE-OR, and COMPLEMENT.
- **ACCUMULATOR**; in most operations this is the source or destination register.
- **Program status word**; indicates the status of the microcontroller.
- **Program counter**; supplies a 13-bit address to the program memory.
- **Instruction decoder**; decodes the instructions and supplies control signals to several parts of the microcontroller.
- **Control and timing**; contains the control, oscillator and timing circuitry. A machine cycle consists of 10 states, each state contains 3 XTAL clock periods. Thus, one machine cycle consists of 30 XTAL clock periods. All instructions take 1 or 2 machine cycles.

## 5 MEMORY AND REGISTERS

### 5.1 Program memory

Members of the MAB84XX family contain 1 K, 2 K, 4 K, or 6 K bytes of on-chip read-only memory (ROM); there are also ROM-less versions which can address up to 8 K bytes of external ROM. Each location is directly addressable by the program counter. The program memory is mask-programmed at the factory. Figure 2a shows the program memory map.

Four program memory locations are of special importance:

- Location 0: first instruction to be executed after the controller is reset.
- Location 3: first instruction of an external interrupt ($\overline{INT}$/T0) service routine.
- Location 5: first instruction of a serial I/O interrupt service routine.
- Location 7: first instruction of a timer/event counter interrupt service routine.

Program memory is arranged in banks of 2 K bytes. Memory banks are preselected by the SEL MB instructions. Memory bank boundaries can only be crossed by using the unconditional jump (JMP) or subroutine call (CALL) instructions, after the appropriate memory bank has been selected. The program memory is further divided into 'pages', each of 256 bytes. Page boundaries can not be crossed by conditional jumps and indirect jump (JMPP) instructions.

The MOVP A,@A instruction permits efficient table look-up from the current ROM page.

### 5.2 Data memory

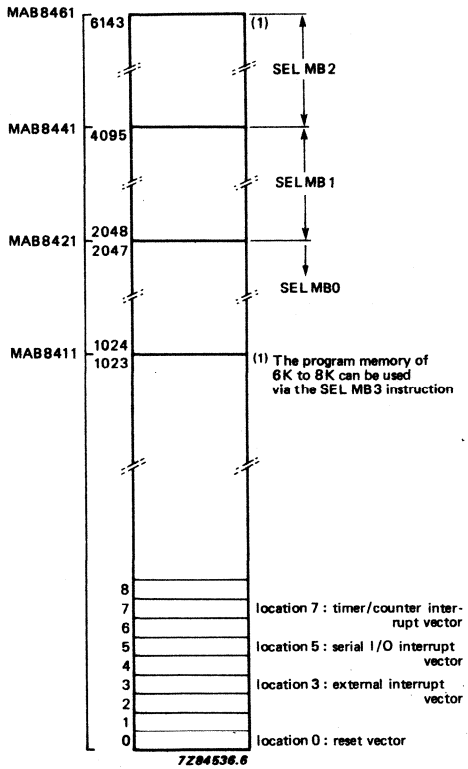
Members of the MAB84XX family contain up to 128 bytes of random access data memory (RAM). All locations are indirectly addressable using RAM pointer registers; up to 16 designated locations are directly addressable (register banks 0 and 1). Another 16 bytes are designated to an 8-level program counter stack addressed by a 3-bit stack pointer. Figure 2b shows the data memory map.

#### *5.2.1 Registers R0 to R7*

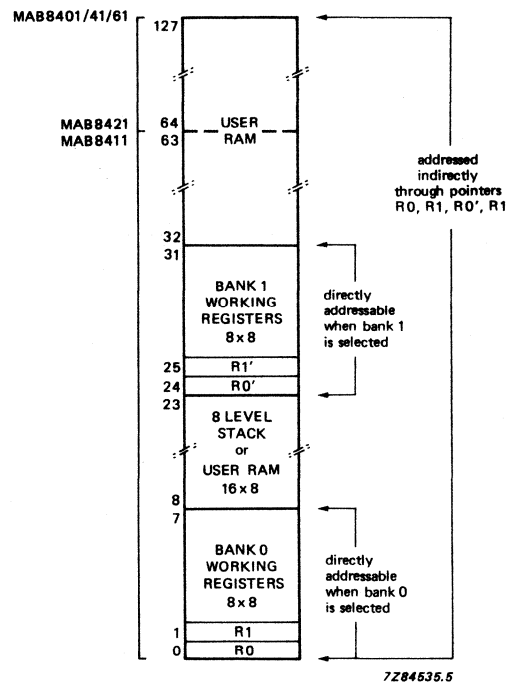
Registers R0 to R7 are directly addressable by the direct register instructions. Ease of addressing, and a minimum requirement of instruction bytes to manipulate their contents, makes these locations suitable for storing frequently addressed intermediate results.

Executing the SEL RB0 (SELect Register Bank 0) instruction designates R0-R7 to data memory locations 0-7. Executing the SEL RB1 instruction designates R0-R7 to data memory locations 24-31. This second register bank may be used as an extension of the first, or it may be reserved for use during interrupt service routines leaving the first bank available for the main program.

The first 2 locations of each bank contain the RAM pointer registers R0, R1 and R0', R1' which indirectly address all data memory locations. Every data memory location can operate as a loop counter when used with the decrement and test instruction DJNZ @Rr,addr.




(a) Program memory map




(b) Data memory map

Fig.2 MAB84XX memory and register map.

*5.2.2 Stack*

Locations 8 to 23 of the data memory are designated to an 8-level program counter stack (2 locations per level). See Fig.3. A 3-bit stack pointer (SP) points to the next free location on the stack. After a RESET, the SP contents are 000 and the SP points to data memory locations 8 and 9.

During a subroutine CALL or interrupt, the contents of the 13 bit program counter (PC) and bits 4, 6 and 7 of the program status word (PSW) are first transferred to the stack. The SP is then incremented. During a RET or RETR instruction, the 13 bit program counter is restored and the stack pointer is decremented. Only the RETR instruction transfers the saved PSW bits to the PSW.

Nesting of subroutines and interrupt service routines can continue to a depth of 8 levels without overflowing the stack. When an overflow occurs the deepest address stored (locations 8 and 9) will be overwritten and lost since the stack pointer overflows from 111 to 000. The stack pointer also underflows from 000 to 111.

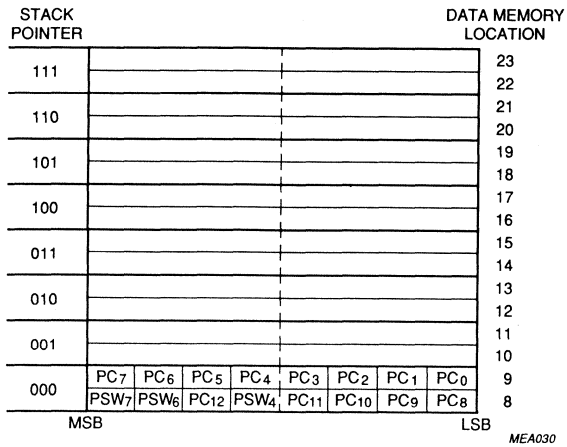



Fig.3 Program counter stack.

*5.2.3 User RAM*

Data memory locations 32 to 127 are designated as user RAM and are indirectly addressable with the RAM pointers R0, R1 or R0', R1'. Unused register and stack locations are also available as user RAM and are addressed in the same way.

## 6 PROGRAM COUNTER

The 13-bit program counter can address up to 8 K bytes of ROM (see Fig.4). The least significant 11 bits (PC0 to PC10) are auto-incrementing. The two most significant bits (PC11 and PC12) are loaded with the contents of two internal flip flops (MBFF0 and MBFF1 respectively) when a JMP or CALL instruction is executed. The contents of these two internal flip-flops are altered using the SEL MB instructions. During an interrupt service routine PC11 and PC12 are forced to logic 0.

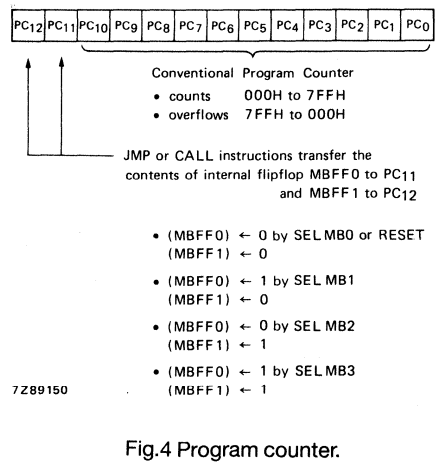



Fig.4 Program counter.

## 7 PROGRAM STATUS WORD

The program status word (PSW) is an 8-bit CPU register which stores information about the current status of the microcontroller (see Fig.5). All bits can be read using the MOV A,PSW instruction. Only the PS bit can be written using the MOV PSW,A instruction. Table 1 shows the function of the PSW bits and when they are affected.

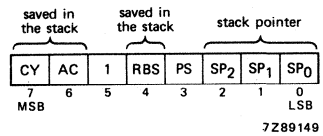



Fig.5 Program status word.

**Table 1** Program Status Word.

| BIT | NAME | FUNCTION | AFFECTED BY |
|---|---|---|---|
| 7 | CY | Carry, signals accumulator overflow | ADD, ADDC, DA, RLC, RRC, CLR C and CPL C instructions |
| 6 | AC | Auxiliary Carry, half carry | ADD and ADDC instructions |
| 5 | - | Not used, always 1 when read | |
| 4 | RBS | Register Bank Select<br>0: select register bank 0<br>1: select register bank 1 | SEL RB instructions |
| 3 | PS | Timer Prescaler Select<br>0: prescaler selected (÷32)<br>1: prescaler not selected (÷1) | MOV PSW,A instruction |
| 2<br>1<br>0 | SP2<br>SP1<br>SP0 | Stack Pointer bits 2-0 | CALL, RET, RETR instructions and interrupts |

## 8 CONDITIONAL JUMP LOGIC

The conditional jump logic enables several conditions to be tested by the user's program. Table 2 lists the conditional jump instructions which may be used to change the program sequence.

The DJNZ instruction decrements a designated register or data memory location and jumps if the contents are not zero. This instruction is useful for looping control.

**Table 2** Conditional jumps.

| TEST | JUMP CONDITION | JUMP INSTRUCTION |
|---|---|---|
| accumulator | all bits zero<br>any bit non-zero | JZ<br>JNZ |
| accumulator bit test | 1 | JB0 to JB7 |
| carry flag | 1<br>0 | JC<br>JNC |
| timer overflow flag | 1<br>0 | JTF<br>JNTF |
| test input T0 | 1<br>0 | JT0<br>JNT0 |
| test input T1 | 1<br>0 | JT1<br>JNT1 |
| register | non-zero | DJNZ Rr |
| data memory location | non-zero | DJNZ @Rr |

## 9 INTERRUPT LOGIC

The MAB84XX family handles external, serial I/O, and timer/event counter interrupts. The interrupt mechanism is single level; an executing interrupt service routine can not be interrupted by other interrupts. If several interrupt requests are detected simultaneously, they are serviced in order of priority (see Table 3). An interrupt request will only be serviced if the interrupt is enabled; each interrupt type has individual enable and disable instructions (see Table 3).

Before an interrupt is serviced, execution of the current instruction is first completed. The contents of the program counter, and bits 4, 6 and 7 of the program status word are then saved on the program counter stack, and a CALL to the corresponding interrupt vector is executed (see Table 3). The maximum interrupt latency time is 4 machine cycles. The interrupt service routine must be terminated by the RETR (return and restore) instruction. At least one instruction in the main program will then be executed before another interrupt service routine is serviced.

**Table 3** Interrupts.

| INTERRUPT | | | INSTRUCTION | |
|---|---|---|---|---|
| **PRIORITY** | **TYPE** | **VECTOR LOCATION** | **ENABLE** | **DISABLE** |
| 1 (highest) | external | 003 (ROM) | EN I | DIS I |
| 2 | serial I/O | 005 (ROM) | EN SI | DIS SI |
| 3 (lowest) | timer/event counter | 007 (ROM) | EN TCNTI | DIS TCNTI |

**Notes on interrupt service routines**

- While an interrupt service routine is in progress, the two most significant bits of the program counter are frozen at zero.
  Therefore: interrupt service routines and any subroutines called within interrupt service routines must reside (entirely) in memory bank 0.
  Within an interrupt service routine and within any subroutines called within an interrupt service routine, other memory banks cannot be selected and SEL MB instructions must not be used.
- Subroutines and nested subroutines called within an interrupt service routine must all end with RET. RETR would clear the interrupt in progress flag and further pending interrupts would then interfere with the interrupt service routine in progress.

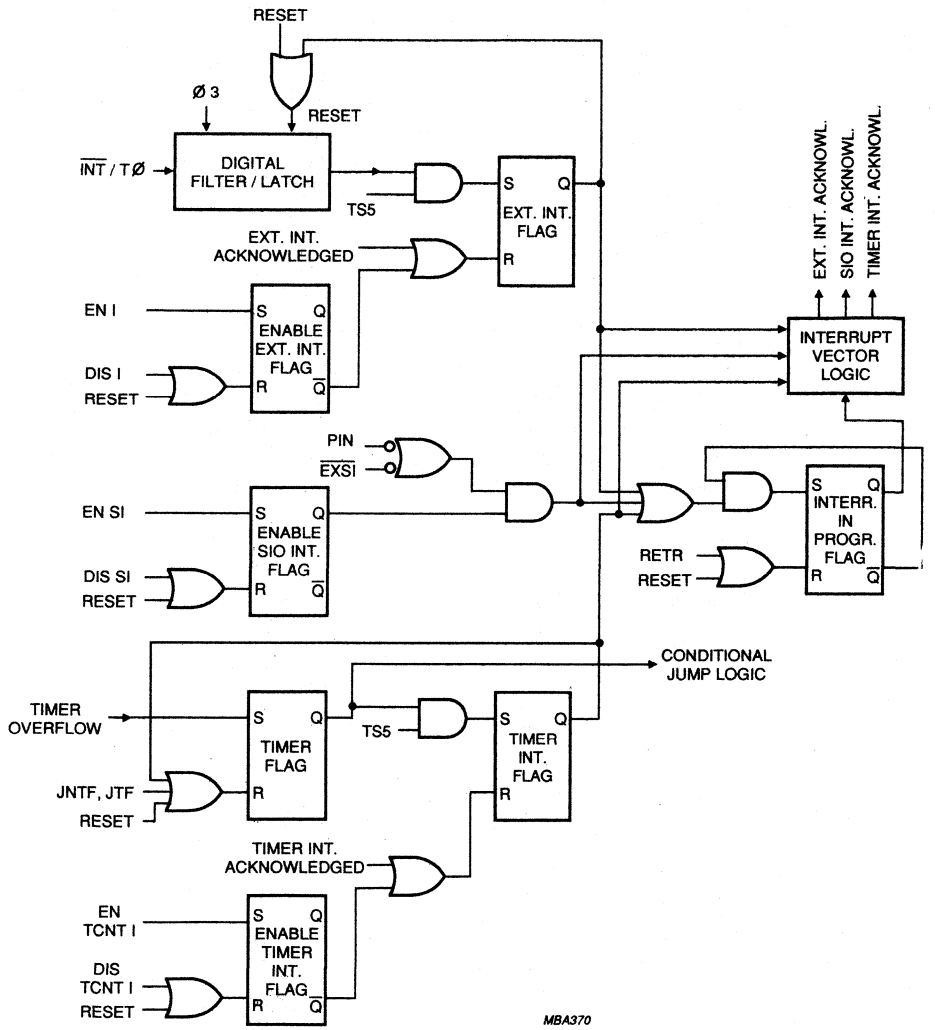



Fig.6 Interrupt logic.

**Notes:**

1. When the Interrupt In Progress (IIP) flag is set, other interrupts are latched but ignored until the RETR instruction is executed.
2. When the timer/counter interrupt flag (TF) has been enabled, the JTF and JNTF instructions will always find the flag at logic 0; this is because the timer flag is set at time slot 3 and reset at time slot 5 of the same cycle.

### 9.1 External interrupt

A HIGH-to-LOW transition on the $\overline{INT}$/T0 pin generates an external interrupt request which is always latched in a digital filter/latch (see Fig.6). To ensure that the transition is latched, the LOW state must exceed 7 XTAL clock periods after a HIGH state of more than 4 XTAL clock periods. When the External Interrupt Flag (EIF) is set, it resets the digital filter/latch (see Fig.6). If the external interrupt is enabled and no interrupt service routine is in progress, the external interrupt will be serviced. Simultaneously, the hardware removes the latched interrupt request.

During the forced CALL to the external interrupt vector location, EIF is reset and the digital filter/latch is re-enabled; the Interrupt In Progress (IIP) flag bit is also set so that other interrupts are latched but ignored until the RETR instruction is executed. After the RETR instruction has been executed, latched interrupts will be serviced in order of priority. Execution of a DIS I (disable external interrupt) instruction cancels a pending external interrupt request which has been latched in the EIF. An interrupt request which is latched in the digital filter/latch after the external interrupt has been disabled can't be cancelled.

### 9.2 Serial I/O interrupt

The serial I/O interrupt may be requested by the serial I/O interface. An interrupt request from the SIO interface will force the PIN interrupt flag to logic 0. If the serial I/O interrupt is enabled and no interrupt service routine is in progress, the serial I/O interrupt will be serviced. The interrupt flag is NOT automatically reset to the inactive state (by hardware) when a serial I/O interrupt is serviced; this must be done by user software.

### 9.3 Timer/event counter interrupt

A timer/event counter overflow generates a timer/event counter interrupt request. This interrupt request is always latched. If the timer/event counter interrupt has been enabled and if no interrupt service routine is in progress, the timer/event counter interrupt will be serviced. Simultaneously, the hardware removes the latched interrupt request. Execution of a DIS TCNTI (disable timer/event counter interrupt) instruction cancels a pending timer/event counter interrupt request which has been latched in the Timer Interrupt Flag.

### 9.4 Interrupt operation examples

Figures 7 to 10 show examples of how the interrupt mechanism operates.

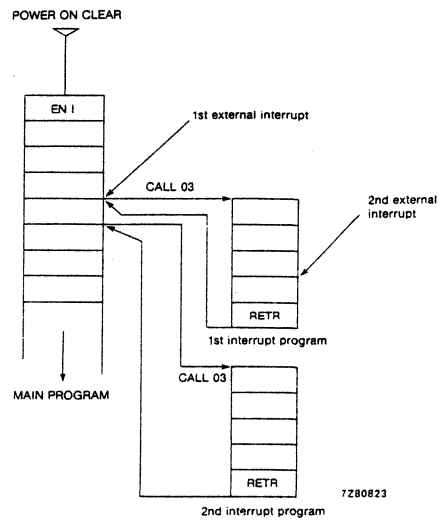



Fig.7 An external interrupt which occurs while another interrupt service routine in being executed remains latched until the executing routine is complete.

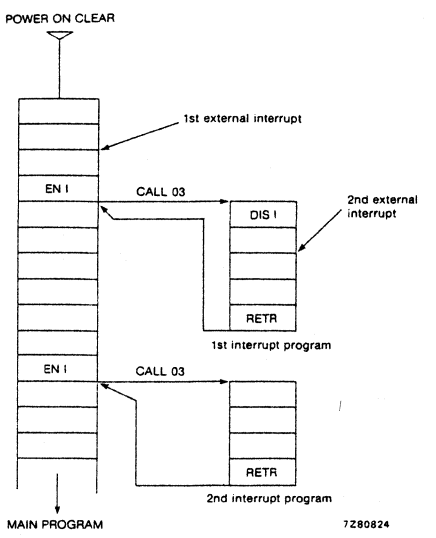



Fig.8 External interrupts are postponed until enabled.

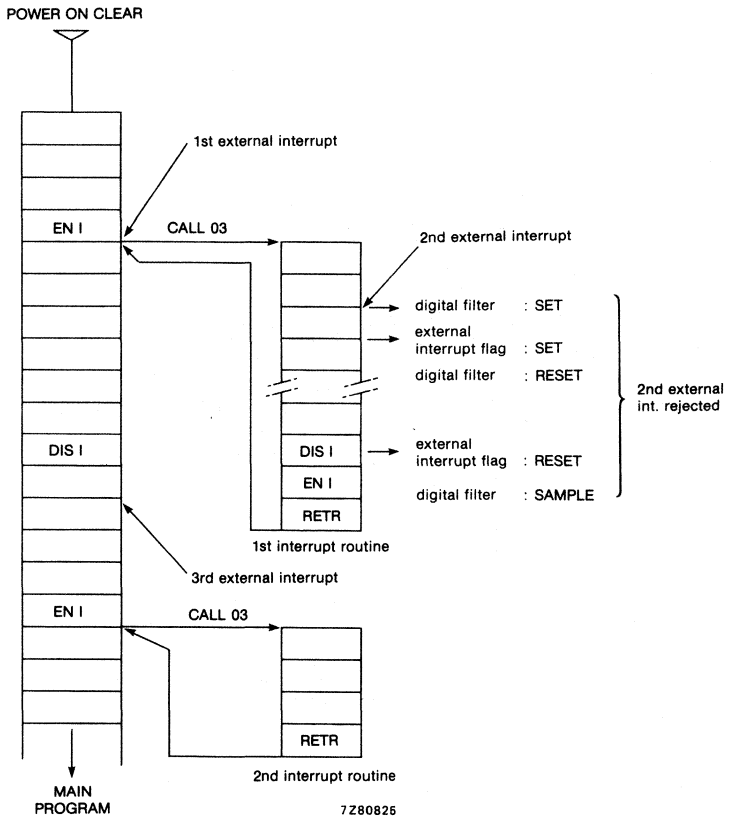



Fig.9 A previous external interrupt is cleared.

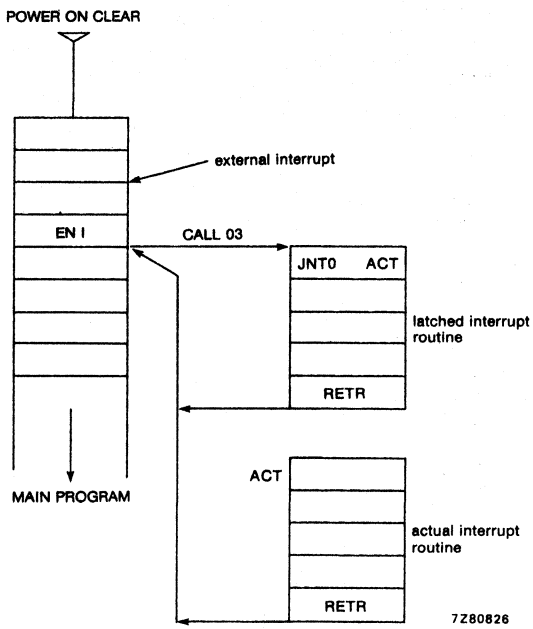



Fig.10 Detection of previous and actual external interrupt.

## 10 TIMER/EVENT COUNTER

The Timer/event counter (see Fig.11) can operate either as a timer or as an event counter.

- **Timer mode**; depending on the state of the PS-bit in the program status word, the internal 8-bit up-counter is incremented every machine cycle (PS = 1) or every 32 machine cycles (PS = 0).
- **Counter mode**; the internal 8-bit up-counter is incremented on every LOW-to-HIGH transition on pin T1. The HIGH state of T1 must exceed 4 XTAL clock periods after a LOW state of more than 4 XTAL clock periods. The maximum count rate is one increment per machine cycle.

When the 8-bit up-counter overflows, the timer overflow flag is set. If the timer/event counter interrupt is enabled, the timer interrupt flag (see Fig.6) is also set and the timer overflow flag is reset again in the same machine cycle. Table 4 details the instructions that control the timer/event counter. Testing the timer overflow flag, using the timer JTF (jump if timer flag = 1) or JNTF instructions, also clears the timer overflow flag. Starting the timer, using the STRT T instruction, also clears the pre-scaler. Reading the 8-bit up-counter does not disturb the counting process.

**Table 4** Timer/event counter control.

| FUNCTION | TIMER MODE | COUNTER MODE |
|---|---|---|
| clear | MOV T,A (A)=0 or RESET | MOV T,A (A)=0 or RESET |
| preset | MOV T,A | MOV T,A |
| start | STRT T | STRT CNT |
| stop | STOP TCNT or RESET | STOP TCNT or RESET |
| test | JTF/JNTF | JTF/JNTF |
| read | MOV A,T | MOV A,T |

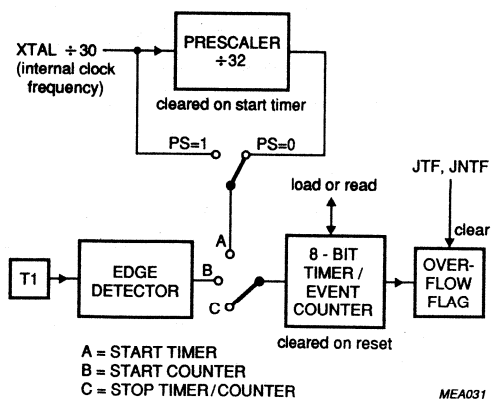



Fig.11 Timer/event counter.

## 11 I/O

Members of the MAB84XX family have up to 23 I/O lines arranged as follows:

- two 8-bit parallel ports
- one 4-bit parallel port
- a serial I/O which consists of the following two lines:
  - P2.3/SDA; data line shared with port line P2.3
  - **SCLK, dedicated clock line**
- $\overline{\text{INT}}$/T0, an external interrupt/ test input
- T1, a test input which may also be used as an input to the counter/timer

### 11.1 Parallel I/O ports

Up to three parallel ports are controlled by dedicated port instructions.

These ports can have up to 20 I/O port lines arranged as:

- Port 0: 8-bit parallel port (P0.0 to P0.7)
- Port 1: 8-bit parallel port (P1.0 to P1.7)
- Port 2: 4-bit parallel port (P2.0 to P2.3).

Output data written to a port is latched and remains unchanged until rewritten. Input data is not latched and must therefore remain present until read by an input instruction. Fig.12 shows the timing diagram for all ports using IN, OUTL, ANL and ORL instructions. For the OUTL instruction, data changes on time slot 7 of cycle 1. For the ANL and ORL instructions, port data changes on time slot 7 of cycle 2.

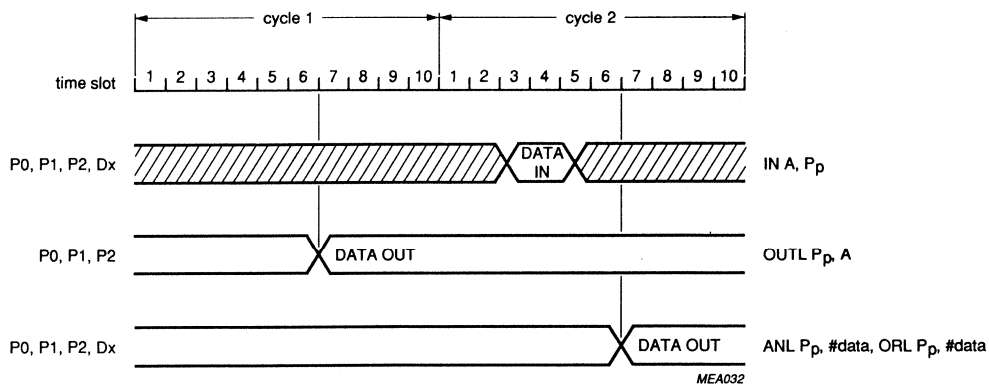



Fig.12 Port timing.

Input port lines are fully TTL compatible; output port lines can drive one TTL load. The high drive capability of Port 1 was intended to enable the 8 Port 1 output lines to drive 8 LEDs.

Three I/O mask options make it possible for parallel I/O port lines to be individually configured as follows:

- **Option 1** STANDARD I/O; quasi-bidirectional I/O (see Fig.13) consisting of a push-pull output with a weak pull-up transistor (TR3). If the output latch contains a logic 1, then the port line is pulled up to $V_{CC}$ via TR3. TR3 provides sufficient current for a HIGH level to one TTL input, yet can be pulled LOW by an external TTL or CMOS device, thus allowing the same port line to be used for both input and output. When used as an input, the port latch must be in a logic one state. When a logic 0 is written to the port latch, a low impedance transistor (TR1) in turned on to provide TTL current sinking capability. To provide a fast logic 0 to logic 1 transition, TR2 is active during 1 XTAL clock cycle when a logic 1 is written to the port latch.
- **Option 2** OPEN DRAIN I/O WITH PULL-UP; open drain output with pull-up transistor, TR3 (see Fig.14).
- **Option 3** OPEN DRAIN I/O WITHOUT PULL-UP; open drain output without pull-up transistor (see Fig.15). Application as an output requires the connection of an external pull-up resistor.

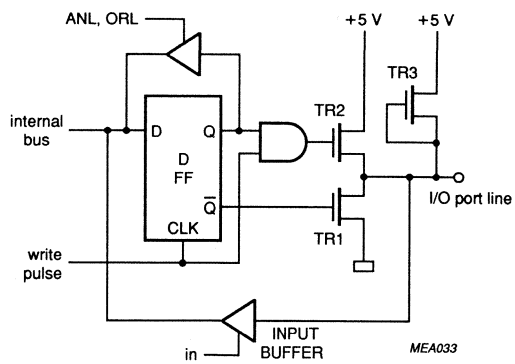



Fig.13 Standard push-pull I/O with pull-up transistor (option 1).

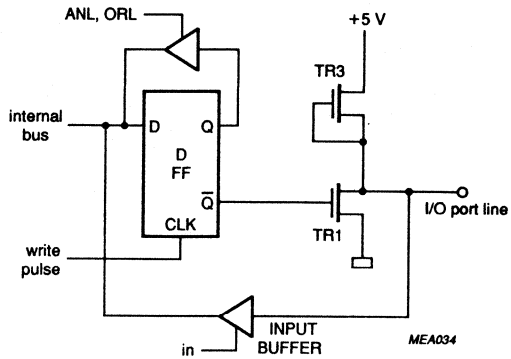



Fig.14 Open drain I/O with pull-up transistor (option 2).

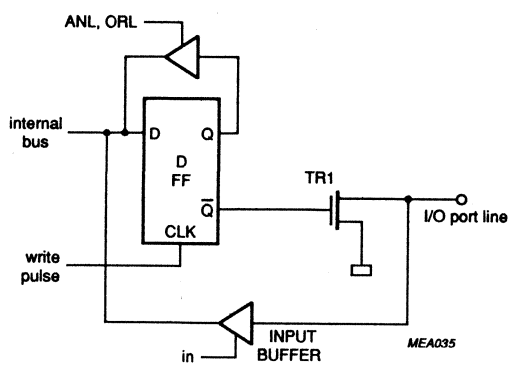



Fig.15 Open drain I/O without pull-up transistor (option 3).

### 11.2 Test inputs T1 and $\overline{INT}$/T0

The T1 input line can be used as:

- a test input for jump instructions JT1 and JNT1
- an input for zero voltage cross-over detection
- an external input to the event counter.

A pull-up resistor can be provided as a ROM mask option; this is useful when the input is from a switch or a standard TTL output.

When used as a test input, the JT1 and JNT1 instructions test the T1 pin for logic 1 and logic 0 respectively.

The T1 input has a self biasing circuit which can detect when an AC signal crosses zero (see Fig.16).

A LOW-to-HIGH transition on the T1 input increments the timer/event counter when the timer/event counter is in the counter mode.

When used in conjunction with the timer/event counter interrupt, zero cross-over detection is useful in thyristor control of power equipment.

The $\overline{INT}$/T0 input line can be used as:

- a test input for jump instructions JT0 and JNT0
- an external interrupt input.

### 11.3 Serial I/O

The MAB84XX on-chip serial I/O (SIO) hardware enables MAB84XX family members to be interconnected via the two-line serial $I^2C$ bus. The SIO allows two or more devices to communicate without interrupting other devices on the bus. This is achieved by allocating a specific 7-bit address to each device on the bus. Each device only reacts to messages prefixed with the specific address of the device or the general call address. Address recognition is handled by the SIO hardware, and the microcontroller is interrupted only when a valid address has been detected. The SIO hardware thus saves considerable CPU resources and memory requirements (compared to a software serial interface).

When address recognition is not required (e.g. a configuration with only two microcontrollers connected to the serial bus), direct data transfer without addressing can be performed. In multi-master systems, an automatic arbitration procedure stops two or more devices from transmitting simultaneously.

For more information on the SIO, see the chapter 'SERIAL I/O'.

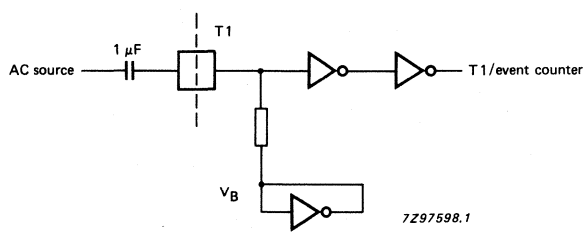



Fig.16 Self biasing circuit.

## 12 OSCILLATOR

The oscillator circuit must be completed by connecting one of the following timing elements between the XTAL1 and XTAL2 pins:

- a quartz crystal
- a ceramic resonator
- an inductor.

Two external load capacitors are also required.

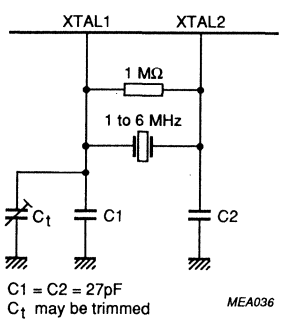
If a quartz crystal is used as the timing element, the crystal must operate in the fundamental mode of vibration. The clock may also be supplied by an external source. In this case the rise and fall times of the external clock ($t_r$, $t_f$) must be less that 10 ns. A pull-up resistor is required if the clock source is a TTL device. Table 5 gives the recommended L and C values for various oscillator frequencies when the timing element is an inductor.

**Table 5** LC oscillator timing.

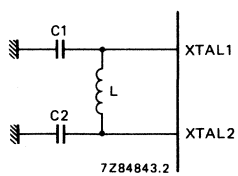
| FREQUENCY (MHz) | C1 = C2 (pF) | L (μH) |
|---|---|---|
| 3.0 | 33 | 100 |
| 4.0 | 33 | 56 |
| 4.4 | 33 | 47 |
| 5.0 | 33 | 33 |
| 6.0 | 33 | 22 |

Oscillator start-up time depends on the external timing element. The start-up time of a quartz crystal is several milliseconds. The start-up time of a ceramic resonator is tenths of a millisecond.

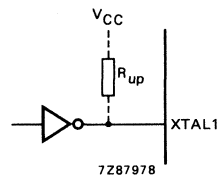
The XTAL clock may also be supplied by an external clock signal at pin XTAL1. In this case, XTAL2 is not connected.




(a) Crystal or ceramic resonator




(b) Inductor




(c) External

Fig.17 Oscillator circuit with various timing elements.

## 13 RESET

A reset signal initializes the microcontroller to a defined state.

### 13.1 Reset sequence

To ensure a correct reset, the reset signal at the RESET pin must be HIGH for at least 2 machine cycles after the power supply and clock have stabilized. The HIGH level at the RESET pin:

- sets the program counter to zero
- selects memory bank 0 and register bank 0
- sets the stack pointer to zero (000), pointing to RAM addresses 8 and 9
- resets interrupt flags (external, timer/event counter and serial I/O) to the inactive state
- disables interrupts (external, timer/event counter and serial I/O)
- stops the timer/event counter, then sets it to zero
- sets the timer prescaler to divide by 32
- resets the timer overflow flag
- sets all ports latches to logic one (input mode), except P2.3/SDA which is high impedance
- sets the serial I/O to the slave receiver mode and disables the serial I/O.

A reset does not affect the data memory contents.

The external power-on-reset circuit should consist of a 1 μF capacitor connected between $V_{CC}$ and the RESET pin (see Fig.18); a diode may also be connected between the RESET pin and ground to ensure a correct reset if the supply voltage falls momentarily. Alternatively the device can be reset by an external TTL compatible signal (see Fig.19). Figure 20 shows typical input characteristics.

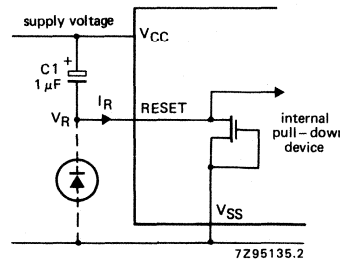



Fig.18 Power-on-reset.

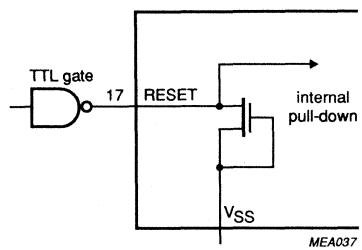



Fig.19 External reset.

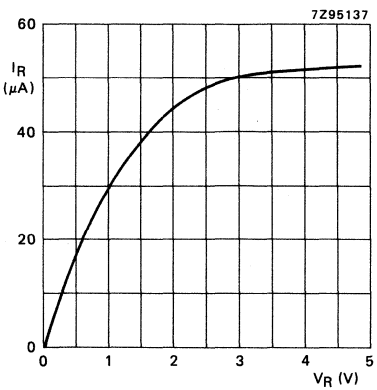



Fig.20 Typical input characteristics.

## 14 INSTRUCTION SET

The MAB84XX family instruction set consists of over 80 one and two byte instructions. Program code efficiency is high because all RAM locations and all ROM locations on a 256 byte page require only a single byte address.

Figure 21 shows the instruction map and Table 6 describes the symbols that are used.

**Table 6** Symbols and definitions.

| SYMBOL | DESCRIPTION |
|---|---|
| A | accumulator |
| addr | program memory address |
| Bb | bit designation (b = 0...7) |
| C | carry bit (bit CY) |
| CNT | event counter |
| data | 8-bit immediate data |
| I | interrupt |
| MBn | memory bank (n = 0...3) |
| MBFFn | memory bank flip-flop (n = 0, 1) |
| P | mnemonic for 'in-page' operation |
| PC | program counter |
| Pp | port designation (p = 0, 1, 2) |
| PSW | program status word |
| RB | register bank |
| RBS | register bank select flag |
| @Rr | 8-bit address register (r = 0, 1) |
| Rr | 8-bit register (r = 0...7) |
| Sn | serial I/O register (n = 0, 1, 2) |
| SP | stack pointer |
| T | timer |
| TCNT | timer/event counter |
| TF | timer flag |
| T0, T1 | test 0 and test 1 inputs |
| # | immediate data prefix |
| @ | indirect address prefix |
| (X) | contents of X |
| ((X)) | contents of location addressed by X |
| ← | is replaced by |
| ↔ | is exchanged with |

first hexadecimal character of opcode

second hexadecimal character of opcode

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | NOP | | | ADD A, #data | JMP page 0 | EN I | JNTF addr | DEC A | IN A,Pp 0 | IN A,Pp 1 | IN A,Pp 2 | | MOV A,Sn 0 | MOV A,Sn 1 | | |
| 1 | INC @ Rr 0 | INC @ Rr 1 | JB0 addr | ADDC A,#data | CALL page 0 | DIS I | JTF addr | INC A | INC Rr 0 | INC Rr 1 | INC Rr 2 | INC Rr 3 | INC Rr 4 | INC Rr 5 | INC Rr 6 | INC Rr 7 |
| 2 | XCH A, @Rr 0 | XCH A, @Rr 1 | | MOV A, #data | JMP page 1 | EN TCNTI | JNT0 addr | CLR A | XCH A,Rr 0 | XCH A,Rr 1 | XCH A,Rr 2 | XCH A,Rr 3 | XCH A,Rr 4 | XCH A,Rr 5 | XCH A,Rr 6 | XCH A,Rr 7 |
| 3 | XCHD A, @Rr 0 | XCHD A, @Rr 1 | JB1 addr | | CALL page 1 | DIS TCNTI | JT0 addr | CPL A | OUTL Pp,A 0 | OUTL Pp,A 1 | OUTL Pp,A 2 | | MOV Sn,A 0 | MOV Sn,A 1 | MOV Sn,A 2 | |
| 4 | ORL A, @Rr 0 | ORL A, @Rr 1 | MOV A, T | ORL A, #data | JMP page 2 | STRT CNT | JNT1 addr | SWAP A | ORL A,Rr 0 | ORL A,Rr 1 | ORL A,Rr 2 | ORL A,Rr 3 | ORL A,Rr 4 | ORL A,Rr 5 | ORL A,Rr 6 | ORL A,Rr 7 |
| 5 | ANL A, @Rr 0 | ANL A, @Rr 1 | JB2 addr | ANL A, #data | CALL page 2 | STRT T | JT1 addr | DA A | ANL A,Rr 0 | ANL A,Rr 1 | ANL A,Rr 2 | ANL A,Rr 3 | ANL A,Rr 4 | ANL A,Rr 5 | ANL A,Rr 6 | ANL A,Rr 7 |
| 6 | ADD A, @Rr 0 | ADD A, @Rr 1 | MOV T, A | | JMP page 3 | STOP TCNT | | RRC A | ADD A,Rr 0 | ADD A,Rr 1 | ADD A,Rr 2 | ADD A,Rr 3 | ADD A,Rr 4 | ADD A,Rr 5 | ADD A,Rr 6 | ADD A,Rr 7 |
| 7 | ADDC A, @Rr 0 | ADDC A, @Rr 1 | JB3 addr | | CALL page 3 | | | RR A | ADDC A,Rr 0 | ADDC A,Rr 1 | ADDC A,Rr 2 | ADDC A,Rr 3 | ADDC A,Rr 4 | ADDC A,Rr 5 | ADDC A,Rr 6 | ADDC A,Rr 7 |
| 8 | | | | RET | JMP page 4 | EN SI | | | ORL Pp,#data 0 | ORL Pp,#data 1 | ORL Pp,#data 2 | | | | | |
| 9 | OUTL PO,A | | JB4 addr | RETR | CALL page 4 | DIS SI | JNZ addr | CLR C | ANL Pp,#data 0 | ANL Pp,#data 1 | ANL Pp,#data 2 | | MOV Sn,#data 0 | MOV Sn,#data 1 | MOV Sn,#data 2 | |
| A | MOV @ Rr,A 0 | MOV @ Rr,A 1 | | MOVP A,@A | JMP page 5 | SEL MB2 | | CPL C | MOV Rr,A 0 | MOV Rr,A 1 | MOV Rr,A 2 | MOV Rr,A 3 | MOV Rr,A 4 | MOV Rr,A 5 | MOV Rr,A 6 | MOV Rr,A 7 |
| B | MOV @Rr, #data 0 | MOV @Rr, #data 1 | JB5 addr | JMPP @A | CALL page 5 | SEL MB3 | | | MOV Rr,#data 0 | MOV Rr,#data 1 | MOV Rr,#data 2 | MOV Rr,#data 3 | MOV Rr,#data 4 | MOV Rr,#data 5 | MOV Rr,#data 6 | MOV Rr,#data 7 |
| C | DEC @Rr 0 | DEC @Rr 1 | | | JMP page 6 | SEL RB0 | JZ addr | MOV A,PSW | DEC Rr 0 | DEC Rr 1 | DEC Rr 2 | DEC Rr 3 | DEC Rr 4 | DEC Rr 5 | DEC Rr 6 | DEC Rr 7 |
| D | XRL A, @Rr 0 | XRL A, @Rr 1 | JB6 addr | XRL A,#data | CALL page 6 | SEL RB1 | | MOV PSW,A | XRL A,Rr 0 | XRL A,Rr 1 | XRL A,Rr 2 | XRL A,Rr 3 | XRL A,Rr 4 | XRL A,Rr 5 | XRL A,Rr 6 | XRL A,Rr 7 |
| E | DJNZ @ Rr,addr 0 | DJNZ @ Rr,addr 1 | | | JMP page 7 | SEL MB0 | JNC addr | RL A | DJNZ Rr,addr 0 | DJNZ Rr,addr 1 | DJNZ Rr,addr 2 | DJNZ Rr,addr 3 | DJNZ Rr,addr 4 | DJNZ Rr,addr 5 | DJNZ Rr,addr 6 | DJNZ Rr,addr 7 |
| F | MOV A, @Rr 0 | MOV A, @Rr 1 | JB7 addr | | CALL page 7 | SEL MB1 | JC addr | RLC A | MOV A,Rr 0 | MOV A,Rr 1 | MOV A,Rr 2 | MOV A,Rr 3 | MOV A,Rr 4 | MOV A,Rr 5 | MOV A,Rr 6 | MOV A,Rr 7 |

MBA369

Fig.21 Instruction map.

CONTENTS

## 1 INTRODUCTION

This data sheet describes the shared properties of the PCF84CXXXA family of microcontrollers and its quickly growing number of derivative microcontrollers. For a particular microcontroller, this data sheet should be read in conjunction with the individual data sheet of the specific device.

## 2 FEATURES

- 8-bit CPU, ROM, RAM, I/O all in one package
- Up to 8 kbytes ROM
- Up to 256 bytes RAM
- Over 100 instructions (based on MAB8048) all of 1 or 2 cycles
- 8 or more quasi-bidirectional I/O port lines
- 8-bit programmable timer/event counter 1
- 3 single-level vectored interrupts: external, timer/event counter, SIO/derivative
- Two test inputs, one of which also serves as the external interrupt input
- Serial I/O interface (some devices only)
- Power-on-reset, Stop and Idle modes
- Supply voltage range: 2.5 to 6 V
- Clock frequency: 1 to 16 MHz
- Operating temperature: –40 to +85 °C
- Manufactured in silicon gate CMOS process.

## 3 GENERAL DESCRIPTION

The PCF84CXXXA family of microcontrollers provide up to 8 kbytes of program memory and up to 256 bytes of RAM. All devices include flexible I/O ports, an 8-bit programmable timer/event counter and a choice of single-level vectored interrupts. The instruction set is based on that of the well-known MAB8048. Being similar to the MAB8400 family of NMOS controllers, some devices can serve as CMOS replacements, especially where the lower power consumption and higher speed provide advantages.

A range of prototyping devices with external program memory and 'Piggy-backs', as well as emulation probes and prototyping systems are available.

## 4 BLOCK DIAGRAM

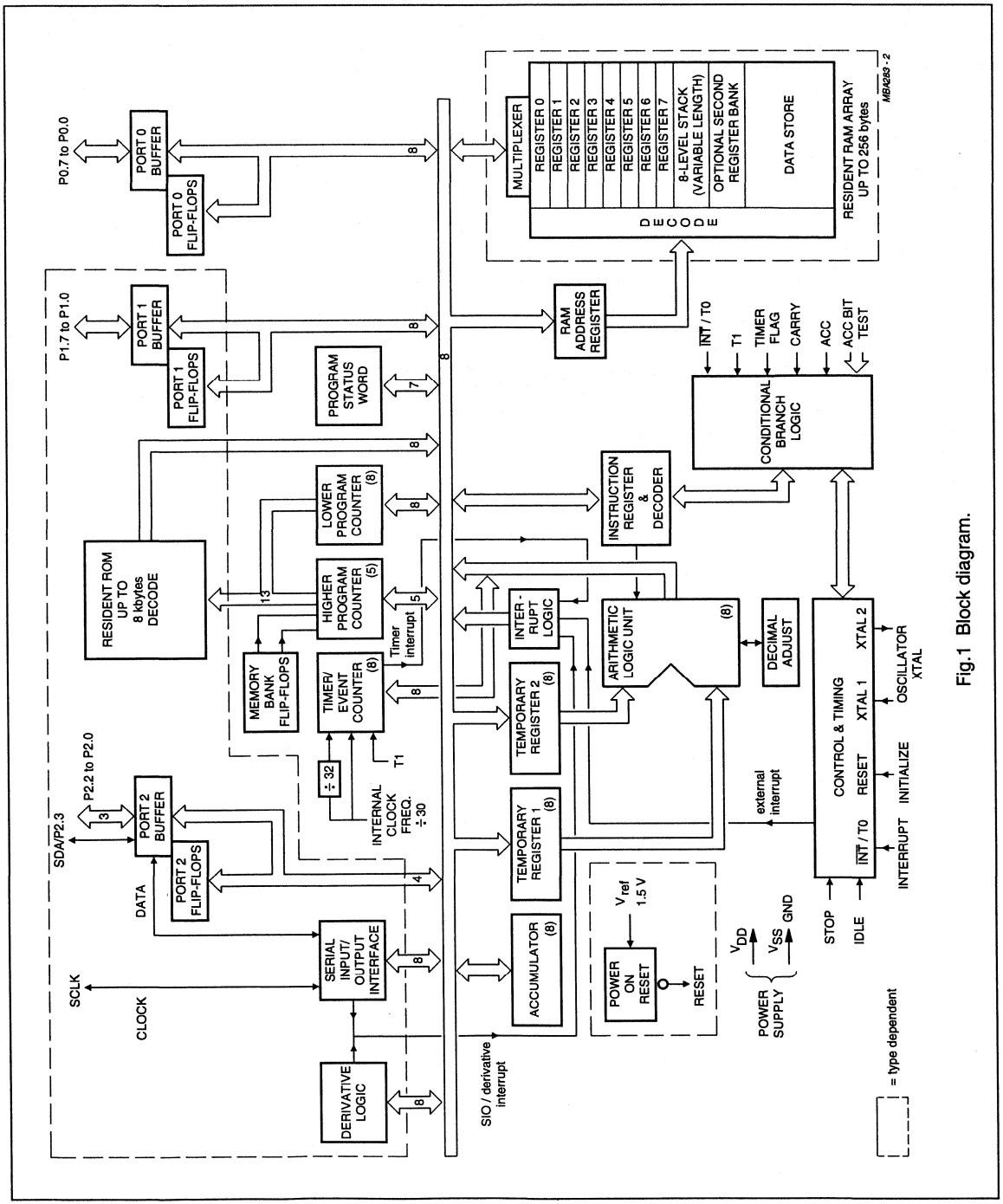



Fig.1 Block diagram.

## 5 PINNING INFORMATION

### 5.1 Pinning

For individual pinning configurations consult the data sheet of the specific device.

### 5.2 Pin description

Table 1 describes the common functions of the devices. For full details of pin descriptions consult the data sheet of the specific device.

**Table 1** Common functions.

| SYMBOL | TYPE | DESCRIPTION |
|---|---|---|
| $V_{SS}$ | P | Ground |
| $V_{DD}$ | P | Positive supply voltage |
| XTAL1 | I | Crystal oscillator/external clock input |
| XTAL2 | O | Crystal oscillator output |
| RESET | I | Reset input |
| $\overline{INT}$/T0 | I | Interrupt/Test 0 |
| T1 | I | Test 1/count input of 8-bit timer/event counter 1 |
| P0.0 to P0.7 | I/O | Port 0: quasi-bidirectional I/O lines |
| P1.0 to P1.7 | I/O | Port 1: quasi-bidirectional I/O lines |
| P2.0 to P2.2 | I/O | Port 2: quasi-bidirectional I/O lines |
| SDA/P2.3 | I/O | bidirectional data line of the serial I/O interface/Port 2: quasi-bidirectional I/O line |
| SCLK | I/O | bidirectional clock line of the serial I/O interface |

## 6 FUNCTIONAL DESCRIPTION

### 6.1 Central processing unit

The PCF84CXXXA family provides an adequate instruction set with arithmetic, logic, branching, input/output and control facilities. Special highlights are the instructions for BCD arithmetic, nibble handling, conditional branches, loop control (DJNZ) and table look-up (MOVP).

Code and execution efficiency is achieved by using a maximum of two bytes and two execution cycles per instruction (see Chapter 7).

### 6.2 Program memory

The program memory consists of up to 8 kbytes of read-only memory (ROM). Each location is directly addressable by the Program Counter. The program memory is mask-programmed at the factory. Figure 2 illustrates the program memory map.

Four program memory locations are of special importance:

- Location 0: first instruction to be executed after the processor is reset
- Location 3: first instruction of an external interrupt ($\overline{INT}$/T0) routine
- Location 5: first instruction of a SIO/derivative interrupt routine
- Location 7: first instruction of a timer/event counter interrupt routine.

Only 11 bits of the 13-bit Program Counter function as a counter. The two most significant bits can only be preset. The program memory is therefore, structured into banks of 2 kbytes. Transfer of control to other memory banks is performed by unconditional branches (JMP) or subroutine calls (CALL) when another memory bank has been pre-selected (by SEL MB instruction).

Each program memory bank is further divided into 8 pages of 256 bytes. Indirect (JMPP) and conditional branches cannot cross page boundaries.

### 6.3 Data memory

Data memory consists of up to 256 bytes of random access memory (RAM). All locations are indirectly addressable using RAM pointer registers. Up to 16 register locations are directly addressable. Data memory also includes an 8-level Program Counter stack addressed by a 3-bit Stack Pointer. All RAM locations make efficient program loop counters if used with the decrement register and test instruction (DJNZ). Figure 3 illustrates the data memory map.

#### 6.3.1 Working registers

Locations 0 to 7 are working registers. They are accessible by efficient one byte/one cycle instructions, thus making these locations suitable for frequently accessed intermediate results.

As an alternative to locations 0 to 7, locations 24 to 31 may be used as working registers. Register bank selection is made by SEL RB0/RB1 instructions. Register bank 1 may be used as an extension of register bank 0, as an alternative register bank for interrupt service or as general purpose data memory.

The first two locations of each bank (R0, R1, R0' and R1') serve as RAM pointers that indirectly address all RAM locations.

#### 6.3.2 Program Counter stack

Locations 8 to 23 may be used as an 8-level Program Counter stack reserving 2 locations per level, or as general purpose RAM. The stack (see Fig.5) saves return addresses and status during interrupt or subroutine servicing. Nesting of subroutines and/or interrupts is permitted up to 8-levels deep.

The 3-bit Stack Pointer always points to the next free stack level. Following device reset, the Stack Pointer points to level 0 (locations 8 and 9). On each subroutine call (CALL) or interrupt, the contents of the Program Counter and bits 4, 6 and 7 of the Program Status Word are transferred to the level indicated by the Stack Pointer. The Stack Pointer increments and points to the next free level. Overflow from level 7 to level 0 occurs after nesting eight levels deep. Further subroutine calls and/or interrupts must not occur at this stage since this would result in loss of program content; overriding level 0 content.

Return from interrupt must be performed by the RETR instruction, which decrements the Stack Pointer and restores the Program Counter and Program Status Word, valid before the interrupt occurred. Return from subroutine should be performed by the RET instruction. In contrast to RETR, RET does not restore the Program Status Word.

As a general rule, the use of RETR in conjunction with a subroutine call is not recommended. The use of RETR must also be avoided with subroutines called from interrupt routines because it prematurely terminates the interrupt state (see Section 6.6).

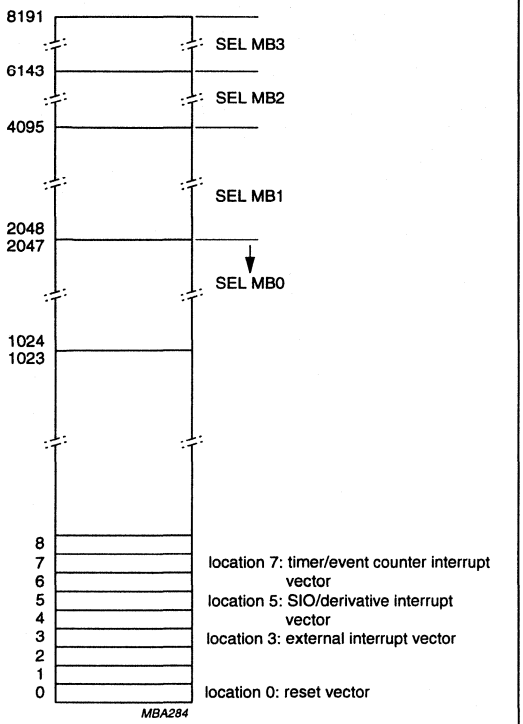



Fig.2 Program memory map.

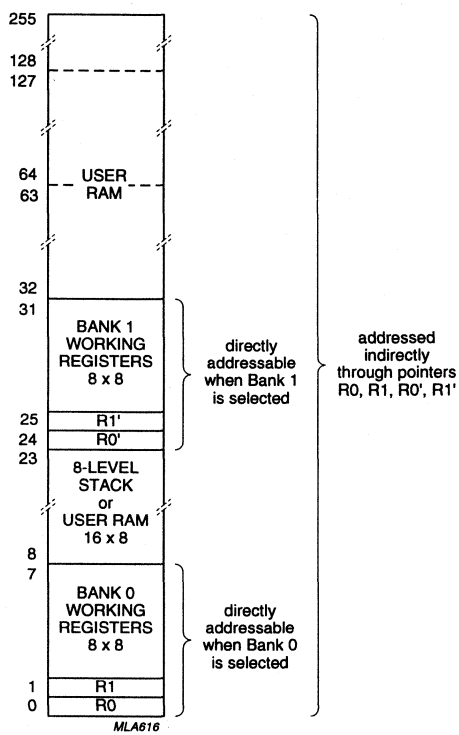



Fig.3 Data memory map.

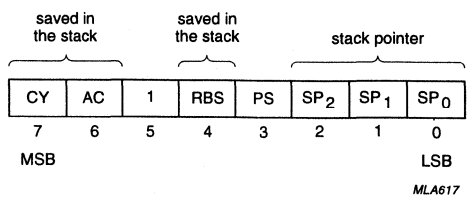



Fig.4 Program Status Word.

### 6.4 Program Counter

The 13-bit Program Counter is able to address up to 8 kbytes of ROM (see Fig.6). 11 bits (PC0 to PC10) are auto-incrementing. The two most significant bits (PC11 and PC12) must be changed under program control by SEL MB followed by a JMP or CALL instruction.

### 6.5 Program Status Word

The Program Status Word (PSW) is an 8-bit register in the CPU which stores information about the current status of the microcontroller (see Fig.4).

The PSW bits are:

- Bits 0 to 2: Stack Pointer bits (SP0, SP1, SP2)
- Bit 3: timer Prescaler Select (PS); 0 = modulo-32, 1 = modulo-1 (no prescaling)
- Bit 4: working Register Bank Select (RBS); 0 = register bank 0, 1 = register bank 1
- Bit 5: not used (fixed at 1)
- Bit 6: Auxiliary Carry (AC); half-carry bit generated by an ADD instruction and used by the decimal adjust instruction DA A
- Bit 7: Carry (CY); the carry flag indicates that the previous operation resulted in an overflow of the Accumulator.

All bits can be read using the MOV A, PSW instruction. Bits 0, 1 and 2 are affected by CALL, RET, RETR and interrupts. Bit 3 can be controlled by MOV PSW, A and bit 4 by SEL RB instructions. Bit 6 is set and cleared as a side-effect of ADD and ADDC instructions. Bit 7 is affected by ADD, ADDC, DA, RLC, RRC, CLR C and CPL C instructions.

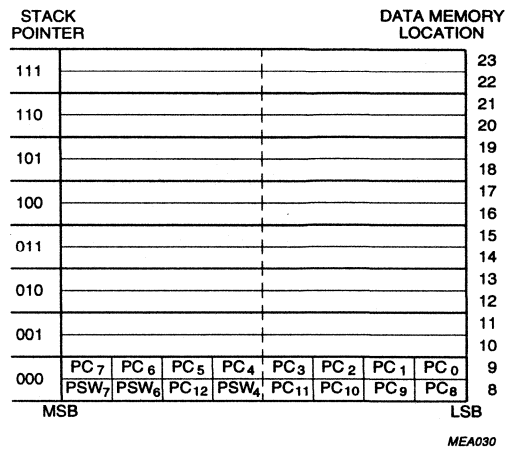



Fig.5 Program Counter Stack.

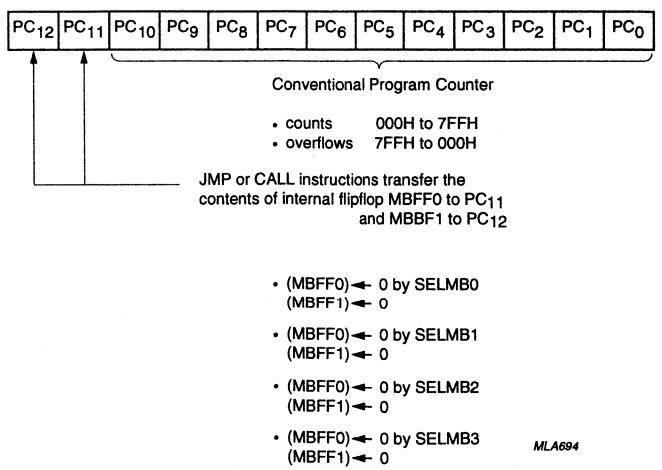



Fig.6 Program Counter.

## 6.6 Interrupts

External, SIO/derivative and timer/event counter interrupts are handled by the PCF84CXXXA family. The interrupt mechanism is single level, i.e. an executing interrupt routine cannot be pre-empted unless by reset. Further interrupt requests are latched. If several interrupt requests are detected simultaneously, they are honoured according to their priority:

- External interrupt (highest priority)
- SIO/derivative interrupt
- Timer/event counter interrupt (lowest priority).

An interrupt request is only sensed if the corresponding enable flag is set (see Fig.7). When the request is honoured, the contents of the Program Counter and bits 4, 6 and 7 of the Program Status Word are saved on the Program Counter stack. The Program Counter is loaded with the appropriate interrupt vector, thereby indicating the beginning of the interrupt routine. Since the Accumulator is not automatically saved, it must be saved and restored by user software. The interrupt routine must be terminated by the RETR (return and restore) instruction. At least one instruction of the main program will then be executed before another interrupt routine is entered. To avoid erroneous real-time programs, a few words of caution:

- While the interrupt is in progress, the two most significant bits of the Program Counter are frozen at zero. Thus, interrupt routines and subroutines called from interrupt routines must reside entirely in bank 0.
- The SEL MB instruction must not be used in interrupt routines and in subroutines called from interrupt routines. Otherwise, the changed contents of MBFF0 and MBFF1 (see Fig.6) may lead to erroneous JMP and CALL destinations after return from interrupt.
- Subroutines and nested subroutines called from the interrupt routine must all end with RET since RETR clears the Interrupt In Progress flag (IIP), as a side-effect (see Figs 7 and 8). Further pending interrupts would then interfere with the interrupt routine in progress.

### 6.6.1 External interrupt

A HIGH-to-LOW transition on the $\overline{INT}$/T0 pin is latched in the digital filter/latch if the LOW state exceeds 7 clock periods after a HIGH state of more than 4 clock periods. If the external interrupt is enabled the External Interrupt Flag (EIF) is also asserted, thus constituting a valid external interrupt request. As soon as the IIP is clear, indicating that no interrupt routine is in progress, the external interrupt is invoked by a forced CALL to location 3. The EIF is simultaneously cleared (see Figs 7 and 8). The interrupt routine may acknowledge the interrupt via port lines. Execution of a DIS I (disable external interrupt) instruction cancels a stored interrupt request by clearing both the digital filter/latch and the EIF.

### 6.6.2 SIO/Derivative interrupt

The SIO/derivative interrupt is shared between the serial I/O interface (if available) and the derivative logic (if available). Software polling may be necessary to determine the origin of a request.

An interrupt condition in the serial I/O interface and/or the derivative logic will pull the PIN line LOW. If the SIO/derivative interrupt is enabled and no interrupt routine is in progress, the SIO/derivative interrupt routine will be invoked by a forced CALL to program memory location 5. The SIO/derivative interrupt routine must include instructions that will remove the cause of the SIO/derivative interrupt and thus reset PIN to its inactive HIGH state (for further details see Section 6.11). For derivative interrupts, consult the data sheet of the specific device.

### 6.6.3 Timer/event counter interrupt

If the timer/event counter interrupt is enabled, a timer/event counter 1 overflow sets the Timer Interrupt Flag (TIF). As soon as IIP is clear, meaning that no interrupt routine is in progress, the timer/event counter interrupt routine is invoked by a forced CALL to program memory location 7. The TIF is simultaneously cleared (see Figs 7 and 8). Execution of a DIS TCNTI (disable timer/event counter interrupt) instruction cancels a stored interrupt request by clearing TIF.

The timer/event counter interrupt may also be used to simulate a second external interrupt. After an enable timer/event counter interrupt (EN TCNTI), the counter mode is enabled by a STRT CNT instruction which loads FFH (the state preceding overflow) into the counter. A positive edge on the T1 pin will overflow the counter and set TIF.

## 6.7 Interrupt/Test 0 Input ($\overline{INT}$/T0)

The $\overline{INT}$/T0 input has two purposes:

- External interrupt input (see Section 6.6)
- Test 0 input.

When used as a Test 0 input (external interrupt disabled) the conditional branch instruction JT0 will cause a jump if $\overline{INT}$/T0 = 1. The conditional branch instruction JNT0 will also cause a jump if $\overline{INT}$/T0 = 0. If $\overline{INT}$/T0 is not used, it must be tied to $V_{DD}$ or $V_{SS}$.

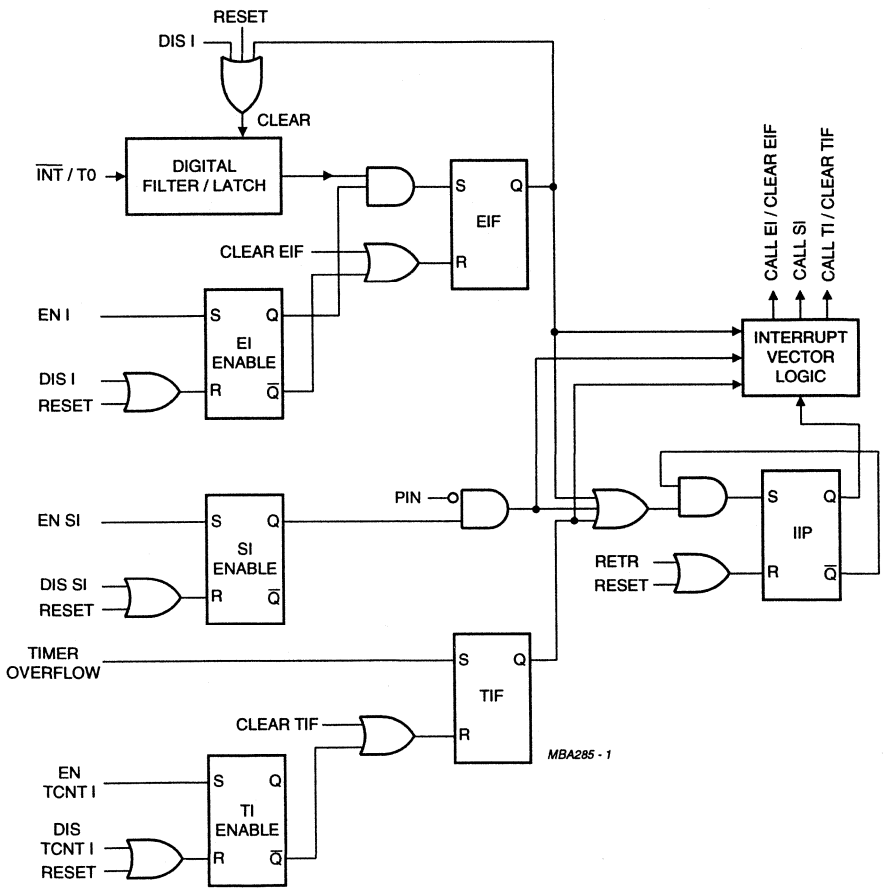



Fig.7 Simplified interrupt logic schematic (the R input overrules the S input for all flags).

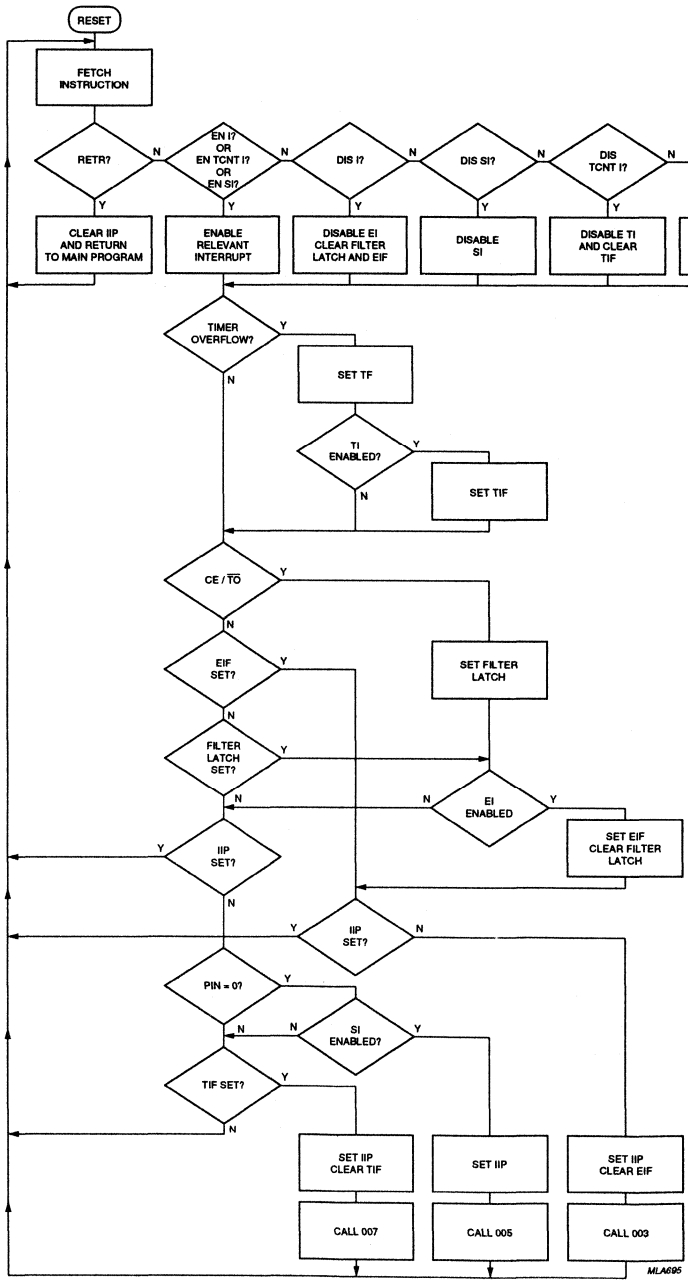



Fig.8 Flow chart illustrating CPU control in the presence of interrupts.

### 6.8 Timer/event counter 1

An internal 8-bit up counter is provided. The counter can be preset and read by the MOV T, A and MOV A, T instructions.

When the counter is to be used in the timer mode, a STRT T (start timer) instruction must be executed. Depending on the PS bit in the Program Status Word, the counter will increment every machine cycle (PS = 1, $\frac{1}{30}f_{xtal}$) or every 32 machine cycles (PS = 0, $\frac{1}{960}f_{xtal}$). STRT T clears the prescaler (see Fig.9) which is not otherwise accessible.

To count external events a STRT CNT (start event counter) instruction must be executed. A LOW-to-HIGH transition on pin T1 is counted if the HIGH state exceeds 4 clock periods after a LOW state of more than 4 clock periods. The maximum count rate is one increment per machine cycle ($\frac{1}{30}f_{xtal}$).

The timer mode and the event counter mode are both inhibited after reset or by executing a STOP TCNT (stop timer/event counter) instruction (see Fig.9).

In both the timer and in event counter modes, overflow has two effects:

- If the timer/event counter interrupt is enabled TIF is asserted thereby generating a timer/event counter interrupt request (see Section 6.6).
- The Timer Flag (TF) is set. TF can be tested by conditional branch instructions JTF (jump if TF = 1) or JNTF (jump if TF = 0). The JTF and JNTF instruction, as a side-effect, reset TF. The only other way to clear TF is to reset the microcontroller.

### 6.9 Test 1/count input (T1)

The T1 input has two purposes:

- Count input of 8-bit timer/event counter 1 (see Section 6.8)
- Test 1 input.

When used as a Test 1 input the conditional branch instruction JT1 will cause a jump if T1 = 1. The conditional branch instruction JNT1 will also cause a jump if T1 = 0. If T1 is not used, it must be tied to $V_{DD}$ or $V_{SS}$.

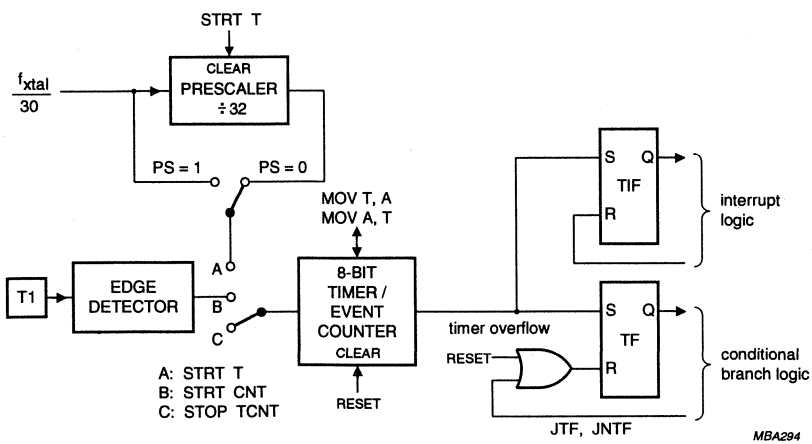



Fig.9 Timer/event counter 1.

### 6.10 Parallel ports

Three standard quasi-bidirectional I/O ports are defined:

- Port 0: parallel port of 8 lines (P0.0 to P0.7)
- Port 1: parallel port of 8 lines (P1.0 to P1.7)
- Port 2: parallel port of 4 lines (P2.0 to P2.2, SDA/P2.3).

Several members of the PCF84CXXXA family provide all 20 port lines. The eight Port 0 lines (P0.0 to P0.7) are available as a minimum. In addition to the standard ports, many PCF84CXXXA microcontrollers offer a variety of derivative ports. Please consult the data sheet of the specific device.

In general, all parallel ports can be used as either inputs or outputs. Output data written to a port is latched and remains unchanged until rewritten. If the port is used as an input, the external data is not latched and must remain stable until it is accessed by the CPU.

The standard port configuration is illustrated in Fig.11. When a logic 0 is written to the master/slave flip-flop, TR2 and TR3 are both in the OFF condition. TR1 turns ON and drives the output to $V_{SS}$.

When a logic 1 is written to the master/slave flip-flop, TR1 turns OFF. TR2 and TR3 both turn ON driving the output rapidly to $V_{DD}$. TR2 remains in the ON condition for the duration of the write pulse only. The constant current source is responsible for keeping the output line high. Sufficient source current is available for a TTL load HIGH level; the line can, however, be overridden by an external device. This is used when the port line serves as an input, but it may also be useful for wired-OR applications. In the latter case, unnecessary current through external devices is avoided since repeated logic 1 write operations will not activate TR2. The booster transistor TR2 is only asserted during a LOW-to-HIGH transition of the master/slave flip-flop. If the port line is to be used as an input, a logic 1 should first be stored in the master/slave flip-flop to turn TR1 OFF.

Access to Ports 0, 1 and 2 is provided by the parallel input/output instructions IN, OUTL, ANL and ORL. IN inputs port data to the Accumulator. OUTL outputs Accumulator data to the port. ANL and ORL are used for data manipulation in the port flip-flop. In contrast to Ports 0, 1 and 2, derivative ports are accessed by the derivative input/output instructions MOV, ANL and ORL. ANL and ORL are used for data manipulation in the port flip-flop. MOV is used for all data transfers between port and Accumulator. The source data for the Accumulator can be loaded from either the port line or the port flip-flop. Two derivative addresses are therefore provided per port (see Table 2).

All standard and derivative port accesses are performed by two-cycle instructions. Their instruction timing is shown in Fig.10. For input, data on port lines is sensed during timeslots 3 and 4 of machine cycle 2 (see Sections 6.12 and 6.13). For output, the data change occurs in timeslot 7. For OUTL, data changes during machine cycle 1. For ANL, ORL and MOV Dx, A, data changes during machine cycle 2.

**Table 2** Derivative port address pair.

| ADDRESS | TYPE | ACCESS |
|---|---|---|
| 8-bit line address | R | derivative port line |
| 8-bit flip-flop address | R/W | derivative port flip-flop |

Three port output mask options are available:

Option 1 Standard Port; quasi-bidirectional I/O with switched pull-up current source of 100 µA (typ.) and p-channel booster transistor TR2. TR2 is only active for 1 clock cycle during LOW-to-HIGH transitions (see Fig.11).

Option 2 Open Drain; quasi-bidirectional I/O with only an n-channel open drain output. Application as an output requires connection of an external pull-up resistor (see Fig.12). If unused, an option 2 output should be tied to $V_{SS}$. This keeps the input path from floating, thereby avoiding undesirable current flow through input stages.

Option 3 Push-pull; drive capability of the output will be 5 mA (typ.) at $V_{DD}$ = 3 V in both polarities. Since short circuit currents would flow during input, push-pull lines must only be used as outputs (see Fig.13).

If available, SDA/P2.3 is shared between the parallel Port 2 and the serial I/O interface. Therefore, only option 2 is permitted for SDA/P2.3. For the remaining standard port lines (P0.0 to P2.2), all three options are generally available.

Besides port output mask options, the port flip-flop state, after reset, may be specified for each individual port line (except SDA/P2.3). Usually the 'set option' will be selected, which avoids short-circuits for ports intended as inputs. However, there may be cases in which the port should output a logic zero after reset. The user may then specify the 'reset option' for certain port lines.

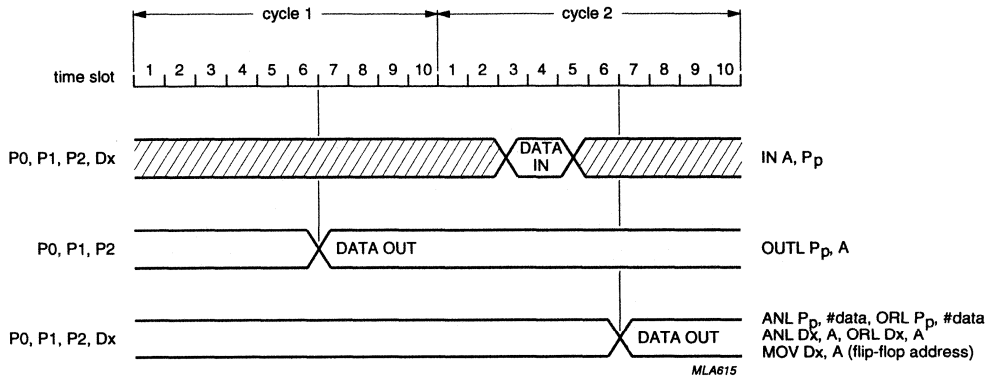



Fig.10 Input /output timing of standard and derivative ports.

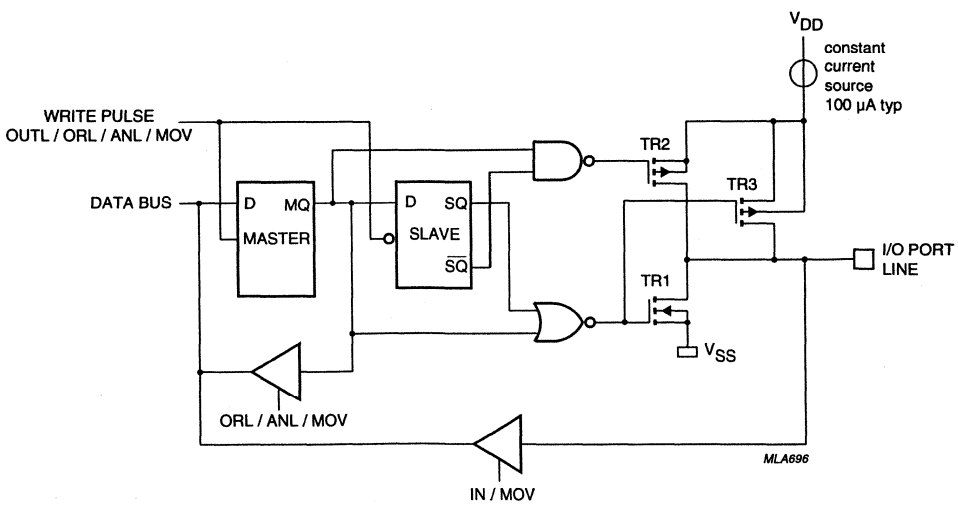



Fig.11 Standard output with switched current source.

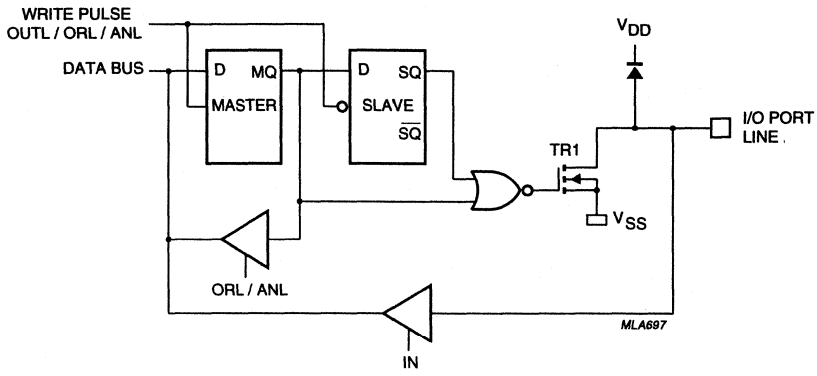



Fig.12 Open drain output.

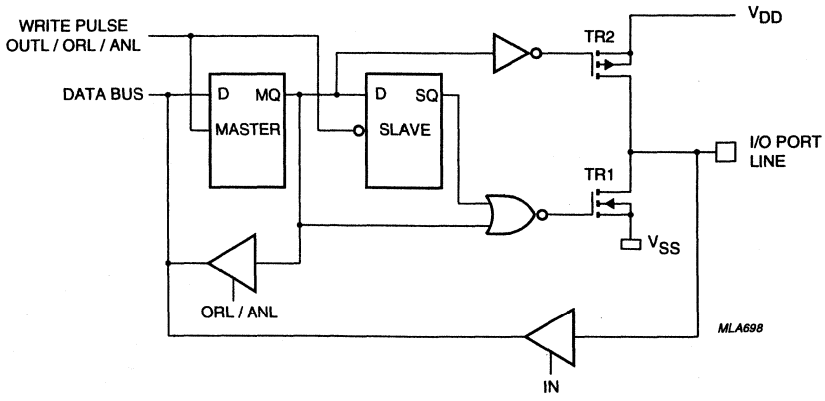



Fig.13 Push-pull output.

### 6.11 Serial I/O interface

Many members of the PCF84CXXXA family have a serial I/O interface ($I^2C$-bus or 'Inter-Integrated Circuit Bus'). This two-line serial bus extends the microcontroller capabilities when implemented with the powerful $I^2C$-bus devices of the PCF85XX, PCD33XX and 'Clips' peripheral families.

Microcontrollers that do not have a serial I/O interface can simulate it by software, by using port pins. However, such microcontrollers must continuously monitor the serial bus. As well as degrading the maximum data transfer rate, this approach may also consume significant processing and memory resources.

If available, however, the serial I/O interface detects the valid 7-bit $I^2C$-bus address of the device, transfers serial data and provides data conversion to and from parallel format, all without disrupting program execution. Only when a complete byte has been transferred, an interrupt is requested by which the next data byte can be written to or read out of the serial I/O interface. The serial I/O interface also facilitates the implementation of multimaster systems in which two or more microcontrollers communicate via the same $I^2C$-bus. An automatic arbitration procedure resolves bus conflicts.

The $I^2C$-bus consists of a bidirectional clock line (SCL) and a bidirectional data line (SDA). Whereas SCL uses the dedicated pin SCLK, SDA and Port line P2.3 share the pin, SDA/P2.3. When the serial I/O interface is enabled, SDA/P2.3 is disabled as a port line. Input signals on SCLK and SDA are filtered for enhanced noise immunity. When used as outputs, SCLK and SDA/P2.3 require an external pull-up resistor because they are open drain. If unused, SCLK and SDA/P2.3 should be tied to $V_{SS}$ (see Section 6.10, Option 2).

Communication between CPU and serial I/O interface is handled through the four serial I/O interface registers S0, S0', S1 and S2 (see Fig.14).

A detailed description of the $I^2C$-bus specification, with applications, is given in the brochure *"The $I^2C$-bus and how to use it"*. This brochure may be ordered using the code 9398 393 40011.

#### 6.11.1 Data shift register (S0)

The data shift register converts serial data to a parallel format and vice versa. The leading bit of a serial transfer corresponds to the most significant bit of the parallel word. An interrupt request is issued after transfer of a complete byte and after detection of the valid $I^2C$-bus address. Register S0 is read by MOV A, S0. It is written by MOV S0, A or MOV S0, #data if the ESO (Enable Serial I/O) bit in the Status Register (S1) is set.

#### 6.11.2 Address register (S0')

The address register contains the 7-bit $I^2C$-bus address of the device and the ALS (Always Selected) bit. When ALS is zero, which is the recommended mode of operation, bus transfers are ignored unless the valid device address immediately follows the start condition. Besides the stored 7-bit address, the 'general call address' (pre-defined as zero) is also acceptable as a valid address. If ALS is set, however, any transfer on the bus will be stored in the data shift register.

The address register S0' is write-only. It can be written by MOV S0, A and MOV S0, #data if the ESO (Enable Serial I/O) bit in the Status Register (S1) is zero.

#### 6.11.3 Clock control register (S2)

The Clock Control Register defines the frequency of $f_{SCLK}$ as the microcontroller clock frequency divided by an integer (see Table 3). It also defines ASC (Asymmetrical Clock) and ACK (Acknowledge).

If ASC = 1, the generated SCLK has a duty cycle of approximately 75%. The asymmetrical clock limits the $I^2C$-bus transmission rate to below 55 kHz. Divisors 39, 45 and 51 are not allowed if ASC = 1. However, an SCLK duty cycle of approximately 50% results if ASC = 0. This permits $I^2C$-bus transmission rates of up to 100 kHz. All divisors of Table 3 are available. It is, therefore, recommended to select ASC = 0.

For the normal $I^2C$-bus protocol ACK must be set. After each byte transfer an extra SCLK pulse is generated during which the receiver may acknowledge reception. If ACK is zero, no acknowledge phase is available. This mode is temporarily used when a master/receiver refuses the acknowledgement in order to signal an end of transmission to the slave transmitter (see Section 6.11.4.9).

The Clock Control Register (S2) is write-only. It can be written by MOV S2, A and MOV S2, #data.

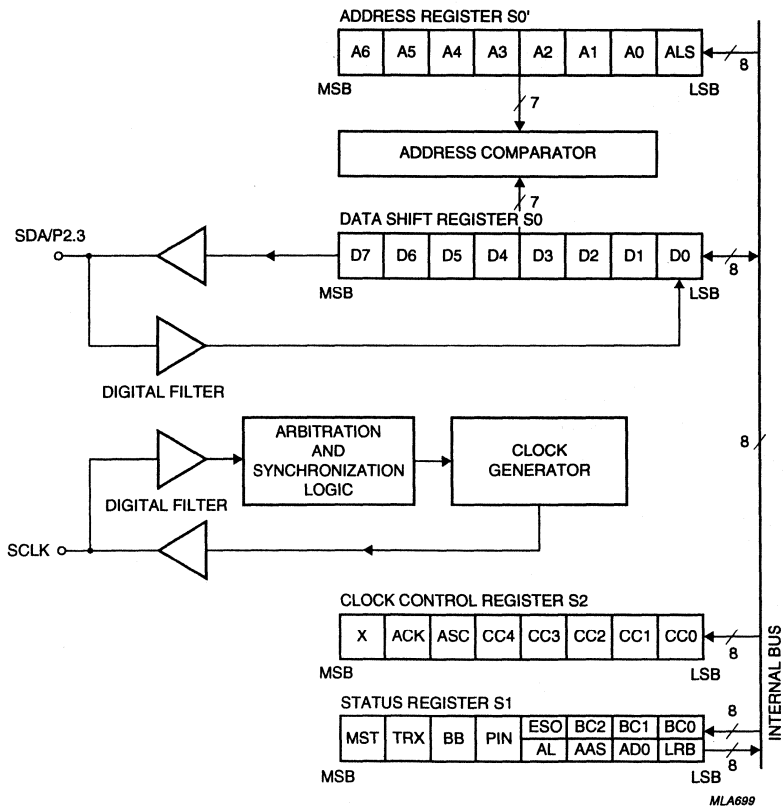



Fig.14 Block diagram of the serial I/O interface.

**Table 3** $f_{SCLK}$ as defined by Clock Control Register (S2).

| CC4 TO CC0 (HEX) | $f_{xtal}$ DIVISOR (DF) | $f_{SCLK}$ (kHz) at | | |
|---|---|---|---|---|
| | | $f_{xtal}$ = 3.58 MHz | $f_{xtal}$ = 10 MHz | $f_{xtal}$ = 16 MHz |
| 00 | forbidden | – | – | – |
| 01 | 39 | 91.8 | 256.4(1) | 410.3(1) |
| 02 | 45 | 79.5 | 222.2(1) | 355.6(1) |
| 03 | 51 | 70.2 | 196.1(1) | 313.7(1) |
| 04 | 63 | 56.8 | 158.7(1) | 254.0(1) |
| 05 | 75 | 47.7 | 133.3(1) | 213.3(1) |
| 06 | 87 | 41.1 | 114.9(1) | 183.9(1) |
| 07 | 99 | 36.2 | 101.0(1) | 161.6(1) |
| 08 | 123 | 29.1 | 81.3 | 130.1(1) |
| 09 | 147 | 4.4 | 68.0 | 108.8(1) |
| 0A | 171 | 20.9 | 58.5 | 93.6 |
| 0B | 195 | 18.4 | 51.3 | 82.1 |
| 0C | 243 | 14.7 | 41.2 | 65.8 |
| 0D | 291 | 12.3 | 34.4 | 55.0 |
| 0E | 339 | 10.6 | 29.5 | 47.2 |
| 0F | 387 | 9.2 | 25.8 | 41.3 |
| 10 | 483 | 7.4 | 20.7 | 33.1 |
| 11 | 579 | 6.2 | 17.3 | 27.6 |
| 12 | 675 | 5.3 | 14.8 | 23.7 |
| 13 | 771 | 4.6 | 13.0 | 20.8 |
| 14 | 963 | 3.7 | 10.4 | 16.6 |
| 15 | 1155 | 3.1 | 8.7 | 13.9 |
| 16 | 1347 | 2.7 | 7.4 | 11.9 |
| 17 | 1539 | 2.3 | 6.5 | 10.4 |
| 18 | 1923 | 1.9 | 5.2 | 8.3 |
| 19 | 2307 | 1.6 | 4.3 | 6.9 |
| 1A | 2691 | 1.3 | 3.7 | 5.9 |
| 1B | 3075 | 1.2 | 3.3 | 5.2 |
| 1C | 3843 | 0.9 | 2.6 | 4.2 |
| 1D | 4611 | 0.8 | 2.2 | 3.5 |
| 1E | 5379 | 0.7 | 1.9 | 3.0 |
| 1F | 6147 | 0.6 | 1.6 | 2.6 |

**Note**

1. Not permitted; maximum $f_{SCLK}$ = 100 kHz in $I^2C$-bus systems.

### 6.11.4 STATUS REGISTER (S1)

The Status Register controls the serial I/O interface and provides feedback concerning on-going bus transfers. Register S1 can be accessed by MOV A, S1, MOV S1, A and MOV S1, #data. The lower nibble of the Status Register is twofold: control bits BC0 to BC2 and ESO can only be written, whereas feedback bits LRB, AD0, AAS and AL can only be read. Table 4 describes the status bits.

The status bits interact in intricate ways with each other. This must be kept in mind when an $I^2C$-bus application is programmed.

#### *6.11.4.1 Master bit (MST) and Transmitter bit (TRX)*

MST and TRX together define the state of the serial I/O interface. When not engaged in a bus transfer MST and TRX should always be at zero, the slave/receiver state (see Fig.15). Return to this state is always performed by software. If the previous state was the master state, the transition (to slave/receiver by MOV1,#D8H) involves a stop condition which, as a consequence, clears both MST and TRX.

The transition to the master/transmitter state is also a programmed event. However, transitions to the master/receiver and the slave/transmitter states occur automatically if ALS = 0 (standard $I^2C$-bus protocol). A slave/receiver becomes a slave/transmitter if $R/\overline{W} = 1$ in its valid address (following the start condition). A master/transmitter becomes a master/receiver if $R/\overline{W} = 1$ in the transmitted address.

#### *6.11.4.2 Pending Interrupt Not bit (PIN)*

If MST = 1 or, if ALS = 1, PIN is set to zero after every byte transfer. Conversely, PIN becomes zero when a valid address is detected and after each byte of the following transfer. In addition, the serial interrupt request, PIN = 0 initiates 'clock synchronization', i.e. the SCLK line is pulled to $V_{SS}$ as long as PIN = 0. With this feature a slave may slow down a master, thus providing time to read the Data Register (in the case of a slave/receiver) or to write to the data register (in the case of a slave/transmitter). PIN is cancelled by an access to register S0 or by explicitly setting PIN to one.

If the SIO/derivative interrupt is disabled, the serial I/O interface may be serviced by testing PIN directly in user software.

#### *6.11.4.3 Bus Busy bit (BB)*

The Bus Busy bit (BB) is controlled by the serial I/O interface or by software in the bus master to generate the start and stop conditions. When a master clears BB (by MOV S1, #D8H), the serial I/O interface automatically clears MST and TRX, thereby returning to the slave/receiver state (see Fig.15). If BB = 1, write access to S1 is inhibited, except for the master or an addressed slave. Should BB be inadvertently set by excessive noise on the bus, the deadlock can be resolved by two consecutive MOV S1, #18H, the first of which just clears BB.

When a slave/transmitter detects an end of transmission (signalled by the lack of an acknowledgment from the master receiver), it has to access S1 in order to cancel PIN and to become slave/receiver. However, BB should remain set. This is reflected by MOV S1, #38H as illustrated in Fig.15. With PIN = 1, 'clock synchronization' terminates, enabling the master to generate the stop condition.

A start condition must only be generated when BB = 0; otherwise the serial I/O interface will respond as if bus arbitration has been lost (see Section 6.11.4.4).

#### *6.11.4.4 Arbitration Lost bit (AL)*

The AL bit is set by the serial I/O interface when it loses a bus arbitration in the master/transmitter mode. MST and TRX are cleared simultaneously to enable the interface, now in slave/receiver mode, to determine if it is validly addressed by the device that won the arbitration. PIN is activated when the byte transfer is complete. AL will be cleared when the serial interrupt is cancelled.

#### *6.11.4.5 Addressed As Slave bit (AAS)*

AAS is set by the serial I/O interface following a start condition when the valid address is detected (ALS = 0 in register S0') or when the first byte is received (ALS = 1 in register S0'). AAS is cleared when the serial interrupt is cancelled.

#### *6.11.4.6 Address Zero bit (AD0)*

AD0 is set, independently of ALS, by the serial I/O interface when byte 00H, the 'general call' address, is detected following a start condition. AD0 is cleared after a repeated start or a stop condition.

*6.11.4.7 Last Received Bit (LRB)*

LRB corresponds to the last bit transferred. If ACK = 1, LRB contains the acknowledgement bit. It remains valid as long as PIN = 0.

*6.11.4.8 Enable Serial I/O bit (ESO)*

When ESO = 0 access to register S0' is enabled. SCLK is in the high-impedance state and SDA/P2.3 is available as a normal port line.

When ESO = 1 the serial I/O interface and access to register S0 is enabled. Only when ESO = 1 may the other bits of register S1 be changed. SCLK and SDA/P2.3 are enabled as serial clock and data lines, respectively.

To avoid bus deadlock, ESO must be set to zero prior to the execution of the STOP instruction.

*6.11.4.9 Bit Counter bits (BC0, BC1 and BC2)*

The bit counter bits BC0, BC1 and BC2 should all be at zero for normal I²C-bus operation. The bit counter is always cleared by a start condition. Therefore, all eight bits of the first byte are transferred.

If a non-zero bit counter value is chosen, it is only valid for one register S0 transfer since the counter decrements to zero. An important use of the bit counter arises when a master/receiver signals an end of transmission by sending a negative acknowledge after the last byte received. To do this, the last byte is received with bit ACK = 0 in register S2. The negative acknowledge is then issued by setting the bit counter to one and 'receiving' one bit from the HIGH level available on the SDA line. The slave/transmitter interprets the same signals as a negative acknowledgement.

**Table 4** Overview of Status Register bits.

| BIT | NAME | TYPE | DESCRIPTION |
|---|---|---|---|
| MST | Master | R/W | MST = 0: slave (SCLK input).<br>MST = 1: master (SCLK output). |
| TRX | Transmitter | | TRX = 0: receiver (SDA/P2.3 input.<br>TRX = 1: transmitter (SDA/P2.3 output). |
| BB | Bus Busy | R/W | BB = 0: bus inactive (R)/generates stop condition (W).<br>BB = 1: bus busy (R)/generates start condition (W). |
| PIN | Pending Interrupt Not | R/W | PIN = 0: serial interrupt pending (after byte transfer, valid address or lost arbitration). SCLK line forced to $V_{SS}$.<br>PIN = 1: no serial interrupt pending. |
| ESO | Enable Serial Output | W | ESO = 0: serial I/O interface disabled/write access to S0' possible.<br>ESO = 1: serial I/O interface enabled write access to S0 possible. |
| BC0 to BC2 | Bit Counter 0 to 2 | W | 3-bit binary value of 0 to 7, counting down the number of bits transferred (0 used for complete byte). |
| AL | Arbitration Lost | R | Set: when a bus conflict is lost.<br>Reset: when corresponding serial interrupt (PIN) is cancelled. |
| AAS | Addressed As Slave | R | Set: following a start condition if valid address is detected (ALS = 0) or if first byte is received (ALS = 1).<br>Reset: when corresponding serial interrupt (PIN) is cancelled. |
| AD0 | Address zero | R | Set: following a start condition if byte 00H ('general call' address) is detected.<br>Reset: after a repeated start or a stop condition. |
| LRB | Last Received Bit | R | Set or Reset depending on the value of the last bit transferred, acknowledgement bit if ACK = 1. |

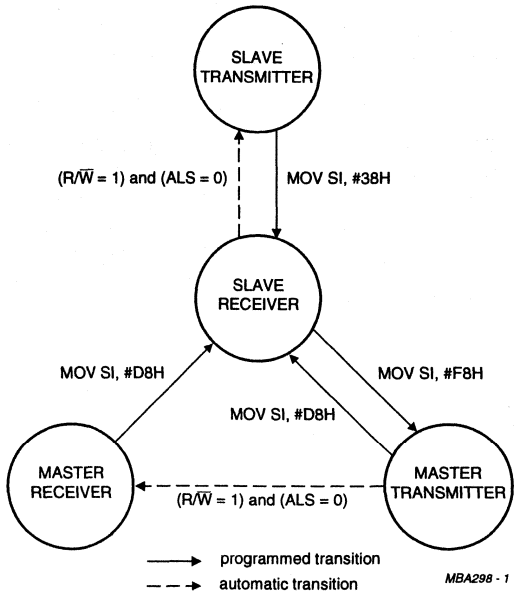



Fig.15 State diagram of the serial I/O interface.

### 6.12 Timing

Every machine cycle consists of 10 time slots which are again subdivided into 3 clock periods each (see Fig.16).

Permitted clock frequencies range from 1 MHz to a maximum, which is a function of the supply voltage. At $V_{DD} \geq 4.5$ V, a 16 MHz maximum clock frequency is guaranteed.

The clock signal may be internally generated by an on-chip oscillator. Alternatively, an external clock may be applied to pin XTAL1. In this configuration, a short circuit with an internal pull-up transistor on XTAL1 may occur while the oscillator is inhibited (see Section 6.13). Care should be taken to avoid excessive current flow.

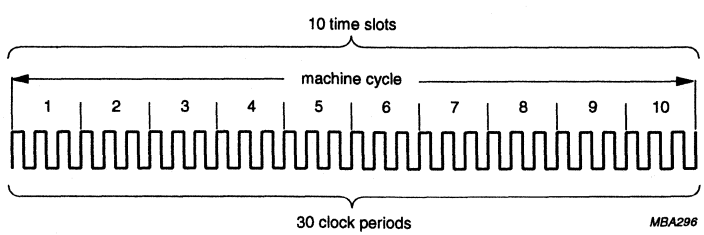



Fig.16 Timing of a machine cycle.

### 6.13 Oscillator

The on-chip oscillator basically consists of an inverter stage which includes a feedback resistor and load capacitors (see Fig.17). In most applications, a quartz crystal will be connected between XTAL1 and XTAL2. Alternatively, a ceramic resonator or an inductor may be used as a timing element.

When the supply voltage drops below the power-on reference level, the oscillator is inhibited. The internal oscillator can also be inhibited by the STOP instruction under software control (see Section 6.16).

The transconductance ($g_m$) of the inverter stage can be mask-programmed, thereby optimizing the oscillator for a specific frequency and resonator. Three standard transconductance options, referred to as LOW, MEDIUM and HIGH, can be specified by the user. Table 6 is intended as a rough selection guide for typical quartz and PXE resonators.

With $C_1 = C_2 = 10$ pF on-chip, external capacitors are not required for quartz oscillators. However, for adequate frequency stability, PXE resonators need external capacitors in the order of the static resonator capacitance $C_0$, such as external $C_1 = C_2 = 30$ to 100 pF.

Oscillator start-up time depends mainly on the external timing element. The start-up time of a quartz crystal is several milliseconds because of the narrow crystal bandwidth. For proper oscillator start-up, the transconductance ($g_m$) of the inverter stage must fulfil relationship (1); shown below.

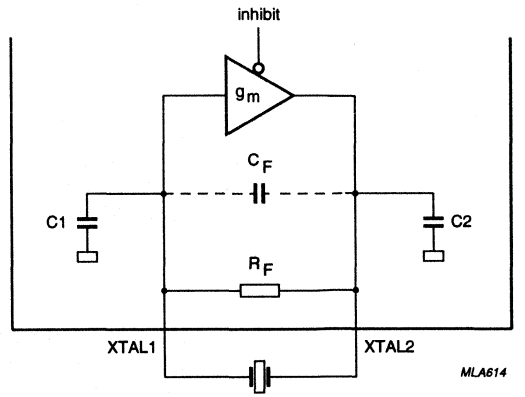



Fig.17 Oscillator with integrated elements.

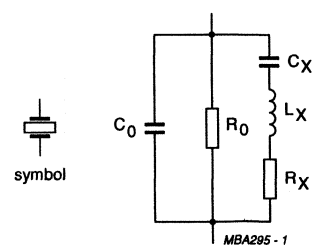



Fig.18 Crystal unit equivalent circuit.

**Table 5** Notation to relationship (see Figs 17 and 18).

| SYMBOL | DEFINITION |
|---|---|
| $R_X$ | resonator series resistance |
| $C_0$ | static resonator capacitance |
| $R_0$ | resonator loss resistance |
| $R_P$ | $R_0$ // $R_F$ |
| $R_F$ | feedback resistor |
| $C_L$ | C1 × C2/(C1 + C2) (load capacitance) |
| $C_F$ | parasitic feedback capacitance (typically 2 pF on-chip, external value depends on printed-circuit board wiring) |
| $\omega$ | $2\pi f_{osc}$ |

$$4.2\left[R_X\omega^2(C_L+C_0+C_F)^2+\frac{1}{R_P}\right] < g_m < \frac{C1\times C2}{\left[R_X(C_0+C_F)^2+\frac{1}{\omega^2 R_P}\right]} \quad (1)$$

**Table 6** Recommended transconductance options for popular quartz and PXE resonators.

| OPTION | TYP. $g_m$ at 5 V (mS) | $f_{osc}$ FOR QUARTZ (MHz) | $f_{osc}$ FOR PXE (MHz) |
|---|---|---|---|
| LOW ($g_{mL}$) | 0.4 | 1 to 6 | 1 to 2.4 |
| MEDIUM ($g_{mM}$) | 1.6 | 4 to 12 | 1 to 6 |
| HIGH ($g_{mH}$) | 4.5 | 10 to 16 | 3 to 16 |

### 6.14 Reset

To ensure proper start-up, the microcontroller must be initialized to a defined starting condition. The device executes the first instruction 1866 clock cycles after the falling edge of the internal reset.

#### 6.14.1 PASSIVE RESET

A passive reset is generated by the RC circuit illustrated in Fig.19. While $V_{DD}$ rises, the discharged $C_{reset}$ keeps the RESET pin near the $V_{DD}$ level. When $V_{DD}$ crosses the power-on reference level ($V_{ref}$) the power-on reset circuit generates a reset pulse of approximately 50 µs. This pulse is without effect since it feeds into the reset signal forced by the one on the RESET pin. The $f_{xtal}$ dependent minimum $V_{DD}$ must be reached before the voltage on RESET drops below $V_{IH} = 0.7V_{DD}$. This translates into a lower bound for $C_{reset}R_{reset}$ equal to twice the rise time of $V_{DD}$ (for linearly rising $V_{DD}$) or eight times the time constant of $V_{DD}$ (for exponentially rising $V_{DD}$). The internal diode rapidly discharges $C_{reset}$ when $V_{DD}$ falls off, ensuring reliable reset even after short interruptions of supply voltage. To avoid overload of the internal diode, an external diode should be added in parallel if $C_{reset} > 2.2$ µF.

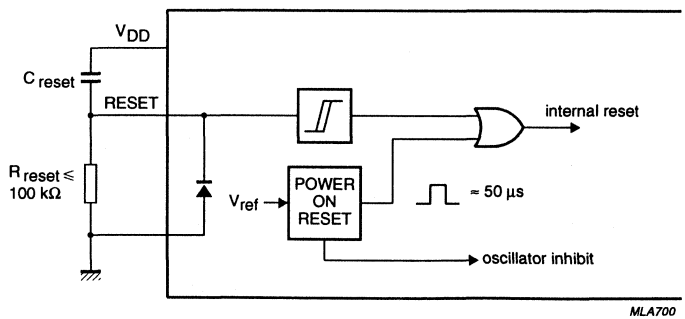



Fig.19 Passive power-on reset.

### 6.14.2 Internal reset

In systems where $V_{DD}$ reaches its $f_{xtal}$ dependent minimum operating value before the clock $f_{xtal}$ is applied, reset can be performed without external components. This condition is generally fulfilled with quartz and PXE resonators since oscillator start-up takes several milliseconds. Besides, rapid power-up is usually available in battery-powered systems.

If the internal power-on reset is used the RESET pin should be connected to $V_{SS}$. When $V_{DD}$ increases above the power-on reference level $V_{ref}$, the power-on-reset circuit generates a reset pulse of approximately 50 µs. This pulse guarantees proper initialization under the conditions defined above.

The power-on reference level $V_{ref}$ is a mask option. The user can select a reference voltage between 1.2 V and 3.6 V in discrete steps of 100 mV. The accuracy of the reference voltage is ±500 mV for the $V_{ref}$ range 1.2 V to 3.0 V and ±800 mV for the $V_{ref}$ range 3.1 V to 3.6 V. The chosen $V_{ref}$ should have sufficient margin regarding the minimum intended $V_{DD}$.

A mask option without an internal power-on reset circuit is also available. It is recommended if the user does not intend to use the internal power-on-reset circuit. In this case, the supply current requirements in Stop mode (see Section 6.16) will reduce to the level of leakage currents, i.e. virtually zero at ambient temperature.

### 6.14.3 Active reset

An active reset can be generated by driving the RESET pin HIGH from an external logic device. Such an active reset pulse should not fall off before $V_{DD}$ has reached its $f_{xtal}$ dependent minimum operating value.

### 6.14.4 Reset state

After a reset, the device state is characterized as follows:

- Program Counter 0
- Memory bank 0
- Register bank 0 - Stack Pointer 0 (location pair 8 and 9)
- All interrupts disabled
- Timer/event counter 1 stopped and cleared
- Timer prescaler modulo-32 (PS = 0)
- Timer flag cleared
- All port flip-flops (except SDA/P2.3) set to 1 (set option) or 0 (reset option) as selected by the user
- SDA/P2.3 is high-impedance with the port flip-flop set to 1
- SCLK is high-impedance
- Serial I/O interface disabled (ESO = 0) and in slave/receiver mode (S0, S0', S1 and S2 cleared except for PIN = 1)
- Idle and Stop modes cancelled.

### 6.15 Idle mode

The Idle mode is very useful in low-power applications. When all computational tasks are completed, the device can be put into standby instead of into a busy waiting loop. Nevertheless, the device is on the alert and ready to respond rapidly to any interrupt.

The microcontroller enters the Idle mode via the IDLE instruction. In the Idle mode, all activity is halted except for the oscillator, the timer/event counter 1 and the serial I/O interface (if available).

The microcontroller leaves the Idle mode when an enabled interrupt occurs. The interrupt routine is executed before operation resumes with the instruction following the IDLE opcode.

For timer/event counter interrupts and SIO/derivative interrupts, termination of the Idle mode is straightforward. However, care must be taken when the Idle mode is left by the external interrupt since $\overline{INT}$/T0 is negative-edge responding. If $\overline{INT}$/T0 was LOW prior to entering the Idle mode, it must be taken HIGH before the negative edge can be generated. Figure 20 specifies the exact timing for leaving the Idle mode via the external interrupt $\overline{INT}$/T0.

If no interrupt is enabled, the Idle mode can only be terminated by an active signal on the RESET pin. A normal reset sequence is executed (see Fig.20).

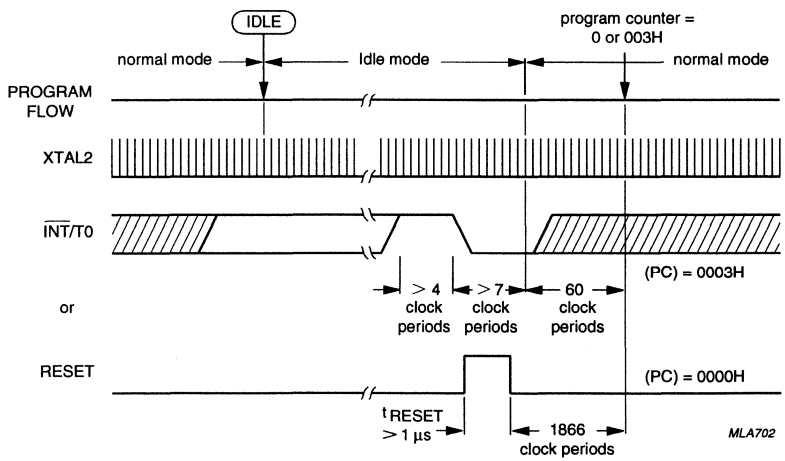



Fig.20 Entering and leaving the Idle mode.

### 6.16 Stop mode

The Stop mode allows very low-power applications. When all computational tasks are completed, the device can be almost completely shut off by stopping its oscillator. In contrast to the Idle mode, the device is not ready to respond to any interrupt with little latency.

The microcontroller enters the Stop mode via the STOP instruction. The oscillator is switched off. All internal states and I/O levels are maintained.

The microcontroller leaves the Stop mode by a LOW level on $\overline{INT}$/T0 or a reset. In the latter case, a normal reset sequence is executed (see Fig.21).

In contrast to the Idle mode and the external interrupt mechanism, the microcontroller responds to a LOW level on $\overline{INT}$/T0 rather than to a negative edge. If $\overline{INT}$/T0 is LOW when the STOP instruction is executed, the Stop mode will not be entered.

A negative edge on $\overline{INT}$/T0 continues program execution after a 1866 clock cycle delay, which ensures proper oscillator start-up. If the external interrupt is enabled, the device executes the instruction following the STOP opcode before diverting to the interrupt routine. If the external interrupt is disabled, program execution continues with the instructions following the STOP opcode (see Fig.21).

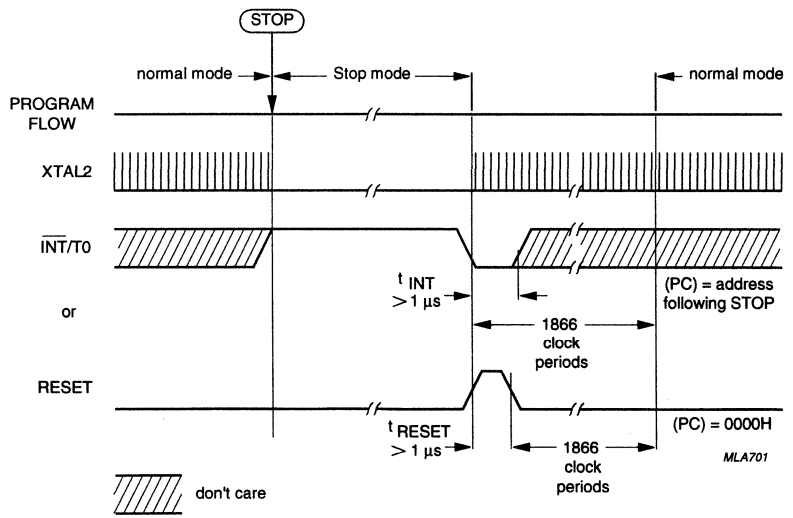



Fig.21 Entering and leaving the Stop mode.

### 6.17 Derivative logic

Derivative logic is provided with many members of the PCF84CXXXA family. The detailed description of the derivative circuitry is given in the data sheet of the specific device. In this section, the shared principles of derivative logic are briefly reviewed.

Derivative registers are accessed over the internal bus. The derivative registers are write-only, read-only or read/write (see Fig.22). They are addressed through the derivative address register when the derivative input/output instructions (MOV A, Dx; MOV Dx, A; ANL Dx, A and ORL Dx, A) are executed.

Derivative interrupts share the line PIN with the SIO interrupt (if available). When the derivative interrupt routine is executed, the PIN line must be de-activated by software.

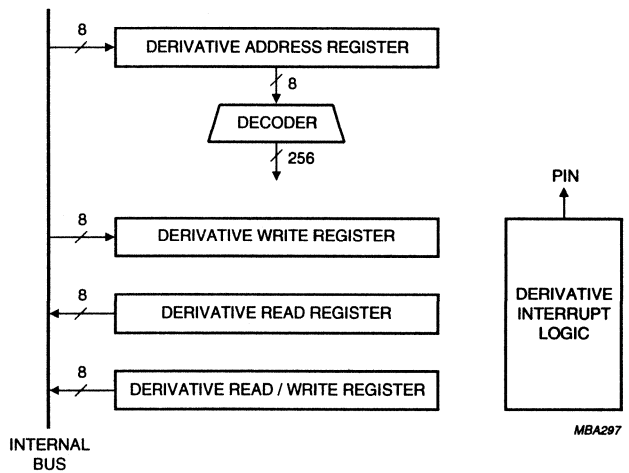



Fig.22 Block diagram of derivative logic.

**Table 7** Summary of mask options.

| FEATURE | OPTION | DESCRIPTION |
|---|---|---|
| ROM | any mix of instructions | program; size restricted by ROM size (see Tables 8 and 9) |
| Ports | option 1 | standard output (see Fig.11) |
| | option 2 | open drain output (see Fig.12) |
| | option 3 | push-pull output (see Fig.13) |
| | set | flip-flop at logic 1 after reset |
| | reset | flip-flop at logic 0 after reset |
| Oscillator | $g_{mL}$ | LOW transconductance (see Table 6) |
| | $g_{mM}$ | MEDIUM transconductance (see Table 6) |
| | $g_{mH}$ | HIGH transconductance (see Table 6) |

## 7 INSTRUCTION SET

The PCF84CXXXA instruction set consists of over 100 one and two-byte instructions. Program code efficiency is high because all RAM locations and all ROM locations on a 256-byte page require only a single-byte address. Table 9 lists the symbols that are used in Table 8 and the Instruction map is shown in Section 7.1.

**Table 8** PCF84CXXX family instruction set.

| MNEMONIC | OPCODE (HEX) | BYTES/ CYCLES | DESCRIPTION | FUNCTION | NOTES |
|---|---|---|---|---|---|
| **ACCUMULATOR** | | | | | |
| ADD A, Rr[1] | 6<8 + r> | 1/1 | Add register contents to A | (A)←(A) + (Rr) | r = 0 to 7 |
| ADD A, @Rr[1] | 6r | 1/1 | Add RAM data, addressed by Rr, to A | (A)←(A) + ((Rr)) | r = 0, 1 |
| ADD A, #data[1] | 03 data | 2/2 | Add immediate data to A | (A)←(A) + data | |
| ADDC A, Rr[1] | 7<8 + r> | 1/1 | Add carry and register contents to A | (A)←(A) + (Rr) + (C) | r = 0 to 7 |
| ADDC A, @Rr[1] | 7r | 1/1 | Add carry and RAM data, addressed by Rr, to A | (A)←(A) + ((Rr)) + (C) | r = 0, 1 |
| ADDC A, #data[1] | 13 data | 2/2 | Add carry and immediate data to A | (A)←(A) + data + (C) | |
| ANL A, Rr | 5<8 + r> | 1/1 | AND Rr with A | (A)←(A) AND (Rr) | r = 0 to 7 |
| ANL A, @Rr | 5r | 1/1 | AND RAM data addressed by Rr, with A | (A)←(A) AND ((Rr)) | r = 0, 1 |
| ANL A, #data | 53 data | 2/2 | AND immediate data with A | (A)←(A) AND data | |
| ORL A, Rr | 4<8 + r> | 1/1 | OR Rr with A | (A)←(A) OR (Rr) | r = 0 to 7 |
| ORL A, @Rr | 4r | 1/1 | OR RAM data, addressed by Rr, with A | (A)←(A) OR ((Rr)) | r = 0, 1 |
| ORL A, #data | 43 data | 2/2 | OR immediate data with A | (A)←(A) OR data) | |
| XRL A, Rr | D<8 +r> | 1/1 | XOR Rr with A | (A)←(A) XOR (Rr) | r = 0 to 7 |
| XRL A, @Rr | Dr | 1/1 | XOR RAM data, addressed by Rr, with A | (A)←(A) XOR ((Rr)) | r = 0, 1 |
| XRL A, #data | D3 data | 2/2 | XOR immediate data with A | (A)←(A) XOR data) | |
| INC A | 17 | 1/1 | Increment A by 1 | (A)←(A) + 1 | |
| DEC A | 07 | 1/1 | Decrement A by 1 | (A)←(A) – 1 | |
| CLR A | 27 | 1/1 | Clear A to zero | (A)←0 | |
| CPL A | 37 | 1/1 | One's complement A | (A)←NOT(A) | |
| RL A | E7 | 1/1 | Rotate A left | $(A_{n+1})\leftarrow(A_n)$, $(A_0)\leftarrow(A_7)$ | n = 0 to 6 |
| RLC A[2] | F7 | 1/1 | Rotate A left through carry | $(A_{n+1})\leftarrow(A_n)$, $(A_0)\leftarrow(C)$, $(C)\leftarrow(A_7)$ | n = 0 to 6 |
| RR A | 77 | 1/1 | Rotate A right | $(A_n)\leftarrow(A_{n+1})$, $(A_7)\leftarrow(A_0)$ | n = 0 to 6 |
| RRC A[2] | 67 | 1/1 | Rotate A right through carry | $(A_n)\leftarrow(A_{n+1})$, $(A_7)\leftarrow(C)$, $(C)\leftarrow(A_0)$ | n = 0 to 6 |
| DA A[2] | 57 | 1/1 | Decimal adjust A | (A)←(A) + 06H if AC = 1 or $(A_{0-3})>9$; (A)←(A) + 60H if $(A_{4-7})>9$ | |
| SWAP A[2] | 47 | 1/1 | Swap nibbles of A | $(A_{4-7})\leftrightarrow(A_{0-3})$ | |

| MNEMONIC | OPCODE (HEX) | BYTES/ CYCLES | DESCRIPTION | FUNCTION | NOTES |
|---|---|---|---|---|---|
| **DATA MOVES** | | | | | |
| MOV A, Rr | F<8 + r> | 1/1 | Move register contents to A | (A)←(Rr) | r = 0 to 7 |
| MOV A, @Rr | Fr | 1/1 | Move RAM data addressed by Rr, to A | (A)←((Rr)) | r = 0, 1 |
| MOV A, #data | 23 data | 2/2 | Move immediate data to A | (A)←data | |
| MOV Rr, A | A<8 + r> | 1/1 | Move Accumulator contents to register | (Rr)←(A) | r = 0 to 7 |
| MOV@Rr, A | Ar | 1/1 | Move Accumulator contents to RAM location addressed by Rr | ((Rr))←(A) | r = 0, 1 |
| MOV Rr, #data | B<8 + r> data | 2/2 | Move immediate data to Rr | (Rr)←data | r = 0 to 7 |
| MOV @Rr, #data | Br data | 2/2 | Move immediate data to RAM location addressed by Rr | ((R0))←data | r = 0, 1 |
| XCH A, Rr | 2<8 + r> | 1/1 | Exchange A contents with Rr | (A)↔(Rr) | r = 0 to 7 |
| XCH A, @Rr | 2r | 1/1 | Exchange Accumulator contents with RAM data addressed by Rr | (A)↔((Rr)) | r = 0, 1 |
| XCHD A, @Rr | 3r | 1/1 | Exchange lower nibbles of A and RAM data addressed by Rr | $(A_{0-3})\leftrightarrow((Rr_{0-3}))$ | r = 0, 1 |
| MOV A, PSW | C7 | 1/1 | Move PSW contents to Accumulator | (A)←(PSW) | |
| MOV PSW, A[3] | D7 | 1/1 | Move Accumulator bit 3 to $PSW_3$ (PS) | $(PS)\leftarrow(A_3)$ | |
| MOV P A, @A | A3 | 1/2 | Move indirectly addressed data in current page to A | $(PC_{0-7})\leftarrow(A)$, (A)←((PC)) | |
| **CARRY FLAG** | | | | | |
| CLR C[2] | 97 | 1/1 | Clear carry bit | (C)←0 | |
| CPL C[2] | A7 | 1/1 | Complement carry bit | (C)←NOT(C) | |
| **REGISTER** | | | | | |
| INC Rr | 1<8 + r> | 1/1 | Increment register by 1 | (Rr)←(Rr) + 1 | r = 0 to 7 |
| INC @Rr | 1r | 1/1 | Increment RAM data, addressed by Rr, by 1 | ((Rr))←((Rr)) + 1 | r = 0,1 |
| DEC Rr | C<8 + r> | 1/1 | Decrement register by 1 | (Rr)←(Rr) – 1 | r = 0 to 7 |
| DEC @Rr | Cr | 1/1 | Decrement RAM data addressed by Rr, by 1 | ((Rr))←((Rr)) – 1 | r = 0, 1 |

| MNEMONIC | OPCODE (HEX) | BYTES/ CYCLES | DESCRIPTION | FUNCTION | NOTES |
|---|---|---|---|---|---|
| **BRANCH** | | | | | |
| JMP addr | <2n>4 addr | 2/2 | Unconditional jump within a 2 kbyte bank | ($PC_{8-10}$)←n ($PC_{0-7}$)←addr ($PC_{11-12}$)← (MBFF0-1) | n = 0 to 7 |
| JMPP @A | B3 | 1/2 | Indirect jump within a page | ($PC_{0-7}$)←((A)) | |
| DJZN Rr, addr | E<8 + r> addr | 2/2 | Decrement Rr by 1 and jump if not zero to addr | (Rr)←(Rr) – 1; if (Rr) not zero, then ($PC_{0-7}$)←addr | r = 0 to 7 |
| DJNZ @Rr, addr | Er | 2/2 | Decrement RAM data, addressed by Rr, by 1 and jump if not zero to addr | ((Rr))←((Rr)) – 1; if ((Rr)) not zero, then ($PC_{0-7}$)←addr | r = 0 to 1 |
| JBb addr | <2b + 1> 2 addr | 2/2 | Jump to addr if Accumulator bit b = 1 | If ($A_b$) = 1, then ($PC_{0-7}$)←addr | b = 0 to 7 |
| JC addr | F6 addr | 2/2 | Jump to addr if C = 1 | If (C) = 1, then ($PC_{0-7}$)←addr | |
| JNC addr | E6 addr | 2/2 | Jump to addr if C = 0 | If (C) = 0, then ($PC_{0-7}$)←addr | |
| JZ addr | C6 addr | 2/2 | Jump to addr if A = 0 | If (A) = 0, ($PC_{0-7}$)←addr | |
| JNZ addr | 96 addr | 2/2 | Jump to addr if A is NOT zero | If (A) ≠ 0, then ($PC_{0-7}$)←addr | |
| JT0 addr | 36 addr | 2/2 | Jump to addr if T0 = 1 | If T0 = 1, then ($PC_{0-7}$)←addr | |
| JNT0 addr | 26 addr | 2/2 | Jump to addr if T0 = 0 | If T0 = 0, then ($PC_{0-7}$)←addr | |
| JT1 addr | 56 addr | 2/2 | Jump to addr if T1 = 1 | If T1 = 1, then ($PC_{0-7}$)←addr | |
| JNT1 addr | 46 addr | 2/2 | Jump to addr if T1 = 0 | If T0 = 0, then ($PC_{0-7}$)←addr | |
| JTF addr[4] | 16 addr | 2/2 | Jump to addr if Timer Flag = 1 | If TF = 1, then ($PC_{0-7}$)←addr | |
| JNTF addr[4] | 06 addr | 2/2 | Jump to addr if Timer Flag = 0 | If T0 = 0, then ($PC_{0-7}$)←addr | |

| MNEMONIC | OPCODE (HEX) | BYTES/ CYCLES | DESCRIPTION | FUNCTION | NOTES |
|---|---|---|---|---|---|
| **TIMER/EVENT COUNTER** | | | | | |
| MOV A,T | 42 | 1/1 | Move timer/event counter contents to A | (A)←(T) | |
| MOV T, A | 62 | 1/1 | Move A contents to timer/event counter | (T)←(A) | |
| STRT CNT | 45 | 1/1 | Start event counter | | |
| STRT T | 55 | 1/1 | Start timer | | |
| STOP TCNT | 65 | 1/1 | Stop timer/event counter | | |
| EN TCNTI | 25 | 1/1 | Enable timer/event counter interrupt | | |
| DIS TCNTI | 35 | 1/1 | Disable timer/event counter interrupt | | |
| **CONTROL** | | | | | |
| EN I | 05 | 1/1 | Enable external (chip enable) interrupt | | |
| DIS I | 15 | 1/1 | Disable external (chip enable) interrupt | | |
| SEL RB0[5] | C5 | 1/1 | Select register bank 0 | (RBS)←0 | |
| SEL RB1[5] | D5 | 1/1 | Select register bank 1 | (RBS)←1 | |
| SEL MB0[10] | E5 | 1/1 | Select program memory bank 0 | (MBFF0)←0, (MBFF1)←0 | |
| SEL MB1[10] | F5 | 1/1 | Select program memory bank 1 | (MBFF0)←1, (MBFF1)←0 | |
| SEL MB2[10] | A5 | 1/1 | Select program memory bank 2 | (MBFF0)←0, (MBFF1)←1 | |
| SEL MB3[10] | B5 | 1/1 | Select program memory bank 3 | (MBFF0)←1, (MBFF1)←1 | |
| STOP | 22 | 1/1 | Enter Stop mode | | |
| IDLE | 01 | 1/1 | Enter Idle mode | | |
| **SUBROUTINE** | | | | | |
| CALL addr[6] | <2n + 1> 4addr | 2/2 | Jump to subroutine | ((SP))←(PC) ($PSW_{4,6,7}$), (SP)←(SP) + 1, ($PC_{8-10}$)←n, ($PC_{0-7}$)←addr, ($PC_{11-12}$) ←(MBFF0-1) | n = 0 to 7 |
| RET[6] | 83 | 1/2 | Return from subroutine | (SP)←(SP) – 1, (PC)←((SP)) | |
| RETR[6] | 93 | 1/2 | Return from interrupt and restore bits 4, 6 and 7 of PSW | (SP)←(SP) – 1, ($PSW_{4,6,7}$) + (PC)←((SP)) | |

| MNEMONIC | OPCODE (HEX) | BYTES/ CYCLES | DESCRIPTION | FUNCTION | NOTES |
|---|---|---|---|---|---|
| **PARALLEL INPUT/OUTPUT** | | | | | |
| IN A, P0 | 08 | 1/2 | Input Port 0 data to Accumulator | (A)←(P0) | |
| IN A, P1 | 09 | 1/2 | Input Port 1 data to Accumulator | (A)←(P1) | |
| IN A, P2[7] | 0A | 1/2 | Input Port 2 data to Accumulator | (A)←(P2) | |
| OUTL P0, A | 38 | 1/2 | Output A data to Port 0 | (P0)←(A) | |
| OUTL P1, A | 39 | 1/2 | Output A data to Port 1 | (P1)←(A) | |
| OUTL P2, A | 3A | 1/2 | Output A data to Port 2 | (P2)←(A) | |
| ANL P0, #data | 98 data | 2/2 | AND Port 0 data with immediate data | (P0)←(P0) AND data | |
| ANL P1, #data | 99 data | 2/2 | AND Port 1 data with immediate data | (P1)←(P1) AND data | |
| ANL P2, #data | 9A data | 2/2 | AND Port 2 data with immediate data | (P2)←(P2) AND data | |
| ORL P0, #data | 88 data | 2/2 | OR Port 0 data with immediate data | (P0)←(P0) OR data | |
| ORL P1, #data | 89 data | 2/2 | OR Port 1 data with immediate data | (P1)←(P1) OR data | |
| ORL P2, #data | 8A data | 2/2 | OR Port 2 data with immediate data | (P2)←(P2) OR data | |
| **DERIVATIVE INPUT/OUTPUT** | | | | | |
| MOV A, Dx[8] | 8C direct | 2/2 | Move derivative register contents to A | (A)←(Dx) | x = 0 to 255 |
| MOV Dx, A[8] | 8D direct | 2/2 | Move A contents to derivative register | (Dx)←(A) | x = 0 to 255 |
| ANL Dx, A[8] | 8E direct | 2/2 | AND derivative register with A | (Dx)←(Dx) AND (A) | x = 0 to 255 |
| ORL Dx, A[8] | 8F direct | 2/2 | OR derivative register with A | (Dx)←(Dx) OR (A) | x = 0 to 255 |
| **SERIAL INPUT/OUTPUT** | | | | | |
| MOV A, S0 | 0C | 1/2 | Move serial I/O register 0 contents to A | (A)←(S0) | |
| MOV A, S1[9] | 0D | 1/2 | Move serial I/O register 1 contents to A | (A)←(S1) | |
| MOV S0, A | 3C | 1/2 | Move A contents to serial I/O register 0 | (S0)←(A) | |
| MOV S1, A[9] | 3D | 1/2 | Move A contents to serial I/O register 1 | (S1)←(A) | |
| MOV S2, A | 3E | 1/2 | Move A contents to serial I/O register 2 | (S2)←(A) | |
| MOV S0, #data | 9C data | 2/2 | Move immediate data to serial I/O register 0 | (S0)←data | |
| MOV S1, #data[9] | 9D data | 2/2 | Move immediate data to serial I/O register 1 | (S1)←data | |
| MOV S2, #data | 9E data | 2/2 | Move immediate data to serial I/O register 2 | (S2)←data | |
| EN SI | 85 | 1/1 | Enable serial I/O interrupt | | |
| DIS SI | 95 | 1/1 | Disable serial I/O interrupt | | |
| NOP | 00 | 1/1 | No operation | $(PC_{0-10})$← $(PC_{0-10})$ + 1 | |

**Notes to Table 8**

1. PSW CY, AC affected.
2. PSW CY affected.
3. PSW PS affected.
4. Execution of a JTF or JNTF instruction resets the Timer Flag (TF).
5. PSW RBS affected.
6. PSW $SP_0$, $SP_1$ and $SP_2$, affected.
7. (A) = 0000, P2.3, P2.2, P2.1 and P2.0.
8. For more information on the derivative I/O instructions of a particular microcontroller, consult the specific microcontroller data sheet.
9. (S1) has a different meaning for read and write operations. See Section 6.11.4.
10. SEL MB instructions may not be used within interrupt routines.

**Table 9** Definitions of symbols used in Table 8.

| SYMBOL | DESCRIPTION |
|---|---|
| A | Accumulator |
| AC | auxiliary (half) carry |
| addr | program memory address |
| Bb | bit designation (b = 0 to 7) |
| CE/T0 | CE/T0 input |
| CY | carry bit |
| Dx | mnemonic derivative register |
| data | 8-bit number or expression |
| MB0 | program memory bank 0 |
| MB1 | program memory bank 1 |
| MB2 | program memory bank 2 |
| MB3 | program memory bank 3 |
| MBFF0 | memory bank flip-flop 0 |
| MBFF1 | memory bank flip-flop 1 |
| PC | Program Counter |
| PS | timer prescaler select |
| PSW | Program Status Word |
| RB0 | register bank 0 |
| RB1 | register bank 1 |
| RBS | register bank select |
| Rr | register designation (r = 0 to 7) |
| SPn | Stack Pointer (n = 0, 1 or 2) |
| T | Timer 1 |
| T1 | T1 input |
| TF | Timer Flag |
| x | derivative register address (x = 0 to 255) |
| (X) | contents of X |
| ((X)) | contents of location addressed by X |
| ← | is replaced by |
| ↔ | is exchanged with |
| # | immediate data prefix |
| @ | indirect address prefix |
| * | hexadecimal; 8...F selects R0...R7 |
| & | hexadecimal; 0, 2, 4, 6, 8, A, C, E<br>selects page 0...7 in JMP,<br>i.e. $(PC_{8-10}) \leftarrow \&_{1-3}$ |
| % | hexadecimal; 1, 3, 5, 7, 9, B, D, F<br>selects page 0...7 in CALL,<br>i.e. $(PC_{8-10}) \leftarrow \&_{1-3}$<br>selects bit b = 0...7 in JBb,<br>i.e. $b = \&_{1-3}$ |

### 7.1 Instruction map

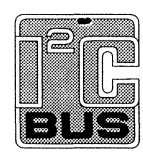
first hexadecimal character of opcode — second hexadecimal character of opcode

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | NOP | IDLE | | ADD A, #data | JMP page 0 | EN I | JNTF addr | DEC A | IN A,Pp 0 | IN A,Pp 1 | IN A,Pp 2 | | MOV A,Sn 0 | MOV A,Sn 1 | | |
| 1 | INC @ Rr 0 | INC @ Rr 1 | JB0 addr | ADDC A,#data | CALL page 0 | DIS I | JTF addr | INC A | INC Rr 0 | INC Rr 1 | INC Rr 2 | INC Rr 3 | INC Rr 4 | INC Rr 5 | INC Rr 6 | INC Rr 7 |
| 2 | XCH A, @Rr 0 | XCH A, @Rr 1 | STOP | MOV A, #data | JMP page 1 | EN TCNTI | JNT0 addr | CLR A | XCH A,Rr 0 | XCH A,Rr 1 | XCH A,Rr 2 | XCH A,Rr 3 | XCH A,Rr 4 | XCH A,Rr 5 | XCH A,Rr 6 | XCH A,Rr 7 |
| 3 | XCHD A, @Rr 0 | XCHD A, @Rr 1 | JB1 addr | | CALL page 1 | DIS TCNTI | JT0 addr | CPL A | OUTL Pp,A 0 | OUTL Pp,A 1 | OUTL Pp,A 2 | | MOV Sn,A 0 | MOV Sn,A 1 | MOV Sn,A 2 | |
| 4 | ORL A, @Rr 0 | ORL A, @Rr 1 | MOV A, T | ORL A, #data | JMP page 2 | STRT CNT | JNT1 addr | SWAP A | ORL A,Rr 0 | ORL A,Rr 1 | ORL A,Rr 2 | ORL A,Rr 3 | ORL A,Rr 4 | ORL A,Rr 5 | ORL A,Rr 6 | ORL A,Rr 7 |
| 5 | ANL A, @Rr 0 | ANL A, @Rr 1 | JB2 addr | ANL A, #data | CALL page 2 | STRT T | JT1 addr | DA A | ANL A,Rr 0 | ANL A,Rr 1 | ANL A,Rr 2 | ANL A,Rr 3 | ANL A,Rr 4 | ANL A,Rr 5 | ANL A,Rr 6 | ANL A,Rr 7 |
| 6 | ADD A, @Rr 0 | ADD A, @Rr 1 | MOV T, A | | JMP page 3 | STOP TCNT | | RRC A | ADD A,Rr 0 | ADD A,Rr 1 | ADD A,Rr 2 | ADD A,Rr 3 | ADD A,Rr 4 | ADD A,Rr 5 | ADD A,Rr 6 | ADD A,Rr 7 |
| 7 | ADDC A, @Rr 0 | ADDC A, @Rr 1 | JB3 addr | | CALL page 3 | | | RR A | ADDC A,Rr 0 | ADDC A,Rr 1 | ADDC A,Rr 2 | ADDC A,Rr 3 | ADDC A,Rr 4 | ADDC A,Rr 5 | ADDC A,Rr 6 | ADDC A,Rr 7 |
| 8 | | | | RET | JMP page 4 | EN SI | | | ORL Pp,#data 0 | ORL Pp,#data 1 | ORL Pp,#data 2 | | MOV A,Dx | MOV Dx,A | ANL Dx,A | ORL Dx,A |
| 9 | | | JB4 addr | RETR | CALL page 4 | DIS SI | JNZ addr | CLR C | ANL Pp,#data 0 | ANL Pp,#data 1 | ANL Pp,#data 2 | | MOV Sn,#data 0 | MOV Sn,#data 1 | MOV Sn,#data 2 | |
| A | MOV @ Rr,A 0 | MOV @ Rr,A 1 | | MOVP A,@A | JMP page 5 | SEL MB2 | | CPL C | MOV Rr,A 0 | MOV Rr,A 1 | MOV Rr,A 2 | MOV Rr,A 3 | MOV Rr,A 4 | MOV Rr,A 5 | MOV Rr,A 6 | MOV Rr,A 7 |
| B | MOV @Rr, #data 0 | MOV @Rr, #data 1 | JB5 addr | JMPP @A | CALL page 5 | SEL MB3 | | | MOV Rr,#data 0 | MOV Rr,#data 1 | MOV Rr,#data 2 | MOV Rr,#data 3 | MOV Rr,#data 4 | MOV Rr,#data 5 | MOV Rr,#data 6 | MOV Rr,#data 7 |
| C | DEC @Rr 0 | DEC @Rr 1 | | | JMP page 6 | SEL RB0 | JZ addr | MOV A,PSW | DEC Rr 0 | DEC Rr 1 | DEC Rr 2 | DEC Rr 3 | DEC Rr 4 | DEC Rr 5 | DEC Rr 6 | DEC Rr 7 |
| D | XRL A, @Rr 0 | XRL A, @Rr 1 | JB6 addr | XRL A,#data | CALL page 6 | SEL RB1 | | MOV PSW,A | XRL A,Rr 0 | XRL A,Rr 1 | XRL A,Rr 2 | XRL A,Rr 3 | XRL A,Rr 4 | XRL A,Rr 5 | XRL A,Rr 6 | XRL A,Rr 7 |
| E | DJNZ @ Rr,addr 0 | DJNZ @ Rr,addr 1 | | | JMP page 7 | SEL MB0 | JNC addr | RL A | DJNZ Rr,addr 0 | DJNZ Rr,addr 1 | DJNZ Rr,addr 2 | DJNZ Rr,addr 3 | DJNZ Rr,addr 4 | DJNZ Rr,addr 5 | DJNZ Rr,addr 6 | DJNZ Rr,addr 7 |
| F | MOV A, @Rr 0 | MOV A, @Rr 1 | JB7 addr | | CALL page 7 | SEL MB1 | JC addr | RLC A | MOV A,Rr 0 | MOV A,Rr 1 | MOV A,Rr 2 | MOV A,Rr 3 | MOV A,Rr 4 | MOV A,Rr 5 | MOV A,Rr 6 | MOV A,Rr 7 |

MBA281

CONTENTS





## 1 INTRODUCTION

This data sheet describes the shared properties of the PCD33XXA family of microcontrollers and its quickly growing number of derivative microcontrollers. For a particular microcontroller, this data sheet should be read in conjunction with the individual data sheet of the specific device.

## 2 FEATURES

- 8-bit CPU, ROM, RAM, I/O all in one package
- Up to 8 kbytes ROM
- Up to 256 bytes RAM
- Over 100 instructions (based on MAB8048) all of 1 or 2 cycles
- 8 or more quasi-bidirectional I/O port lines
- 8-bit programmable timer/event counter 1
- 3 single-level vectored interrupts: external, timer/event counter, SIO/derivative
- Two test inputs, one of which also serves as the external interrupt input
- Serial I/O interface (some devices only)
- Power-on-reset, Stop and Idle modes
- Supply voltage range: 1.8 to 6 V
- Clock frequency: 1 to 16 MHz
- Operating temperature: –25 to +70 °C
- Manufactured in silicon gate CMOS process.

## 3 GENERAL DESCRIPTION

The PCD33XXA family of microcontrollers provide up to 8 kbytes of program memory and up to 256 bytes of RAM. All devices include flexible I/O ports, an 8-bit programmable timer/event counter and a choice of single-level vectored interrupts. The instruction set is based on that of the well-known MAB8048. Being similar to the MAB8400 family of NMOS controllers, some devices can serve as CMOS replacements, especially where the lower power consumption and higher speed provide advantages.

A range of prototyping devices with external program memory and 'Piggy-backs', as well as emulation probes and prototyping systems are available.

## 4 BLOCK DIAGRAM

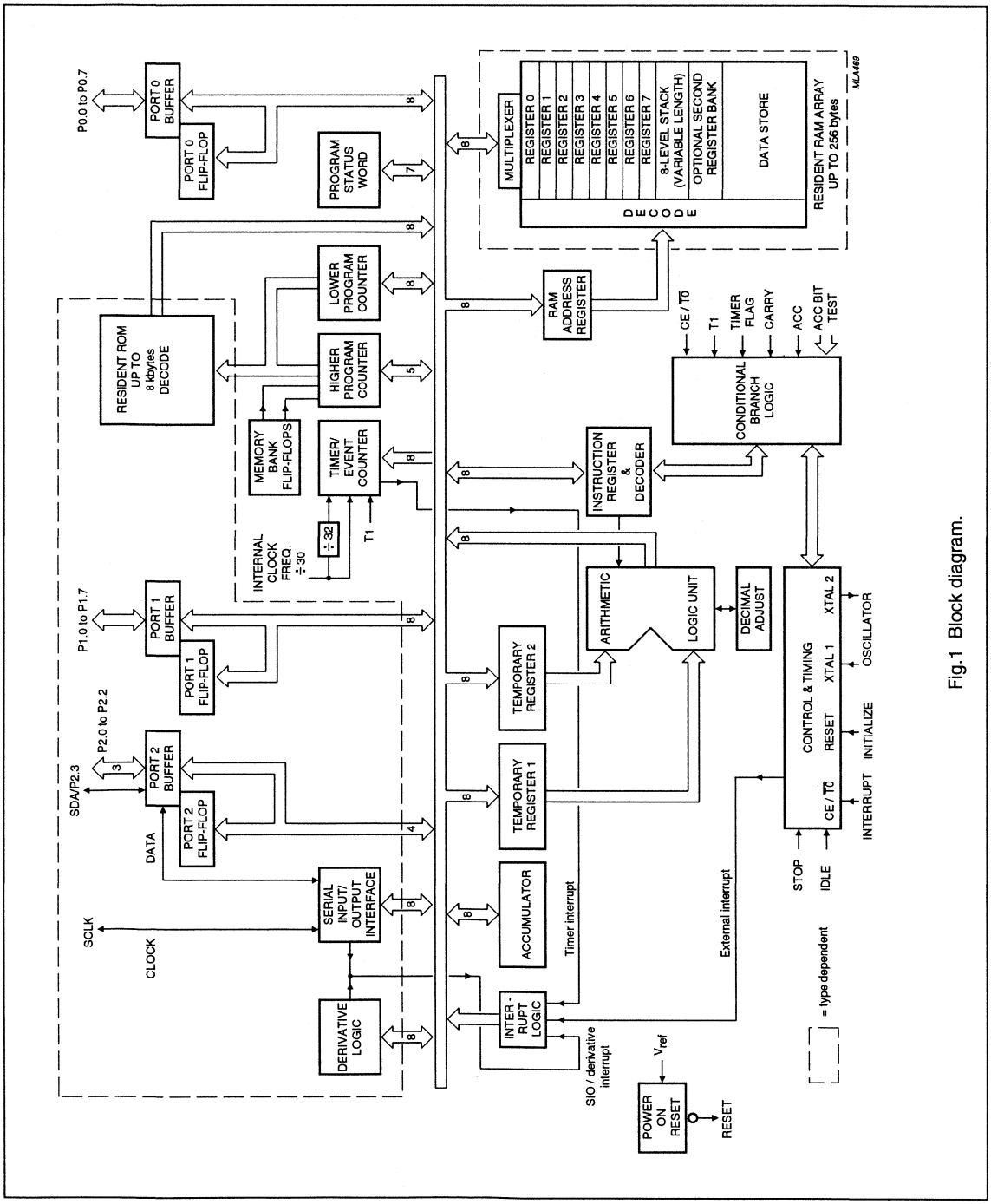



Fig.1 Block diagram.

## 5 PINNING INFORMATION

### 5.1 Pinning

For individual pinning configurations consult the data sheet of the specific device.

### 5.2 Pin description

Table 1 describes the common functions of the devices. For full details of pin descriptions consult the data sheet of the specific device.

**Table 1** Common functions.

| SYMBOL | TYPE | DESCRIPTION |
|---|---|---|
| $V_{SS}$ | P | Ground |
| $V_{DD}$ | P | Positive supply voltage |
| XTAL1 | I | Crystal oscillator/external clock input |
| XTAL2 | O | Crystal oscillator output |
| RESET | I | Reset input |
| CE/$\overline{T0}$ | I | Chip enable/Test 0 |
| T1 | I | Test 1/count input of 8-bit timer/event counter 1 |
| P0.0 to P0.7 | I/O | Port 0: quasi-bidirectional I/O lines |
| P1.0 to P1.7 | I/O | Port 1: quasi-bidirectional I/O lines |
| P2.0 to P2.2 | I/O | Port 2: quasi-bidirectional I/O lines |
| SDA/P2.3 | I/O | bidirectional data line of the serial I/O interface/Port 2: quasi-bidirectional I/O line |
| SCLK | I/O | bidirectional clock line of the serial I/O interface |

## 6 FUNCTIONAL DESCRIPTION

### 6.1 Central processing unit

The PCD33XXA family provides an adequate instruction set with arithmetic, logic, branching, input/output and control facilities. Special highlights are the instructions for BCD arithmetic, nibble handling, conditional branches, loop control (DJNZ) and table look-up (MOVP).

Code and execution efficiency is achieved by using a maximum of two bytes and two execution cycles per instruction (see Chapter 7).

### 6.2 Program memory

The program memory consists of up to 8 kbytes of read-only memory (ROM). Each location is directly addressable by the Program Counter. The program memory is mask-programmed at the factory. Figure 2 illustrates the program memory map.

Four program memory locations are of special importance:

- Location 0: first instruction to be executed after the processor is reset
- Location 3: first instruction of an external interrupt (CE/$\overline{T0}$) routine
- Location 5: first instruction of a SIO/derivative interrupt routine
- Location 7: first instruction of a timer/event counter interrupt routine.

Only 11 bits of the 13-bit Program Counter function as a counter. The two most significant bits can only be preset. The program memory is therefore, structured into banks of 2 kbytes. Transfer of control to other memory banks is performed by unconditional branches (JMP) or subroutine calls (CALL) when another memory bank has been pre-selected (by SEL MB instruction).

Each program memory bank is further divided into 8 pages of 256 bytes. Indirect (JMPP) and conditional branches cannot cross page boundaries.

### 6.3 Data memory

Data memory consists of up to 256 bytes of random access memory (RAM). All locations are indirectly addressable using RAM pointer registers. Up to 16 register locations are directly addressable. Data memory also includes an 8-level Program Counter stack addressed by a 3-bit Stack Pointer. All RAM locations make efficient program loop counters if used with the decrement register and test instruction (DJNZ). Figure 3 illustrates the data memory map.

#### 6.3.1 Working registers

Locations 0 to 7 are working registers. They are accessible by efficient one byte/one cycle instructions, thus making these locations suitable for frequently accessed intermediate results.

As an alternative to locations 0 to 7, locations 24 to 31 may be used as working registers. Register Bank selection is made by SEL RB0/RB1 instructions. Register Bank 1 may be used as an extension of Register Bank 0, as an alternative register bank for interrupt service or as general purpose data memory.

The first two locations of each bank (R0, R1, R0' and R1') serve as RAM pointers that indirectly address all RAM locations.

#### 6.3.2 Program Counter stack

Locations 8 to 23 may be used as an 8-level Program Counter stack reserving 2 locations per level, or as general purpose RAM. The stack (see Fig.5) saves return addresses and status during interrupt or subroutine servicing. Nesting of subroutines and/or interrupts is permitted up to 8-levels deep.

The 3-bit Stack Pointer always points to the next free stack level. Following device reset, the Stack Pointer points to level 0 (locations 8 and 9). On each subroutine call (CALL) or interrupt, the contents of the Program Counter and bits 4, 6 and 7 of the Program Status Word are transferred to the level indicated by the Stack Pointer. The Stack Pointer increments and points to the next free level. Overflow from level 7 to level 0 occurs after nesting eight levels deep. Further subroutine calls and/or interrupts must not occur at this stage since this would result in loss of program content; overriding level 0 content.

Return from interrupt must be performed by the RETR instruction, which decrements the Stack Pointer and restores the Program Counter and Program Status Word, valid before the interrupt occurred. Return from subroutine should be performed by the RET instruction. In contrast to RETR, RET does not restore the Program Status Word.

As a general rule, the use of RETR in conjunction with a subroutine call is not recommended. The use of RETR must also be avoided with subroutines called from interrupt routines because it prematurely terminates the interrupt state (see Section 6.6).

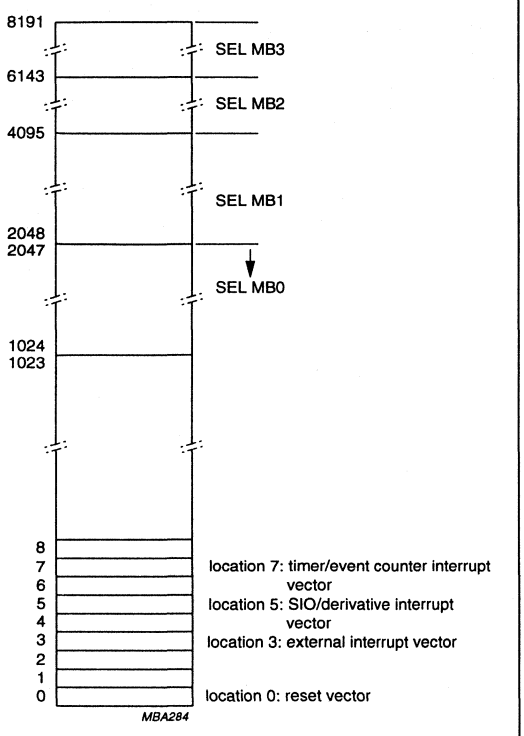



Fig.2 Program memory map,

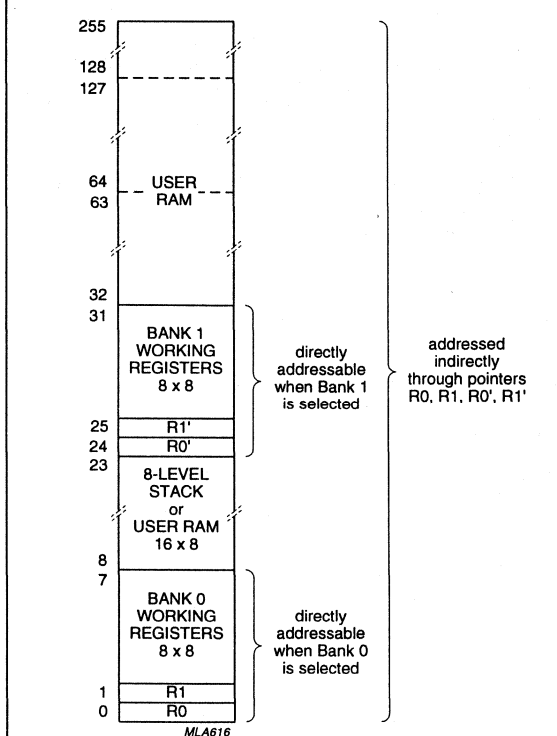



Fig.3 Data memory map.

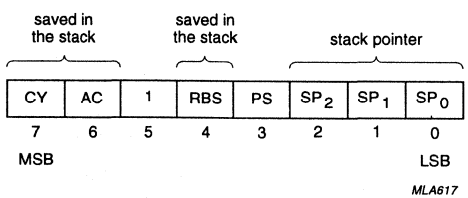



Fig.4 Program Status Word.

### 6.4 Program Counter

The 13-bit Program Counter is able to address up to 8 kbytes of ROM (see Fig.6). 11 bits (PC0 to PC10) are auto-incrementing. The two most significant bits (PC11 and PC12) must be changed under program control by SEL MB followed by a JMP or CALL instruction.

### 6.5 Program Status Word

The Program Status Word (PSW) is an 8-bit register in the CPU which stores information about the current status of the microcontroller (see Fig.4).

The PSW bits are:

- Bits 0 to 2: Stack Pointer bits (SP0, SP1, SP2)
- Bit 3: timer Prescaler Select (PS); 0 = modulo-32, 1 = modulo-1 (no prescaling)
- Bit 4: working Register Bank Select (RBS); 0 = register bank 0, 1 = register bank 1
- Bit 5: not used (fixed at 1)
- Bit 6: Auxiliary Carry (AC); half-carry bit generated by an ADD instruction and used by the decimal adjust instruction DA A
- Bit 7: Carry (CY); the carry flag indicates that the previous operation resulted in an overflow of the Accumulator.

All bits can be read using the MOV A, PSW instruction. Bits 0, 1 and 2 are affected by CALL, RET, RETR and interrupts. Bit 3 can be controlled by MOV PSW, A and bit 4 by SEL RB instructions. Bit 6 is set and cleared as a side-effect of ADD and ADDC instructions. Bit 7 is affected by ADD, ADDC, DA, RLC, RRC, CLR C and CPL C instructions.

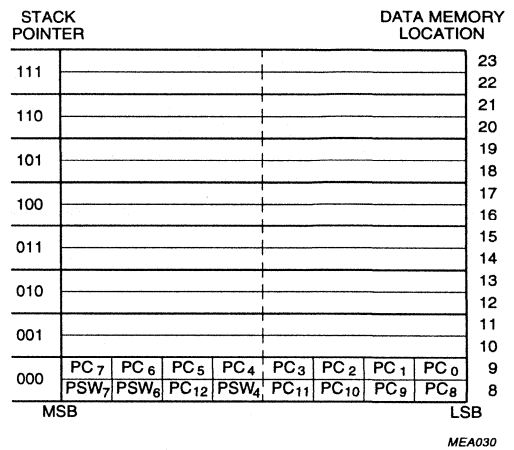



Fig.5 Program Counter Stack.

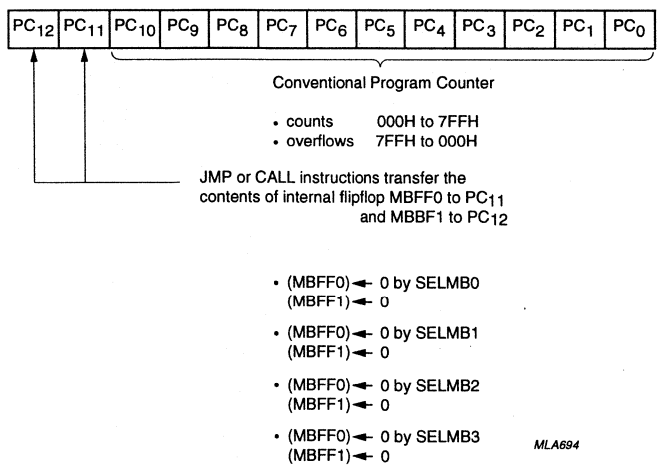



Fig.6 Program Counter.

## 6.6 Interrupts

External, SIO/derivative and timer/event counter interrupts are handled by the PCD33XXA family. The interrupt mechanism is single level, i.e. an executing interrupt routine cannot be pre-empted unless by reset. Further interrupt requests are latched. If several interrupt requests are detected simultaneously, they are honoured according to their priority:

- External interrupt (highest priority)
- SIO/derivative interrupt
- Timer/event counter interrupt (lowest priority).

An interrupt request is only sensed if the corresponding enable flag is set (see Fig.7). When the request is honoured, the contents of the Program Counter and bits 4, 6 and 7 of the Program Status Word are saved on the Program Counter stack. The Program Counter is loaded with the appropriate interrupt vector, thereby indicating the beginning of the interrupt routine. Since the Accumulator is not automatically saved, it must be saved and restored by user software. The interrupt routine must be terminated by the RETR (return and restore) instruction. At least one instruction of the main program will then be executed before another interrupt routine is entered. To avoid erroneous real-time programs, a few words of caution:

- While the interrupt is in progress, the two most significant bits of the Program Counter are frozen at zero. Thus, interrupt routines and subroutines called from interrupt routines must reside entirely in Bank 0.
- The SEL MB instruction must not be used in interrupt routines and in subroutines called from interrupt routines. Otherwise, the changed contents of MBFF0 and MBFF1 (see Fig.6) may lead to erroneous JMP and CALL destinations after return from interrupt.
- Subroutines and nested subroutines called from the interrupt routine must all end with RET since RETR clears the Interrupt In Progress flag (IIP), as a side-effect (see Figs 7 and 8). Further pending interrupts would then interfere with the interrupt routine in progress.

### 6.6.1 External interrupt

A LOW-to-HIGH transition on the CE/$\overline{T0}$ pin is latched in the digital filter/latch if the HIGH state exceeds 7 clock periods after a LOW state of more than 4 clock periods. If the external interrupt is enabled the External Interrupt Flag (EIF) is also asserted, thus constituting a valid external interrupt request. As soon as the IIP is clear, indicating that no interrupt routine is in progress, the external interrupt is invoked by a forced CALL to location 3. The EIF is simultaneously cleared (see Figs 7 and 8). The interrupt routine may acknowledge the interrupt via port lines. Execution of a DIS I (disable external interrupt) instruction cancels a stored interrupt request by clearing both the digital filter/latch and the EIF.

### 6.6.2 SIO/Derivative interrupt

The SIO/derivative interrupt is shared between the serial I/O interface (if available) and the derivative logic (if available). Software polling may be necessary to determine the origin of a request.

An interrupt condition in the serial I/O interface and/or the derivative logic will pull the PIN line LOW. If the SIO/derivative interrupt is enabled and no interrupt routine is in progress, the SIO/derivative interrupt routine will be invoked by a forced CALL to program memory location 5. The SIO/derivative interrupt routine must include instructions that will remove the cause of the SIO/derivative interrupt and thus reset PIN to its inactive HIGH state (for further details see Section 6.11). For derivative interrupts, consult the data sheet of the specific device.

### 6.6.3 Timer/event counter interrupt

If the timer/event counter interrupt is enabled, a timer/event counter 1 overflow sets the Timer Interrupt Flag (TIF). As soon as IIP is clear, meaning that no interrupt routine is in progress, the timer/event counter interrupt routine is invoked by a forced CALL to program memory location 7. The TIF is simultaneously cleared (see Figs 7 and 8). Execution of a DIS TCNTI (disable timer/event counter interrupt) instruction cancels a stored interrupt request by clearing TIF.

The timer/event counter interrupt may also be used to simulate a second external interrupt. After an enable timer/event counter interrupt (EN TCNTI), the counter mode is enabled by a STRT CNT instruction which loads FFH (the state preceding overflow) into the counter. A positive edge on the T1 pin will overflow the counter and set TIF.

## 6.7 Chip Enable/Test 0 Input (CE/$\overline{T0}$)

The CE/$\overline{T0}$ input has two purposes:

- External interrupt input (see Section 6.6)
- Test 0 input.

When used as a Test 0 input (external interrupt disabled) the conditional branch instruction JT0 will cause a jump if CE/$\overline{T0}$ = 1. The conditional branch instruction JNT0 will also cause a jump if CE/$\overline{T0}$ = 0. If CE/$\overline{T0}$ is not used, it must be tied to $V_{DD}$ or $V_{SS}$.

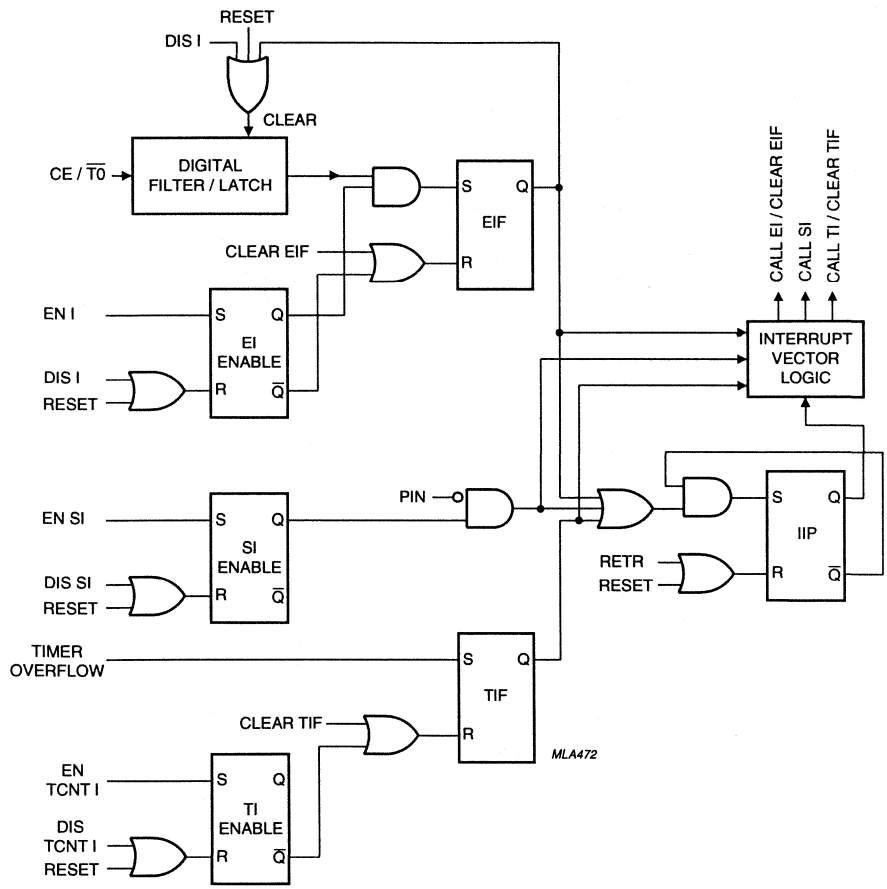



Fig.7 Simplified interrupt logic schematic (the R input overrules the S input for all flags).

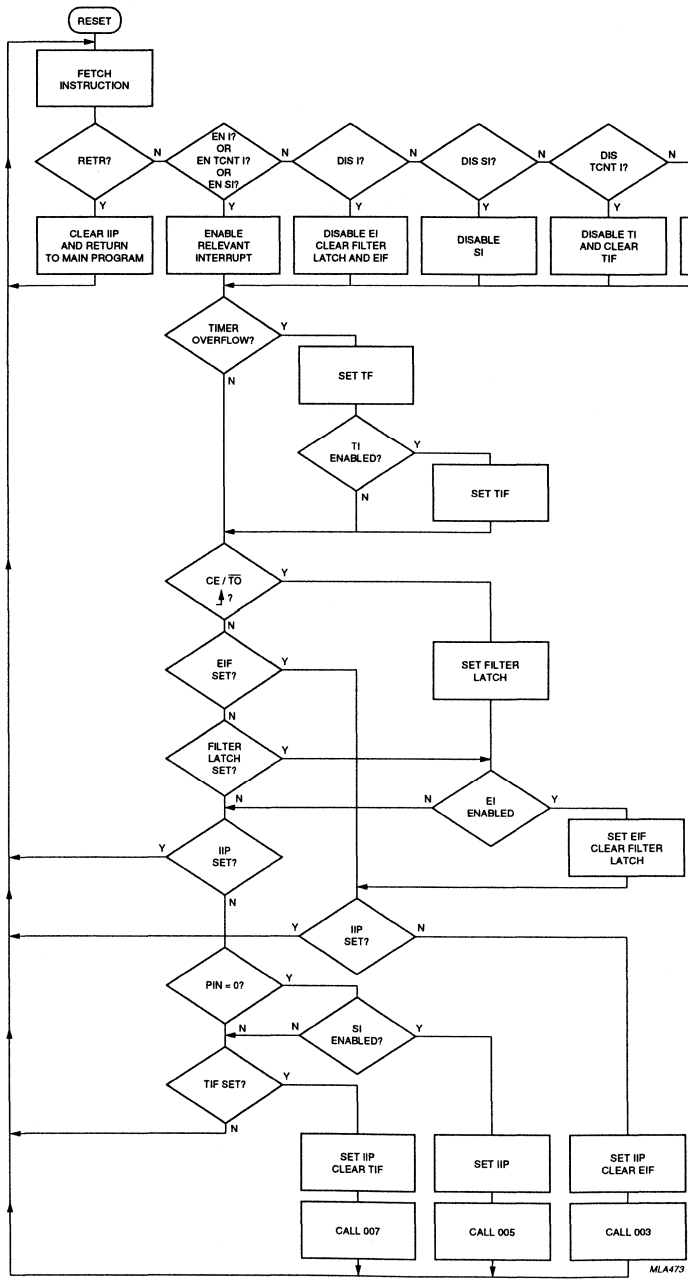



Fig.8 Flow chart illustrating CPU control in the presence of interrupts.

### 6.8 Timer/event counter 1

An internal 8-bit up counter is provided. The counter can be preset and read by the MOV T, A and MOV A, T instructions.

When the counter is to be used in the timer mode, a STRT T (start timer) instruction must be executed. Depending on the PS bit in the Program Status Word, the counter will increment every machine cycle (PS = 1, $\frac{1}{30}f_{xtal}$) or every 32 machine cycles (PS = 0, $\frac{1}{960}f_{xtal}$). STRT T clears the prescaler (see Fig.9) which is not otherwise accessible.

To count external events a STRT CNT (start event counter) instruction must be executed. A LOW-to-HIGH transition on pin T1 is counted if the HIGH state exceeds 4 clock periods after a LOW state of more than 4 clock periods. The maximum count rate is one increment per machine cycle ($\frac{1}{30}f_{xtal}$).

The timer mode and the event counter mode are both inhibited after reset or by executing a STOP TCNT (stop timer/event counter) instruction (see Fig.9).

In both the timer and in event counter modes, overflow has two effects:

- If the timer/event counter interrupt is enabled TIF is asserted thereby generating a timer/event counter interrupt request (see Section 6.6).
- The Timer Flag (TF) is set. TF can be tested by conditional branch instructions JTF (jump if TF = 1) or JNTF (jump if TF = 0). The JTF and JNTF instruction, as a side-effect, reset TF. The only other way to clear TF is to reset the microcontroller.

### 6.9 Test 1/count input (T1)

The T1 input has two purposes:

- Count input of 8-bit timer/event counter 1 (see Section 6.8)
- Test 1 input.

When used as a Test 1 input the conditional branch instruction JT1 will cause a jump if T1 = 1. The conditional branch instruction JNT1 will also cause a jump if T1 = 0. If T1 is not used, it must be tied to $V_{DD}$ or $V_{SS}$.

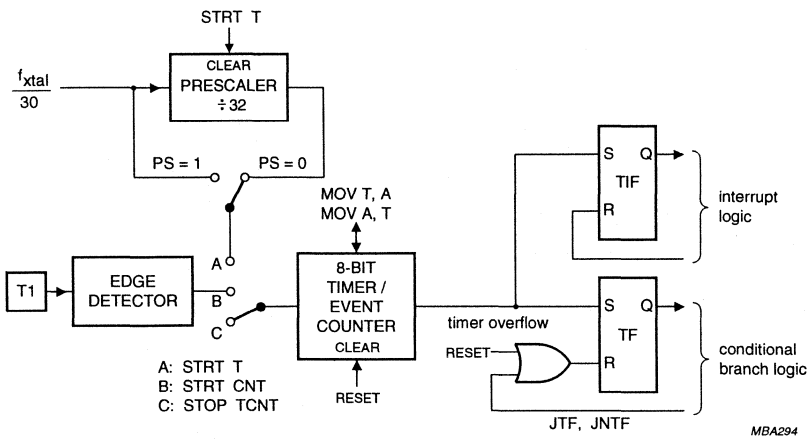



Fig.9 Timer/event counter 1.

### 6.10 Parallel ports

Three standard quasi-bidirectional I/O ports are defined:

- Port 0: parallel port of 8 lines (P0.0 to P0.7)
- Port 1: parallel port of 8 lines (P1.0 to P1.7)
- Port 2: parallel port of 4 lines (P2.0 to P2.2, SDA/P2.3).

Several members of the PCD33XXA family provide all 20 port lines. The eight Port 0 lines (P0.0 to P0.7) are available as a minimum. In addition to the standard ports, many PCD33XXA microcontrollers offer a variety of derivative ports. Please consult the data sheet of the specific device.

In general, all parallel ports can be used as either inputs or outputs. Output data written to a port is latched and remains unchanged until rewritten. If the port is used as an input, the external data is not latched and must remain stable until it is accessed by the CPU.

The standard port configuration is illustrated in Fig.11. When a logic 0 is written to the master/slave flip-flop, TR2 and TR3 are both in the OFF condition. TR1 turns ON and drives the output to $V_{SS}$.

When a logic 1 is written to the master/slave flip-flop, TR1 turns OFF. TR2 and TR3 both turn ON driving the output rapidly to $V_{DD}$. TR2 remains in the ON condition for the duration of the write pulse only. The constant current source is responsible for keeping the output line high. Sufficient source current is available for a TTL load HIGH level; the line can, however, be overridden by an external device. This is used when the port line serves as an input, but it may also be useful for wired-OR applications. In the latter case, unnecessary current through external devices is avoided since repeated logic 1 write operations will not activate TR2. The booster transistor TR2 is only asserted during a LOW-to-HIGH transition of the master/slave flip-flop. If the port line is to be used as an input, a logic 1 should first be stored in the master/slave flip-flop to turn TR1 OFF.

Access to Ports 0, 1 and 2 is provided by the parallel input/output instructions IN, OUTL, ANL and ORL. IN inputs port data to the Accumulator. OUTL outputs Accumulator data to the port. ANL and ORL are used for data manipulation in the port flip-flop. In contrast to Ports 0, 1 and 2, derivative ports are accessed by the derivative input/output instructions MOV, ANL and ORL. ANL and ORL are used for data manipulation in the port flip-flop. MOV is used for all data transfers between port and Accumulator. The source data for the Accumulator can be loaded from either the port line or the port flip-flop. Two derivative addresses are therefore provided per port (see Table 2).

All standard and derivative port accesses are performed by two-cycle instructions. Their instruction timing is shown in Fig.10. For input, data on port lines is sensed during timeslots 3 and 4 of machine cycle 2 (see Sections 6.12 and 6.13). For output, the data change occurs in timeslot 7. For OUTL, data changes during machine cycle 1. For ANL, ORL and MOV Dx, A, data changes during machine cycle 2.

**Table 2** Derivative port address pair.

| ADDRESS | TYPE | ACCESS |
|---|---|---|
| 8-bit line address | R | derivative port line |
| 8-bit flip-flop address | R/W | derivative port flip-flop |

Three port output mask options are available:

Option 1 Standard Port; quasi-bidirectional I/O with switched pull-up current source of 100 µA (typ.) and p-channel booster transistor TR2. TR2 is only active for 1 clock cycle during LOW-to-HIGH transitions (see Fig.11).

Option 2 Open Drain; quasi-bidirectional I/O with only an n-channel open drain output. Application as an output requires connection of an external pull-up resistor (see Fig.12). If unused, an Option 2 output should be tied to $V_{SS}$. This keeps the input path from floating, thereby avoiding undesirable current flow through input stages.

Option 3 Push-pull; drive capability of the output will be 5 mA (typ.) at $V_{DD}$ = 3 V in both polarities. Since short circuit currents would flow during input, push-pull lines must only be used as outputs (see Fig.13).

If available, SDA/P2.3 is shared between the parallel Port 2 and the serial I/O interface. Therefore, only option 2 is permitted for SDA/P2.3. For the remaining standard port lines (P0.0 to P2.2), all three options are generally available.

Besides port output mask options, the port flip-flop state, after reset, may be specified for each individual port line (except SDA/P2.3). Usually the 'set option' will be selected, which avoids short-circuits for ports intended as inputs. However, there may be cases in which the port should output a logic zero after reset. The user may then specify the 'reset option' for certain port lines.

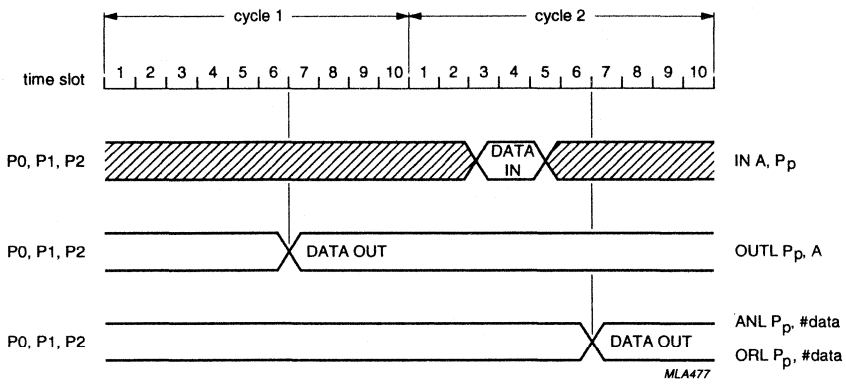



Fig.10 Input /output timing of standard and derivative ports.

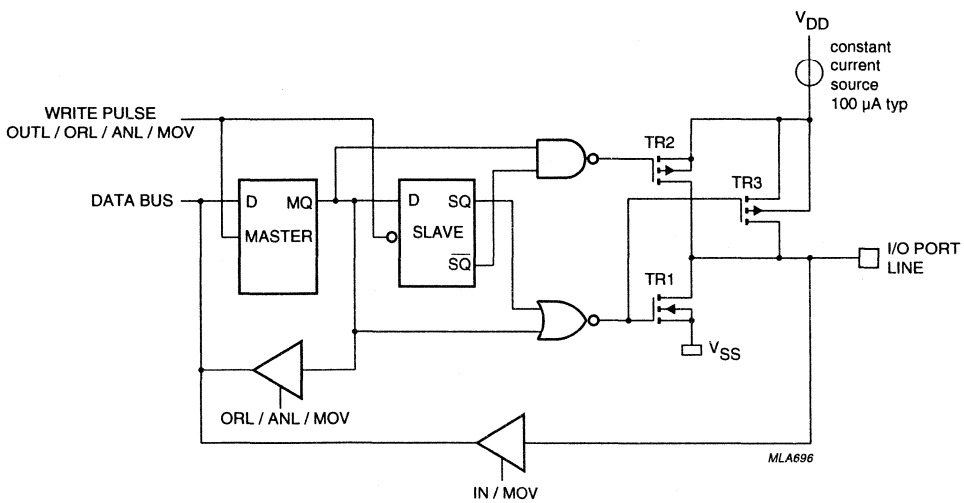



Fig.11 Standard output with switched current source.

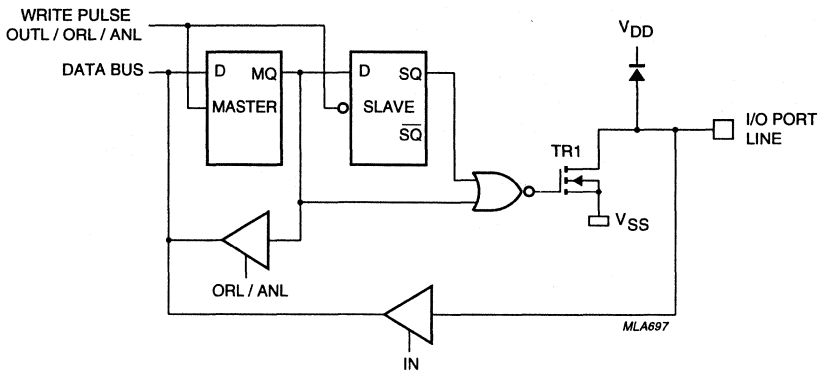



Fig.12 Open drain output.

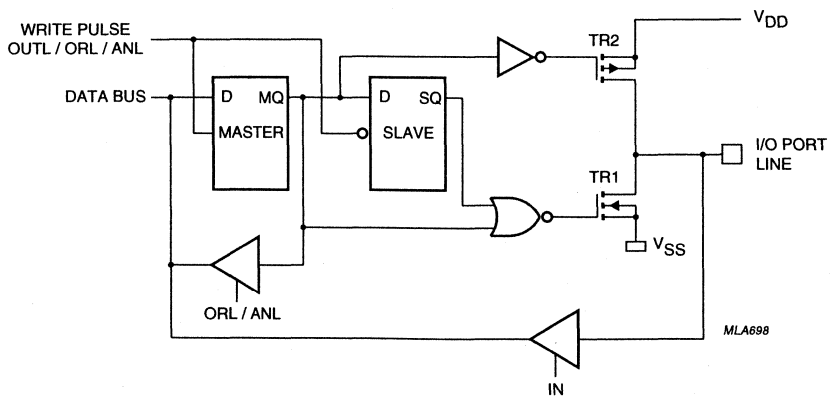



Fig.13 Push-pull output.

### 6.11 Serial I/O interface

Many members of the PCD33XXA family have a serial I/O interface (I²C-bus or 'Inter-Integrated Circuit Bus'). This two-line serial bus extends the microcontroller capabilities when implemented with the powerful I²C-bus devices of the PCF85XX, PCD33XX and 'Clips' peripheral families.

Microcontrollers that do not have a serial I/O interface can simulate it by software, by using port pins. However, such microcontrollers must continuously monitor the serial bus. As well as degrading the maximum data transfer rate, this approach may also consume significant processing and memory resources.

If available, however, the serial I/O interface detects the valid 7-bit I²C-bus address of the device, transfers serial data and provides data conversion to and from parallel format, all without disrupting program execution. Only when a complete byte has been transferred, an interrupt is requested by which the next data byte can be written to or read out of the serial I/O interface. The serial I/O interface also facilitates the implementation of multimaster systems in which two or more microcontrollers communicate via the same I²C-bus. An automatic arbitration procedure resolves bus conflicts.

The I²C-bus consists of a bidirectional clock line (SCL) and a bidirectional data line (SDA). Whereas SCL uses the dedicated pin SCLK, SDA and Port line P2.3 share the pin, SDA/P2.3. When the serial I/O interface is enabled, SDA/P2.3 is disabled as a port line. Input signals on SCLK and SDA are filtered for enhanced noise immunity. When used as outputs, SCLK and SDA/P2.3 require an external pull-up resistor because they are open drain. If unused, SCLK and SDA/P2.3 should be tied to $V_{SS}$ (see Section 6.10, Option 2).

Communication between CPU and serial I/O interface is handled through the four serial I/O interface registers S0, S0', S1 and S2 (see Fig.14).

A detailed description of the I²C-bus specification, with applications, is given in the brochure *"The I²C-bus and how to use it"*. This brochure may be ordered using the code 9398 393 40011.

#### 6.11.1 Data shift register (S0)

The Data Shift Register converts serial data to a parallel format and vice versa. The leading bit of a serial transfer corresponds to the most significant bit of the parallel word. An interrupt request is issued after transfer of a complete byte and after detection of the valid I²C-bus address. Register S0 is read by MOV A, S0. It is written to by MOV S0, A or MOV S0, #data if the ESO (Enable Serial I/O) bit in the Status Register (S1) is set.

#### 6.11.2 Address register (S0')

The Address Register contains the 7-bit I²C-bus address of the device and the ALS (Always Selected) bit. When ALS is zero, which is the recommended mode of operation, bus transfers are ignored unless the valid device address immediately follows the start condition. Besides the stored 7-bit address, the 'general call address' (pre-defined as zero) is also acceptable as a valid address. If ALS is set, however, any transfer on the bus will be stored in the Data Shift Register.

The Address Register S0' is write-only. It can be written by MOV S0, A and MOV S0, #data if the ESO (Enable Serial I/O) bit in the Status Register (S1) is zero.

#### 6.11.3 Clock control register (S2)

The Clock Control Register defines the frequency of $f_{SCLK}$ as the microcontroller clock frequency divided by an integer (see Table 3). It also defines ASC (Asymmetrical Clock) and ACK (Acknowledge).

If ASC = 1, the generated SCLK has a duty cycle of approximately 75%. The asymmetrical clock limits the I²C-bus transmission rate to below 55 kHz. Divisors 39, 45 and 51 are not allowed if ASC = 1. However, an SCLK duty cycle of approximately 50% results if ASC = 0. This permits I²C-bus transmission rates of up to 100 kHz. All divisors of Table 3 are available. It is, therefore, recommended to select ASC = 0.

For the normal I²C-bus protocol ACK must be set. After each byte transfer an extra SCLK pulse is generated during which the receiver may acknowledge reception. If ACK is zero, no acknowledge phase is available. This mode is temporarily used when a master/receiver refuses the acknowledgement in order to signal an end of transmission to the slave transmitter (see Section 6.11.4.9).

The Clock Control Register (S2) is write-only. It can be written to by MOV S2, A and MOV S2, #data.

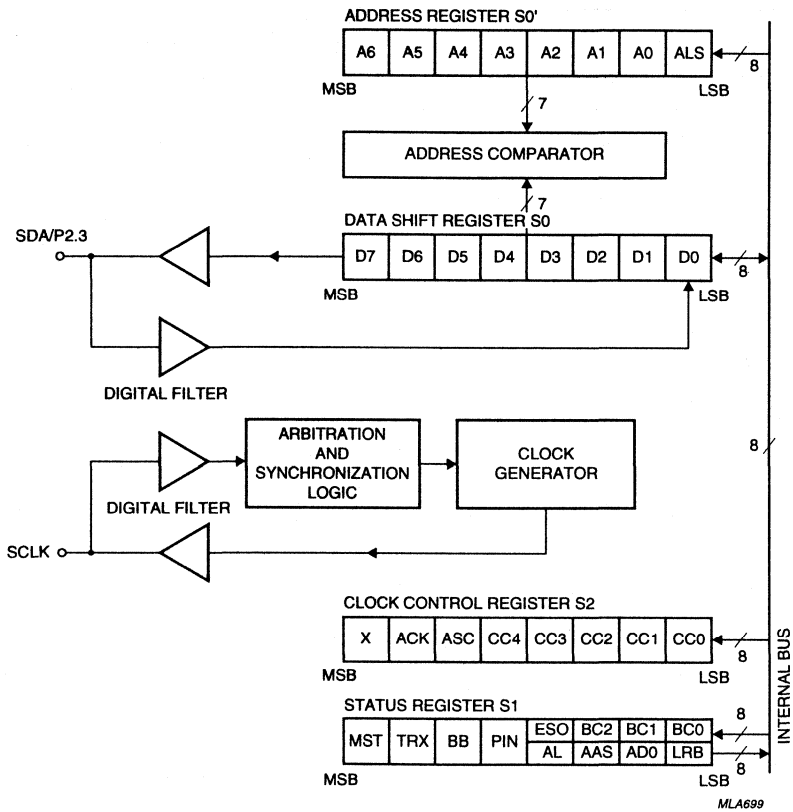



Fig.14 Block diagram of the serial I/O interface.

**Table 3** $f_{SCLK}$ as defined by Clock Control Register (S2).

| CC4 TO CC0 (HEX) | $f_{xtal}$ DIVISOR (DF) | $f_{SCLK}$ (kHz) at | | |
|---|---|---|---|---|
| | | $f_{xtal}$ = 3.58 MHz | $f_{xtal}$ = 10 MHz | $f_{xtal}$ = 16 MHz |
| 00 | forbidden | – | – | – |
| 01 | 39 | 91.8 | 256.4(1) | 410.3(1) |
| 02 | 45 | 79.5 | 222.2(1) | 355.6(1) |
| 03 | 51 | 70.2 | 196.1(1) | 313.7(1) |
| 04 | 63 | 56.8 | 158.7(1) | 254.0(1) |
| 05 | 75 | 47.7 | 133.3(1) | 213.3(1) |
| 06 | 87 | 41.1 | 114.9(1) | 183.9(1) |
| 07 | 99 | 36.2 | 101.0(1) | 161.6(1) |
| 08 | 123 | 29.1 | 81.3 | 130.1(1) |
| 09 | 147 | 4.4 | 68.0 | 108.8(1) |
| 0A | 171 | 20.9 | 58.5 | 93.6 |
| 0B | 195 | 18.4 | 51.3 | 82.1 |
| 0C | 243 | 14.7 | 41.2 | 65.8 |
| 0D | 291 | 12.3 | 34.4 | 55.0 |
| 0E | 339 | 10.6 | 29.5 | 47.2 |
| 0F | 387 | 9.2 | 25.8 | 41.3 |
| 10 | 483 | 7.4 | 20.7 | 33.1 |
| 11 | 579 | 6.2 | 17.3 | 27.6 |
| 12 | 675 | 5.3 | 14.8 | 23.7 |
| 13 | 771 | 4.6 | 13.0 | 20.8 |
| 14 | 963 | 3.7 | 10.4 | 16.6 |
| 15 | 1155 | 3.1 | 8.7 | 13.9 |
| 16 | 1347 | 2.7 | 7.4 | 11.9 |
| 17 | 1539 | 2.3 | 6.5 | 10.4 |
| 18 | 1923 | 1.9 | 5.2 | 8.3 |
| 19 | 2307 | 1.6 | 4.3 | 6.9 |
| 1A | 2691 | 1.3 | 3.7 | 5.9 |
| 1B | 3075 | 1.2 | 3.3 | 5.2 |
| 1C | 3843 | 0.9 | 2.6 | 4.2 |
| 1D | 4611 | 0.8 | 2.2 | 3.5 |
| 1E | 5379 | 0.7 | 1.9 | 3.0 |
| 1F | 6147 | 0.6 | 1.6 | 2.6 |

**Note**

1. Not permitted; maximum $f_{SCLK}$ = 100 kHz in $I^2C$-bus systems.

### 6.11.4 Status Register (S1)

The Status Register controls the serial I/O interface and provides feedback concerning on-going bus transfers. Register S1 can be accessed by MOV A, S1, MOV S1, A and MOV S1, #data. The lower nibble of the Status Register is twofold: control bits BC0 to BC2 and ESO can only be written, whereas feedback bits LRB, AD0, AAS and AL can only be read. Table 4 describes the status bits.

The status bits interact in intricate ways with each other. This must be kept in mind when an $I^2C$-bus application is programmed.

#### *6.11.4.1 Master bit (MST) and Transmitter bit (TRX)*

MST and TRX together define the state of the serial I/O interface. When not engaged in a bus transfer MST and TRX should always be at zero, the slave/receiver state (see Fig.15). A return to this state is always performed by software. If the previous state was the master state, the transition (to slave/receiver by MOV1,#D8H) involves a stop condition which, as a consequence, clears both MST and TRX.

The transition to the master/transmitter state is also a programmed event. However, transitions to the master/receiver and the slave/transmitter states occur automatically if ALS = 0 (standard $I^2C$-bus protocol). A slave/receiver becomes a slave/transmitter if $R/\overline{W} = 1$, in its valid address (following the start condition). A master/transmitter becomes a master/receiver if $R/\overline{W} = 1$, in the transmitted address.

#### *6.11.4.2 Pending Interrupt Not bit (PIN)*

If MST = 1 or, if ALS = 1, PIN is set to zero after every byte transfer. Conversely, PIN becomes zero when a valid address is detected and after each byte of the following transfer. In addition, the serial interrupt request, PIN = 0 initiates 'clock synchronization', i.e. the SCLK line is pulled to $V_{SS}$ as long as PIN = 0. With this feature a slave may slow down a master, thus providing time to read the Data Register (in the case of a slave/receiver) or to write to the Data Register (in the case of a slave/transmitter). PIN is cancelled by an access to register S0 or by explicitly setting PIN to one.

If the SIO/derivative interrupt is disabled, the serial I/O interface may be serviced by testing PIN directly in user software.

#### *6.11.4.3 Bus Busy bit (BB)*

The Bus Busy bit (BB) is controlled by the serial I/O interface or by software in the bus master to generate the start and stop conditions. When a master clears BB (by MOV S1, #D8H), the serial I/O interface automatically clears MST and TRX, thereby returning to the slave/receiver state (see Fig.15). If BB = 1, write access to S1 is inhibited, except for the master or an addressed slave. Should BB be inadvertently set by excessive noise on the bus, the deadlock can be resolved by two consecutive MOV S1, #18H, the first of which just clears BB.

When a slave/transmitter detects an end of transmission (signalled by the lack of an acknowledgment from the master receiver), it has to access S1 in order to cancel PIN and to become slave/receiver. However, BB should remain set. This is reflected by MOV S1, #38H as illustrated in Fig.15. With PIN = 1, 'clock synchronization' terminates, enabling the master to generate the stop condition.

A start condition must only be generated when BB = 0; otherwise the serial I/O interface will respond as if bus arbitration has been lost (see Section 6.11.4.4).

#### *6.11.4.4 Arbitration Lost bit (AL)*

The AL bit is set by the serial I/O interface when it loses a bus arbitration in the master/transmitter mode. MST and TRX are cleared simultaneously to enable the interface, now in slave/receiver mode, to determine if it is validly addressed by the device that won the arbitration. PIN is activated when the byte transfer is complete. AL will be cleared when the serial interrupt is cancelled.

#### *6.11.4.5 Addressed As Slave bit (AAS)*

AAS is set by the serial I/O interface following a start condition when the valid address is detected (ALS = 0 in register S0') or when the first byte is received (ALS = 1 in register S0'). AAS is cleared when the serial interrupt is cancelled.

#### *6.11.4.6 Address Zero bit (AD0)*

AD0 is set, independently of ALS, by the serial I/O interface when byte 00H, the 'general call' address, is detected following a start condition. AD0 is cleared after a repeated start or a stop condition.

*6.11.4.7 Last Received Bit (LRB)*

LRB corresponds to the last bit transferred. If ACK = 1, LRB contains the acknowledgement bit. It remains valid as long as PIN = 0.

*6.11.4.8 Enable Serial I/O bit (ESO)*

When ESO = 0 access to register S0' is enabled. SCLK is in the high-impedance state and SDA/P2.3 is available as a normal port line.

When ESO = 1 the serial I/O interface and access to register S0 are enabled. Only when ESO = 1 may the other bits of register S1 be changed. SCLK and SDA/P2.3 are enabled as serial clock and data lines, respectively.

To avoid bus deadlock, ESO must be set to zero prior to the execution of the STOP instruction.

*6.11.4.9 Bit Counter bits (BC0, BC1 and BC2)*

The bit counter bits BC0, BC1 and BC2 should all be at zero for normal $I^2C$-bus operation. The bit counter is always cleared by a start condition. Therefore, all eight bits of the first byte are transferred.

If a non-zero bit counter value is chosen, it is only valid for one register S0 transfer since the counter decrements to zero. An important use of the bit counter arises when a master/receiver signals an end of transmission by sending a negative acknowledge after the last byte received. To do this, the last byte is received with bit ACK = 0 in register S2. The negative acknowledge is then issued by setting the bit counter to one and 'receiving' one bit from the HIGH level available on the SDA line. The slave/transmitter interprets the same signals as a negative acknowledgement.

**Table 4** Overview of Status Register bits.

| BIT | NAME | TYPE | DESCRIPTION |
|---|---|---|---|
| MST | Master | R/W | MST = 0: slave (SCLK input).<br>MST = 1: master (SCLK output). |
| TRX | Transmitter | | TRX = 0: receiver (SDA/P2.3 input.<br>TRX = 1: transmitter (SDA/P2.3 output). |
| BB | Bus Busy | R/W | BB = 0: bus inactive (R)/generates stop condition (W).<br>BB = 1: bus busy (R)/generates start condition (W). |
| PIN | Pending Interrupt Not | R/W | PIN = 0: serial interrupt pending (after byte transfer, valid address or lost arbitration). SCLK line forced to $V_{SS}$.<br>PIN = 1: no serial interrupt pending. |
| ESO | Enable Serial Output | W | ESO = 0: serial I/O interface disabled/write access to S0' possible.<br>ESO = 1: serial I/O interface enabled write access to S0 possible. |
| BC0 to BC2 | Bit Counter 0 to 2 | W | 3-bit binary value of 0 to 7, counting down the number of bits transferred (0 used for complete byte). |
| AL | Arbitration Lost | R | Set: when a bus conflict is lost.<br>Reset: when corresponding serial interrupt (PIN) is cancelled. |
| AAS | Addressed As Slave | R | Set: following a start condition if valid address is detected (ALS = 0) or if first byte is received (ALS = 1).<br>Reset: when corresponding serial interrupt (PIN) is cancelled. |
| AD0 | Address zero | R | Set: following a start condition if byte 00H ('general call' address) is detected.<br>Reset: after a repeated start or a stop condition. |
| LRB | Last Received Bit | R | Set or Reset depending on the value of the last bit transferred, acknowledgement bit if ACK = 1. |

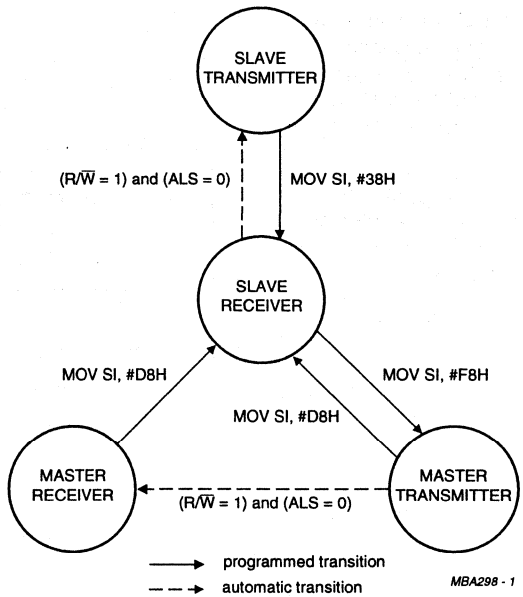



Fig.15 State diagram of the serial I/O interface.

### 6.12 Timing

Every machine cycle consists of 10 time slots which are again subdivided into 3 clock periods each (see Fig.16).

Permitted clock frequencies range from 1 MHz to a maximum, which is a function of the supply voltage. At $V_{DD} \geq 4.5$ V, a 16 MHz maximum clock frequency is guaranteed.

The clock signal may be internally generated by an on-chip oscillator. Alternatively, an external clock may be applied to pin XTAL1. In this configuration, a short circuit with an internal pull-up transistor on XTAL1 may occur while the oscillator is inhibited (see Section 6.13). Care should be taken to avoid excessive current flow.

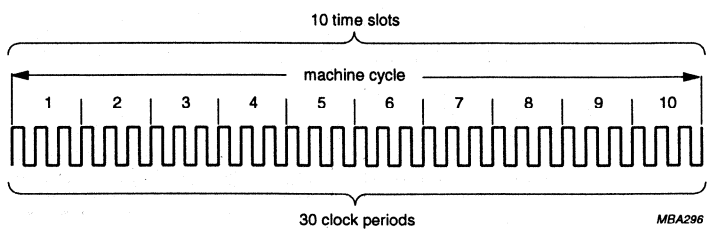



Fig.16 Timing of a machine cycle.

### 6.13 Oscillator

The on-chip oscillator basically consists of an inverter stage which includes a feedback resistor and load capacitors (see Fig.17). In most applications, a quartz crystal will be connected between XTAL1 and XTAL2. Alternatively, a ceramic resonator or an inductor may be used as a timing element.

When the supply voltage drops below the power-on reference level, the oscillator is inhibited. The internal oscillator can also be inhibited by the STOP instruction under software control (see Section 6.16).

The transconductance ($g_m$) of the inverter stage can be mask-programmed, thereby optimizing the oscillator for a specific frequency and resonator. Three standard transconductance options, referred to as LOW, MEDIUM and HIGH, can be specified by the user. Table 6 is intended as a rough selection guide for typical quartz and PXE resonators.

With $C_1 = C_2 = 10$ pF on-chip, external capacitors are not required for quartz oscillators. However, for adequate frequency stability, PXE resonators need external capacitors in the order of the static resonator capacitance $C_0$, such as external $C_1 = C_2 = 30$ to 100 pF.

Oscillator start-up time depends mainly on the external timing element. The start-up time of a quartz crystal is several milliseconds because of the narrow crystal bandwidth. For proper oscillator start-up, the transconductance ($g_m$) of the inverter stage must fulfil relationship (1); shown below.

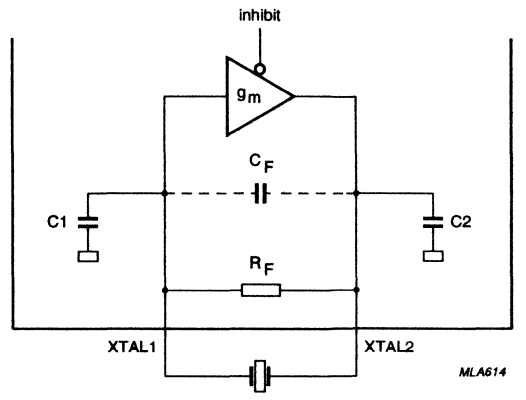



Fig.17 Oscillator with integrated elements.

**Table 5** Notation to relationship (see Figs 17 and 18).

| SYMBOL | DEFINITION |
|---|---|
| $R_X$ | resonator series resistance |
| $C_0$ | static resonator capacitance |
| $R_0$ | resonator loss resistance |
| $R_P$ | $R_0$ // $R_F$ |
| $R_F$ | feedback resistor |
| $C_L$ | C1 x C2/(C1 + C2) (load capacitance) |
| $C_F$ | parasitic feedback capacitance (typically 2 pF on-chip, external value depends on printed-circuit board wiring) |
| $\omega$ | $2\pi f_{osc}$ |

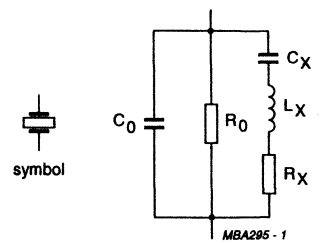



Fig.18 Crystal unit equivalent circuit.

$$4.2\left[R_X\omega^2(C_L+C_0+C_F)^2+\frac{1}{R_P}\right] < g_m < \frac{C1\times C2}{\left[R_X(C_0+C_F)^2+\frac{1}{\omega^2 R_P}\right]} \quad (1)$$

**Table 6** Recommended transconductance options for popular quartz and PXE resonators.

| OPTION | TYP. $g_m$ at 5 V (mS) | $f_{osc}$ FOR QUARTZ (MHz) | $f_{osc}$ FOR PXE (MHz) |
|---|---|---|---|
| LOW ($g_{mL}$) | 0.4 | 1 to 6 | 1 to 2.4 |
| MEDIUM ($g_{mM}$) | 1.6 | 4 to 12 | 1 to 6 |
| HIGH ($g_{mH}$) | 4.5 | 10 to 16 | 3 to 16 |

### 6.14 Reset

To ensure proper start-up, the microcontroller must be initialized to a defined starting condition. The device executes the first instruction 1866 clock cycles after the falling edge of the internal reset.

#### 6.14.1 PASSIVE RESET

A passive reset is generated by the RC circuit illustrated in Fig.19. While $V_{DD}$ rises, the discharged $C_{reset}$ keeps the RESET pin near the $V_{DD}$ level. When $V_{DD}$ crosses the power-on reference level ($V_{ref}$) the power-on reset circuit generates a reset pulse of approximately 50 µs. This pulse is without effect since it feeds into the reset signal forced by the pulse on the RESET pin. The $f_{xtal}$ dependent minimum $V_{DD}$ must be reached before the voltage on RESET drops below $V_{IH} = 0.7V_{DD}$. This translates into a lower bound for $C_{reset}R_{reset}$ equal to twice the rise time of $V_{DD}$ (for linearly rising $V_{DD}$) or eight times the time constant of $V_{DD}$ (for exponentially rising $V_{DD}$). The internal diode rapidly discharges $C_{reset}$ when $V_{DD}$ falls off, ensuring reliable reset even after short interruptions of supply voltage. To avoid overload of the internal diode, an external diode should be added in parallel if $C_{reset} > 2.2$ µF.

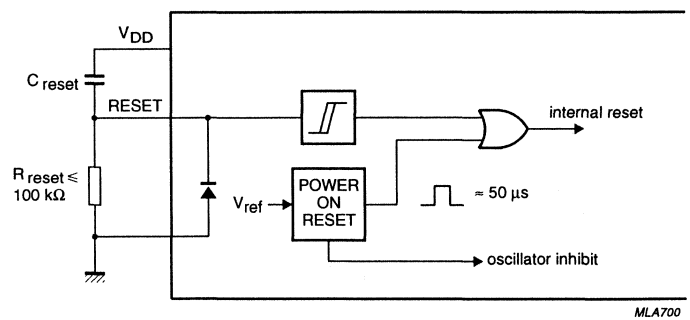



Fig.19 Passive power-on reset.

### 6.14.2 INTERNAL RESET

In systems where $V_{DD}$ reaches its $f_{xtal}$ dependent minimum operating value before the clock $f_{xtal}$ is applied, reset can be performed without external components. This condition is generally fulfilled with quartz and PXE resonators since oscillator start-up takes several milliseconds. Besides, rapid power-up is usually available in battery-powered systems.

If the internal power-on reset is used the RESET pin should be connected to $V_{SS}$. When $V_{DD}$ increases above the power-on reference level $V_{ref}$, the power-on reset circuit generates a reset pulse of approximately 50 µs. This pulse guarantees proper initialization under the conditions defined above.

The power-on reference level $V_{ref}$ is a mask option. The user can select a reference voltage between 1.2 V and 3.6 V in discrete steps of 100 mV. The accuracy of the reference voltage is ±500 mV for the $V_{ref}$ range 1.2 V to 3.0 V and ±800 mV for the $V_{ref}$ range 3.1 V to 3.6 V. The chosen $V_{ref}$ should have sufficient margin regarding the minimum intended $V_{DD}$.

A mask option without an internal power-on reset circuit is also available. It is recommended if the user does not intend to use the internal power-on-reset circuit. In this case, the supply current requirements in Stop mode (see Section 6.16) will reduce to the level of leakage currents, i.e. virtually zero at ambient temperature.

### 6.14.3 ACTIVE RESET

An active reset can be generated by driving the RESET pin HIGH from an external logic device. Such an active reset pulse should not fall off before $V_{DD}$ has reached its $f_{xtal}$ dependent minimum operating value.

### 6.14.4 RESET STATE

After a reset, the device state is characterized as follows:

- Program Counter 0
- Memory bank 0
- Register Bank 0 - Stack Pointer 0 (location pair 8 and 9)
- All interrupts disabled
- Timer/event counter 1 stopped and cleared
- Timer prescaler modulo-32 (PS = 0)
- Timer flag cleared
- All port flip-flops (except SDA/P2.3) set to 1 (set option) or 0 (reset option) as selected by the user
- SDA/P2.3 is high-impedance with the port flip-flop set to 1
- SCLK is high-impedance
- Serial I/O interface disabled (ESO = 0) and in slave/receiver mode (S0, S0', S1 and S2 cleared except for PIN = 1)
- Idle and Stop modes cancelled.

### 6.15 Idle mode

The Idle mode is very useful in low-power applications. When all computational tasks are completed, the device can be put into standby instead of into a busy waiting loop. Nevertheless, the device is on the alert and ready to respond rapidly to any interrupt.

The microcontroller enters the Idle mode via the IDLE instruction. In the Idle mode, all activity is halted except for the oscillator, the timer/event counter 1 and the serial I/O interface (if available).

The microcontroller leaves the Idle mode when an enabled interrupt occurs. The interrupt routine is executed before operation resumes with the instruction following the IDLE opcode.

For timer/event counter interrupts and SIO/derivative interrupts, termination of the Idle mode is straightforward. However, care must be taken when the Idle mode is left by the external interrupt since CE/$\overline{T0}$ is triggered on the rising edge. If CE/$\overline{T0}$ was HIGH prior to entering the Idle mode, it must be taken LOW before the positive edge can be generated. Figure 20 specifies the exact timing for leaving the Idle mode via the external interrupt CE/$\overline{T0}$.

If no interrupt is enabled, the Idle mode can only be terminated by an active signal on the RESET pin. A normal reset sequence is executed (see Fig.20).

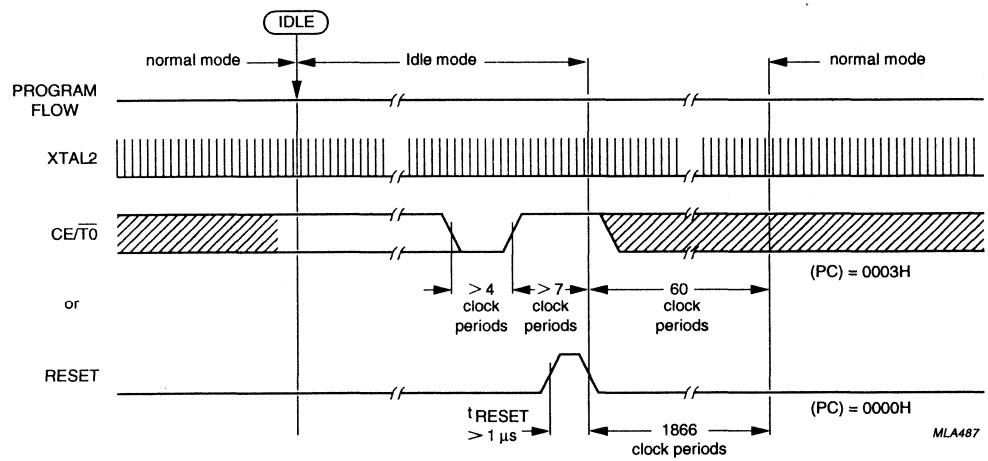



Fig.20 Entering and leaving the Idle mode.

### 6.16 Stop mode

The Stop mode allows very low-power applications. When all computational tasks are completed, the device can be almost completely shut off by stopping its oscillator. In contrast to the Idle mode, the device is not ready to respond rapidly to any interrupt.

When the microcontroller enters the Stop mode via the STOP instruction; the oscillator is switched off. All internal states and I/O levels are maintained.

The microcontroller leaves the Stop mode by a HIGH level on CE/$\overline{T0}$ or a reset. In the latter case, a normal reset sequence is executed (see Fig.21).

In contrast to the Idle mode and the external interrupt mechanism, the microcontroller responds to a HIGH level on CE/$\overline{T0}$ rather than to a positive edge. If CE/$\overline{T0}$ is HIGH when the STOP instruction is executed, the Stop mode will not be entered.

A positive edge on CE/$\overline{T0}$ continues program execution after a 1866 clock cycle delay, which ensures proper oscillator start-up. If the external interrupt is enabled, the device executes the instruction following the STOP opcode before diverting to the interrupt routine. If the external interrupt is disabled, program execution continues with the instructions following the STOP opcode (see Fig.21).

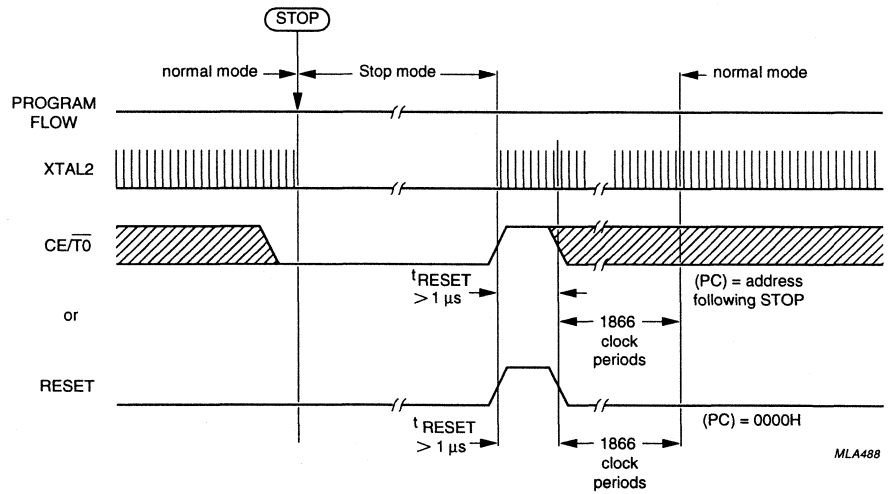



Fig.21 Entering and leaving the Stop mode.

### 6.17 Derivative logic

Derivative logic is provided with many members of the PCD33XXA family. The detailed description of the derivative circuitry is given in the data sheet of the specific device. In this section, the shared principles of derivative logic are briefly reviewed.

Derivative registers are accessed over the internal bus. The derivative registers are write-only, read-only or read/write (see Fig.22). They are addressed through the derivative Address Register when the derivative input/output instructions (MOV A, Dx; MOV Dx, A; ANL Dx, A and ORL Dx, A) are executed.

Derivative interrupts share the line PIN with the SIO interrupt (if available). When the derivative interrupt routine is executed, the PIN line must be de-activated by software.

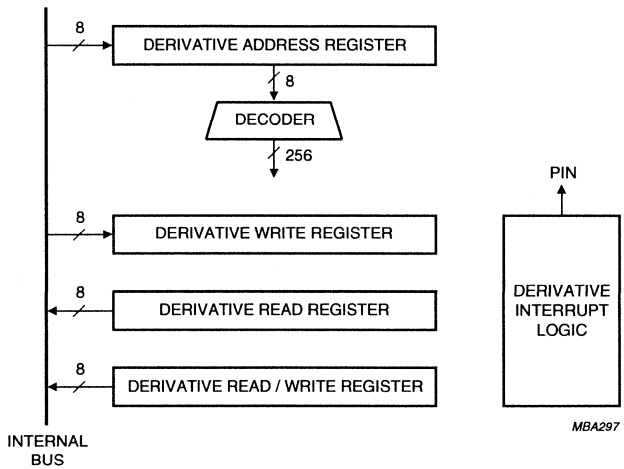



Fig.22 Block diagram of derivative logic.

**Table 7** Summary of mask options.

| FEATURE | OPTION | DESCRIPTION |
|---|---|---|
| ROM | any mix of instructions | program; size restricted by ROM size (see Tables 8 and 9) |
| Ports | option 1 | standard output (see Fig.11) |
| | option 2 | open drain output (see Fig.12) |
| | option 3 | push-pull output (see Fig.13) |
| | set | flip-flop at logic 1 after reset |
| | reset | flip-flop at logic 0 after reset |
| Power-on reference | $V_{ref}$ | 1.2 to 3.6 V in increments of 100 mV; with ±500 mV accuracy |
| Oscillator | $g_{mL}$ | LOW transconductance (see Table 6) |
| | $g_{mM}$ | MEDIUM transconductance (see Table 6) |
| | $g_{mH}$ | HIGH transconductance (see Table 6) |

## 7 INSTRUCTION SET

The PCD33XXA instruction set consists of over 100 one and two-byte instructions. Program code efficiency is high because all RAM locations and all ROM locations on a 256-byte page require only a single-byte address. Table 9 lists the symbols that are used in Table 8 and the Instruction map is shown in Section 7.1.

**Table 8** PCD33XXA family instruction set.

| MNEMONIC | OPCODE (HEX) | BYTES/ CYCLES | DESCRIPTION | FUNCTION | NOTES |
|---|---|---|---|---|---|
| **ACCUMULATOR** | | | | | |
| ADD A, Rr[1] | 6<8 + r> | 1/1 | Add register contents to A | (A)←(A) + (Rr) | r = 0 to 7 |
| ADD A, @Rr[1] | 6r | 1/1 | Add RAM data, addressed by Rr, to A | (A)←(A) + ((Rr)) | r = 0, 1 |
| ADD A, #data[1] | 03 data | 2/2 | Add immediate data to A | (A)←(A) + data | |
| ADDC A, Rr[1] | 7<8 + r> | 1/1 | Add carry and register contents to A | (A)←(A) + (Rr) + (C) | r = 0 to 7 |
| ADDC A, @Rr[1] | 7r | 1/1 | Add carry and RAM data, addressed by Rr, to A | (A)←(A) + ((Rr)) + (C) | r = 0, 1 |
| ADDC A, #data[1] | 13 data | 2/2 | Add carry and immediate data to A | (A)←(A) + data + (C) | |
| ANL A, Rr | 5<8 + r> | 1/1 | AND Rr with A | (A)←(A) AND (Rr) | r = 0 to 7 |
| ANL A, @Rr | 5r | 1/1 | AND RAM data addressed by Rr, with A | (A)←(A) AND ((Rr)) | r = 0, 1 |
| ANL A, #data | 53 data | 2/2 | AND immediate data with A | (A)←(A) AND data | |
| ORL A, Rr | 4<8 + r> | 1/1 | OR Rr with A | (A)←(A) OR (Rr) | r = 0 to 7 |
| ORL A, @Rr | 4r | 1/1 | OR RAM data, addressed by Rr, with A | (A)←(A) OR ((Rr)) | r = 0, 1 |
| ORL A, #data | 43 data | 2/2 | OR immediate data with A | (A)←(A) OR data) | |
| XRL A, Rr | D<8 +r> | 1/1 | XOR Rr with A | (A)←(A) XOR (Rr) | r = 0 to 7 |
| XRL A, @Rr | Dr | 1/1 | XOR RAM data, addressed by Rr, with A | (A)←(A) XOR ((Rr)) | r = 0, 1 |
| XRL A, #data | D3 data | 2/2 | XOR immediate data with A | (A)←(A) XOR data) | |
| INC A | 17 | 1/1 | Increment A by 1 | (A)←(A) + 1 | |
| DEC A | 07 | 1/1 | Decrement A by 1 | (A)←(A) – 1 | |
| CLR A | 27 | 1/1 | Clear A to zero | (A)←0 | |
| CPL A | 37 | 1/1 | One's complement A | (A)←NOT(A) | |
| RL A | E7 | 1/1 | Rotate A left | $(A_{n+1})\leftarrow(A_n)$, $(A_0)\leftarrow(A_7)$ | n = 0 to 6 |
| RLC A[2] | F7 | 1/1 | Rotate A left through carry | $(A_{n+1})\leftarrow(A_n)$, $(A_0)\leftarrow$(C), (C)$\leftarrow(A_7)$ | n = 0 to 6 |
| RR A | 77 | 1/1 | Rotate A right | $(A_n)\leftarrow(A_{n+1})$, $(A_7)\leftarrow(A_0)$ | n = 0 to 6 |
| RRC A[2] | 67 | 1/1 | Rotate A right through carry | $(A_n)\leftarrow(A_{n+1})$, $(A_7)\leftarrow$(C), (C)$\leftarrow(A_0)$ | n = 0 to 6 |
| DA A[2] | 57 | 1/1 | Decimal adjust A | (A)←(A) + 06H if AC = 1 or $(A_{0-3})>9$; (A)←(A) + 60H if $(A_{4-7})>9$ | |
| SWAP A[2] | 47 | 1/1 | Swap nibbles of A | $(A_{4-7})\leftrightarrow(A_{0-3})$ | |

| MNEMONIC | OPCODE (HEX) | BYTES/ CYCLES | DESCRIPTION | FUNCTION | NOTES |
|---|---|---|---|---|---|
| **DATA MOVES** | | | | | |
| MOV A, Rr | F<8 + r> | 1/1 | Move register contents to A | (A)←(Rr) | r = 0 to 7 |
| MOV A, @Rr | Fr | 1/1 | Move RAM data addressed by Rr, to A | (A)←((Rr)) | r = 0, 1 |
| MOV A, #data | 23 data | 2/2 | Move immediate data to A | (A)←data | |
| MOV Rr, A | A<8 + r> | 1/1 | Move Accumulator contents to register | (Rr)←(A) | r = 0 to 7 |
| MOV@Rr, A | Ar | 1/1 | Move Accumulator contents to RAM location addressed by Rr | ((Rr))←(A) | r = 0, 1 |
| MOV Rr, #data | B<8 + r> data | 2/2 | Move immediate data to Rr | (Rr)←data | r = 0 to 7 |
| MOV @Rr, #data | Br data | 2/2 | Move immediate data to RAM location addressed by Rr | ((R0))←data | r = 0, 1 |
| XCH A, Rr | 2<8 + r> | 1/1 | Exchange A contents with Rr | (A)↔(Rr) | r = 0 to 7 |
| XCH A, @Rr | 2r | 1/1 | Exchange Accumulator contents with RAM data addressed by Rr | (A)↔((Rr)) | r = 0, 1 |
| XCHD A, @Rr | 3r | 1/1 | Exchange lower nibbles of A and RAM data addressed by Rr | $(A_{0-3})\leftrightarrow((Rr_{0-3}))$ | r = 0, 1 |
| MOV A, PSW | C7 | 1/1 | Move PSW contents to Accumulator | (A)←(PSW) | |
| MOV PSW, A[3] | D7 | 1/1 | Move Accumulator bit 3 to $PSW_3$ (PS) | (PS)←$(A_3)$ | |
| MOV P A, @A | A3 | 1/2 | Move indirectly addressed data in current page to A | $(PC_{0-7})\leftarrow(A)$, (A)←((PC)) | |
| **CARRY FLAG** | | | | | |
| CLR C[2] | 97 | 1/1 | Clear carry bit | (C)←0 | |
| CPL C[2] | A7 | 1/1 | Complement carry bit | (C)←NOT(C) | |
| **REGISTER** | | | | | |
| INC Rr | 1<8 + r> | 1/1 | Increment register by 1 | (Rr)←(Rr) + 1 | r = 0 to 7 |
| INC @Rr | 1r | 1/1 | Increment RAM data, addressed by Rr, by 1 | ((Rr))←((Rr)) + 1 | r = 0,1 |
| DEC Rr | C<8 + r> | 1/1 | Decrement register by 1 | (Rr)←(Rr) – 1 | r = 0 to 7 |
| DEC @Rr | Cr | 1/1 | Decrement RAM data addressed by Rr, by 1 | ((Rr))←((Rr)) – 1 | r = 0, 1 |

| MNEMONIC | OPCODE (HEX) | BYTES/ CYCLES | DESCRIPTION | FUNCTION | NOTES |
|---|---|---|---|---|---|
| **BRANCH** | | | | | |
| JMP addr | <2n>4 addr | 2/2 | Unconditional jump within a 2 kbyte bank | ($PC_{8-10}$)←n ($PC_{0-7}$)←addr ($PC_{11-12}$)← (MBFF0-1) | n = 0 to 7 |
| JMPP @A | B3 | 1/2 | Indirect jump within a page | ($PC_{0-7}$)←((A)) | |
| DJZN Rr, addr | E<8 + r> addr | 2/2 | Decrement Rr by 1 and jump if not zero to addr | (Rr)←(Rr) – 1; if (Rr) not zero, then ($PC_{0-7}$)←addr | r = 0 to 7 |
| DJNZ @Rr, addr | Er | 2/2 | Decrement RAM data, addressed by Rr, by 1 and jump if not zero to addr | ((Rr))←((Rr)) – 1; if ((Rr)) not zero, then ($PC_{0-7}$)←addr | r = 0 to 1 |
| JBb addr | <2b + 1> 2 addr | 2/2 | Jump to addr if Accumulator bit b = 1 | If ($A_b$) = 1, then ($PC_{0-7}$)←addr | b = 0 to 7 |
| JC addr | F6 addr | 2/2 | Jump to addr if C = 1 | If (C) = 1, then ($PC_{0-7}$)←addr | |
| JNC addr | E6 addr | 2/2 | Jump to addr if C = 0 | If (C) = 0, then ($PC_{0-7}$)←addr | |
| JZ addr | C6 addr | 2/2 | Jump to addr if A = 0 | If (A) = 0, ($PC_{0-7}$)←addr | |
| JNZ addr | 96 addr | 2/2 | Jump to addr if A is NOT zero | If (A) ≠ 0, then ($PC_{0-7}$)←addr | |
| JT0 addr | 36 addr | 2/2 | Jump to addr if T0 = 1 | If T0 = 1, then ($PC_{0-7}$)←addr | |
| JNT0 addr | 26 addr | 2/2 | Jump to addr if T0 = 0 | If T0 = 0, then ($PC_{0-7}$)←addr | |
| JT1 addr | 56 addr | 2/2 | Jump to addr if T1 = 1 | If T1 = 1, then ($PC_{0-7}$)←addr | |
| JNT1 addr | 46 addr | 2/2 | Jump to addr if T1 = 0 | If T0 = 0, then ($PC_{0-7}$)←addr | |
| JTF addr[4] | 16 addr | 2/2 | Jump to addr if Timer Flag = 1 | If TF = 1, then ($PC_{0-7}$)←addr | |
| JNTF addr[4] | 06 addr | 2/2 | Jump to addr if Timer Flag = 0 | If T0 = 0, then ($PC_{0-7}$)←addr | |

| MNEMONIC | OPCODE (HEX) | BYTES/ CYCLES | DESCRIPTION | FUNCTION | NOTES |
|---|---|---|---|---|---|
| **TIMER/EVENT COUNTER** | | | | | |
| MOV A,T | 42 | 1/1 | Move timer/event counter contents to A | (A)←(T) | |
| MOV T, A | 62 | 1/1 | Move A contents to timer/event counter | (T)←(A) | |
| STRT CNT | 45 | 1/1 | Start event counter | | |
| STRT T | 55 | 1/1 | Start timer | | |
| STOP TCNT | 65 | 1/1 | Stop timer/event counter | | |
| EN TCNTI | 25 | 1/1 | Enable timer/event counter interrupt | | |
| DIS TCNTI | 35 | 1/1 | Disable timer/event counter interrupt | | |
| **CONTROL** | | | | | |
| EN I | 05 | 1/1 | Enable external (chip enable) interrupt | | |
| DIS I | 15 | 1/1 | Disable external (chip enable) interrupt | | |
| SEL RB0[5] | C5 | 1/1 | Select Register Bank 0 | (RBS)←0 | |
| SEL RB1[5] | D5 | 1/1 | Select Register Bank 1 | (RBS)←1 | |
| SEL MB0[10] | E5 | 1/1 | Select program memory bank 0 | (MBFF0)←0, (MBFF1)←0 | |
| SEL MB1[10] | F5 | 1/1 | Select program memory bank 1 | (MBFF0)←1, (MBFF1)←0 | |
| SEL MB2[10] | A5 | 1/1 | Select program memory bank 2 | (MBFF0)←0, (MBFF1)←1 | |
| SEL MB3[10] | B5 | 1/1 | Select program memory bank 3 | (MBFF0)←1, (MBFF1)←1 | |
| STOP | 22 | 1/1 | Enter Stop mode | | |
| IDLE | 01 | 1/1 | Enter Idle mode | | |
| **SUBROUTINE** | | | | | |
| CALL addr[6] | <2n + 1> 4addr | 2/2 | Jump to subroutine | ((SP))←(PC) ($PSW_{4,6,7}$), (SP)←(SP) + 1, ($PC_{8-10}$)←n, ($PC_{0-7}$)←addr, ($PC_{11-12}$) ←(MBFF0-1) | n = 0 to 7 |
| RET[6] | 83 | 1/2 | Return from subroutine | (SP)←(SP) – 1, (PC)←((SP)) | |
| RETR[6] | 93 | 1/2 | Return from interrupt and restore bits 4, 6 and 7 of PSW | (SP)←(SP) – 1, ($PSW_{4,6,7}$) + (PC)←((SP)) | |

| MNEMONIC | OPCODE (HEX) | BYTES/ CYCLES | DESCRIPTION | FUNCTION | NOTES |
|---|---|---|---|---|---|
| **PARALLEL INPUT/OUTPUT** | | | | | |
| IN A, P0 | 08 | 1/2 | Input Port 0 data to Accumulator | (A)←(P0) | |
| IN A, P1 | 09 | 1/2 | Input Port 1 data to Accumulator | (A)←(P1) | |
| IN A, P2[7] | 0A | 1/2 | Input Port 2 data to Accumulator | (A)←(P2) | |
| OUTL P0, A | 38 | 1/2 | Output A data to Port 0 | (P0)←(A) | |
| OUTL P1, A | 39 | 1/2 | Output A data to Port 1 | (P1)←(A) | |
| OUTL P2, A | 3A | 1/2 | Output A data to Port 2 | (P2)←(A) | |
| ANL P0, #data | 98 data | 2/2 | AND Port 0 data with immediate data | (P0)←(P0) AND data | |
| ANL P1, #data | 99 data | 2/2 | AND Port 1 data with immediate data | (P1)←(P1) AND data | |
| ANL P2, #data | 9A data | 2/2 | AND Port 2 data with immediate data | (P2)←(P2) AND data | |
| ORL P0, #data | 88 data | 2/2 | OR Port 0 data with immediate data | (P0)←(P0) OR data | |
| ORL P1, #data | 89 data | 2/2 | OR Port 1 data with immediate data | (P1)←(P1) OR data | |
| ORL P2, #data | 8A data | 2/2 | OR Port 2 data with immediate data | (P2)←(P2) OR data | |
| **DERIVATIVE INPUT/OUTPUT** | | | | | |
| MOV A, Dx[8] | 8C direct | 2/2 | Move derivative register contents to A | (A)←(Dx) | x = 0 to 255 |
| MOV Dx, A[8] | 8D direct | 2/2 | Move A contents to derivative register | (Dx)←(A) | x = 0 to 255 |
| ANL Dx, A[8] | 8E direct | 2/2 | AND derivative register with A | (Dx)←(Dx) AND (A) | x = 0 to 255 |
| ORL Dx, A[8] | 8F direct | 2/2 | OR derivative register with A | (Dx)←(Dx) OR (A) | x = 0 to 255 |
| **SERIAL INPUT/OUTPUT** | | | | | |
| MOV A, S0 | 0C | 1/2 | Move serial I/O register 0 contents to A | (A)←(S0) | |
| MOV A, S1[9] | 0D | 1/2 | Move serial I/O register 1 contents to A | (A)←(S1) | |
| MOV S0, A | 3C | 1/2 | Move A contents to serial I/O register 0 | (S0)←(A) | |
| MOV S1, A[9] | 3D | 1/2 | Move A contents to serial I/O register 1 | (S1)←(A) | |
| MOV S2, A | 3E | 1/2 | Move A contents to serial I/O register 2 | (S2)←(A) | |
| MOV S0, #data | 9C data | 2/2 | Move immediate data to serial I/O register 0 | (S0)←data | |
| MOV S1, #data[9] | 9D data | 2/2 | Move immediate data to serial I/O register 1 | (S1)←data | |
| MOV S2, #data | 9E data | 2/2 | Move immediate data to serial I/O register 2 | (S2)←data | |
| EN SI | 85 | 1/1 | Enable serial I/O interrupt | | |
| DIS SI | 95 | 1/1 | Disable serial I/O interrupt | | |
| NOP | 00 | 1/1 | No operation | $(PC_{0\text{-}10})\leftarrow (PC_{0\text{-}10}) + 1$ | |

**Notes to Table 8**

1. PSW CY, AC affected.
2. PSW CY affected.
3. PSW PS affected.
4. Execution of a JTF or JNTF instruction resets the Timer Flag (TF).
5. PSW RBS affected.
6. PSW $SP_0$, $SP_1$ and $SP_2$, affected.
7. (A) = 0000, P2.3, P2.2, P2.1 and P2.0.
8. For more information on the derivative I/O instructions of a particular microcontroller, consult the specific microcontroller data sheet.
9. (S1) has a different meaning for read and write operations. See Section 6.11.4.
10. SEL MB instructions may not be used within interrupt routines.

**Table 9** Definitions of symbols used in Table 8.

| SYMBOL | DESCRIPTION |
|---|---|
| A | Accumulator |
| AC | auxiliary (half) carry |
| addr | program memory address |
| Bb | bit designation (b = 0 to 7) |
| CE/T0 | CE/T0 input |
| CY | carry bit |
| Dx | mnemonic derivative register |
| data | 8-bit number or expression |
| MB0 | program memory bank 0 |
| MB1 | program memory bank 1 |
| MB2 | program memory bank 2 |
| MB3 | program memory bank 3 |
| MBFF0 | memory bank flip-flop 0 |
| MBFF1 | memory bank flip-flop 1 |
| PC | Program Counter |
| PS | timer prescaler select |
| PSW | Program Status Word |
| RB0 | Register Bank 0 |
| RB1 | Register Bank 1 |
| RBS | Register Bank Select |
| Rr | register designation (r = 0 to 7) |
| SPn | Stack Pointer (n = 0, 1 or 2) |
| T | Timer 1 |
| T1 | T1 input |
| TF | Timer Flag |
| x | derivative register address (x = 0 to 255) |
| (X) | contents of X |
| ((X)) | contents of location addressed by X |
| ← | is replaced by |
| ↔ | is exchanged with |
| # | immediate data prefix |
| @ | indirect address prefix |
| * | hexadecimal; 8...F selects R0...R7 |
| & | hexadecimal; 0, 2, 4, 6, 8, A, C, E selects page 0...7 in JMP, i.e. $(PC_{8\text{-}10}) \leftarrow \&_{1\text{-}3}$ |
| % | hexadecimal; 1, 3, 5, 7, 9, B, D, F selects page 0...7 in CALL, i.e. $(PC_{8\text{-}10}) \leftarrow \&_{1\text{-}3}$ selects bit b = 0...7 in JBb, i.e. $b = \&_{1\text{-}3}$ |

### 7.1 Instruction map

first hexadecimal character of opcode — second hexadecimal character of opcode

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | NOP | IDLE | | ADD A, #data | JMP page 0 | EN I | JNTF addr | DEC A | IN A,Pp 0 | IN A,Pp 1 | IN A,Pp 2 | | MOV A,Sn 0 | MOV A,Sn 1 | | |
| 1 | INC @ Rr 0 | INC @ Rr 1 | JB0 addr | ADDC A,#data | CALL page 0 | DIS I | JTF addr | INC A | INC Rr 0 | INC Rr 1 | INC Rr 2 | INC Rr 3 | INC Rr 4 | INC Rr 5 | INC Rr 6 | INC Rr 7 |
| 2 | XCH A, @Rr 0 | XCH A, @Rr 1 | STOP | MOV A, #data | JMP page 1 | EN TCNTI | JNT0 addr | CLR A | XCH A,Rr 0 | XCH A,Rr 1 | XCH A,Rr 2 | XCH A,Rr 3 | XCH A,Rr 4 | XCH A,Rr 5 | XCH A,Rr 6 | XCH A,Rr 7 |
| 3 | XCHD A, @Rr 0 | XCHD A, @Rr 1 | JB1 addr | | CALL page 1 | DIS TCNTI | JT0 addr | CPL A | OUTL Pp,A 0 | OUTL Pp,A 1 | OUTL Pp,A 2 | | MOV Sn,A 0 | MOV Sn,A 1 | MOV Sn,A 2 | |
| 4 | ORL A, @Rr 0 | ORL A, @Rr 1 | MOV A, T | ORL A, #data | JMP page 2 | STRT CNT | JNT1 addr | SWAP A | ORL A,Rr 0 | ORL A,Rr 1 | ORL A,Rr 2 | ORL A,Rr 3 | ORL A,Rr 4 | ORL A,Rr 5 | ORL A,Rr 6 | ORL A,Rr 7 |
| 5 | ANL A, @Rr 0 | ANL A, @Rr 1 | JB2 addr | ANL A, #data | CALL page 2 | STRT T | JT1 addr | DA A | ANL A,Rr 0 | ANL A,Rr 1 | ANL A,Rr 2 | ANL A,Rr 3 | ANL A,Rr 4 | ANL A,Rr 5 | ANL A,Rr 6 | ANL A,Rr 7 |
| 6 | ADD A, @Rr 0 | ADD A, @Rr 1 | MOV T, A | | JMP page 3 | STOP TCNT | | RRC A | ADD A,Rr 0 | ADD A,Rr 1 | ADD A,Rr 2 | ADD A,Rr 3 | ADD A,Rr 4 | ADD A,Rr 5 | ADD A,Rr 6 | ADD A,Rr 7 |
| 7 | ADDC A, @Rr 0 | ADDC A, @Rr 1 | JB3 addr | | CALL page 3 | | | RR A | ADDC A,Rr 0 | ADDC A,Rr 1 | ADDC A,Rr 2 | ADDC A,Rr 3 | ADDC A,Rr 4 | ADDC A,Rr 5 | ADDC A,Rr 6 | ADDC A,Rr 7 |
| 8 | | | | RET | JMP page 4 | EN SI | | | ORL Pp,#data 0 | ORL Pp,#data 1 | ORL Pp,#data 2 | | MOV A,Dx | MOV Dx,A | ANL Dx,A | ORL Dx,A |
| 9 | | | JB4 addr | RETR | CALL page 4 | DIS SI | JNZ addr | CLR C | ANL Pp,#data 0 | ANL Pp,#data 1 | ANL Pp,#data 2 | | MOV Sn,#data 0 | MOV Sn,#data 1 | MOV Sn,#data 2 | |
| A | MOV @ Rr,A 0 | MOV @ Rr,A 1 | | MOVP A,@A | JMP page 5 | SEL MB2 | | CPL C | MOV Rr,A 0 | MOV Rr,A 1 | MOV Rr,A 2 | MOV Rr,A 3 | MOV Rr,A 4 | MOV Rr,A 5 | MOV Rr,A 6 | MOV Rr,A 7 |
| B | MOV @Rr, #data 0 | MOV @Rr, #data 1 | JB5 addr | JMPP @A | CALL page 5 | SEL MB3 | | | MOV Rr,#data 0 | MOV Rr,#data 1 | MOV Rr,#data 2 | MOV Rr,#data 3 | MOV Rr,#data 4 | MOV Rr,#data 5 | MOV Rr,#data 6 | MOV Rr,#data 7 |
| C | DEC @Rr 0 | DEC @Rr 1 | | | JMP page 6 | SEL RB0 | JZ addr | MOV A,PSW | DEC Rr 0 | DEC Rr 1 | DEC Rr 2 | DEC Rr 3 | DEC Rr 4 | DEC Rr 5 | DEC Rr 6 | DEC Rr 7 |
| D | XRL A, @Rr 0 | XRL A, @Rr 1 | JB6 addr | XRL A,#data | CALL page 6 | SEL RB1 | | MOV PSW,A | XRL A,Rr 0 | XRL A,Rr 1 | XRL A,Rr 2 | XRL A,Rr 3 | XRL A,Rr 4 | XRL A,Rr 5 | XRL A,Rr 6 | XRL A,Rr 7 |
| E | DJNZ @ Rr,addr 0 | DJNZ @ Rr,addr 1 | | | JMP page 7 | SEL MB0 | JNC addr | RL A | DJNZ Rr,addr 0 | DJNZ Rr,addr 1 | DJNZ Rr,addr 2 | DJNZ Rr,addr 3 | DJNZ Rr,addr 4 | DJNZ Rr,addr 5 | DJNZ Rr,addr 6 | DJNZ Rr,addr 7 |
| F | MOV A, @Rr 0 | MOV A, @Rr 1 | JB7 addr | | CALL page 7 | SEL MB1 | JC addr | RLC A | MOV A,Rr 0 | MOV A,Rr 1 | MOV A,Rr 2 | MOV A,Rr 3 | MOV A,Rr 4 | MOV A,Rr 5 | MOV A,Rr 6 | MOV A,Rr 7 |

MBA281

## THE 8048 BASED INSTRUCTION SET

### Introduction

This chapter describes the instruction set of the 8048 based microcontrollers. The following terms are used throughout the chapter:

8048: refers to the MAB8048, MAB8049, MAB8050, and PCF80C49 microcontrollers.
84XX: refers to the MAB84XX family of microcontrollers.
84CXXX: refers to the PCF84CXXX family of microcontrollers.
33XX: refers to the PCD33XX family of microcontrollers.

The chapter is split into the following sections:

- Instructions common to all 8048 based microcontrollers; this section describes instructions that are common to all microcontrollers based on the 8048 instruction set. For full details on the instruction set of a particular microcontroller family, this section should be read in conjunction with the section that details the additional instructions which are specific to that microcontroller family.

- Additional 8048 instructions; this section describes the 8048 instructions which are not included in the common section; it should be used in conjunction with the common section.

- Additional 84XX instructions; this section describes the 84XX instructions which are not included in the common section; it should be used in conjunction with the common section.

- Additional 84CXXX instructions; this section describes the 84CXXX instructions which are not included in the common section; it should be used in conjunction with the common section.

- Additional 33XX instructions; this section describes the 33XX instructions which are not included in the common section; it should be used in conjunction with the common section.

Within the above sections, the instruction sets are described in alphabetical order. Some of the sections contain details on devices whose instruction set deviates from the family to which they belong. Any other differences are described in the data sheets.

Each instruction is introduced by its mnemonic followed by a descriptive title. The hexadecimal opcode is presented together with a byte pattern or binary equivalent. A description of the logical operation performed follows, and simple examples are given for most of the instructions.

Table 1 details the symbols and abbreviations used throughout the chapter. Table 2 gives a summary of **all** of the instructions. Note that a particular device will use only a subset of the instructions shown in Table 2.

**Table 1** Symbols and abbreviations.

| SYMBOL | DESCRIPTION |
|---|---|
| A | accumulator |
| AC | auxiliary carry |
| addr | program memory address |
| Bb | bit designation (b = 0 to 7) |
| BUS | BUS port |
| C | carry (bit CY) |
| CNT | event counter |
| D | mnemonic for a 4-bit digit (nibble) |
| Dx | mnemonic for a derivative register (x = 0 to 255) |
| direct | 8-bit derivative register address |
| data | 8-bit immediate data |
| F0, F1 | Flag 0, Flag 1 |
| H | hexadecimal data |
| I | interrupt |
| MBn | memory bank (n = 0 to 3) |
| MBFFn | memory bank flip-flop (n = 0 or 1) |
| P | mnemonic for 'in-page' operation |
| PC | program counter |
| Pp | port designation (p = 0, 1, or 2) |
| PS | timer prescaler select |
| PSW | program status word |
| RB | register bank |
| RBS | register bank select flag |
| @Rr | 8-bit address register (r = 0, 1) |
| Rr | 8-bit register (r = 0 to 7) |
| Sn | serial I/O register (n = 0, 1, or 2) |
| SP | stack pointer |
| T | timer |
| TCNT | timer/event counter |
| TF | timer flag |
| T0, T1 | test 0 and test 1 inputs |
| # | immediate data prefix |
| @ | indirect address prefix |
| (X) | contents of X |
| ((X)) | contents of location addressed by X |
| ← | is replaced by |
| ↔ | is exchanged with |
| [x] | x represents a hex digit |

**Table 2** Instruction set summary.

| MNEMONIC | OPCODE (HEX) | BYTES/ CYCLES | DESCRIPTION | FUNCTION | NOTES | |
|---|---|---|---|---|---|---|
| **ACCUMULATOR** | | | | | | |
| ADD A,Rr | 6[8+r] | 1/1 | Add register contents to A | (A) ← (A)+(Rr) | r = 0 to 7 | 1 |
| ADD A,@Rr | 6r | 1/1 | Add RAM data to A | (A) ← (A)+((Rr)) | r = 0, 1 | 1 |
| ADD A,#data | 03 data | 2/2 | Add immediate data to A | (A) ← (A)+data | | 1 |
| ADDC A,Rr | 7[8+r] | 1/1 | Add carry and register contents to A | (A) ← (A)+(Rr)+(C) | r = 0 to 7 | 1 |
| ADDC A,@Rr | 7r | 1/1 | Add carry and RAM data to A | (A) ← (A)+((Rr))+(C) | r = 0, 1 | 1 |
| ADDC A,#data | 13 data | 2/2 | Add carry and immediate data to A | (A) ← (A)+data+(C) | | 1 |
| ANL A,Rr | 5[8+r] | 1/1 | 'AND' Rr with A | (A) ← (A) AND (Rr) | r = 0 to 7 | |
| ANL A,@Rr | 5r | 1/1 | 'AND' RAM data with A | (A) ← (A) AND ((Rr)) | r = 0, 1 | |
| ANL A,#data | 53 data | 2/2 | 'AND' immediate data with A | (A) ← (A) AND data | | |
| ORL A,Rr | 4[8+r] | 1/1 | 'OR' Rr with A | (A) ← (A) OR (Rr) | r = 0 to 7 | |
| ORL A,@Rr | 4r | 1/1 | 'OR' RAM data with A | (A) ← (A) OR ((Rr)) | r = 0, 1 | |
| ORL A,#data | 43 data | 2/2 | 'OR' immediate data with A | (A) ← (A) OR data | | |
| XRL A,Rr | D[8+r] | 1/1 | 'XOR' Rr with A | (A) ← (A) XOR (Rr) | r = 0 to 7 | |
| XRL A,@Rr | Dr | 1/1 | 'XOR' RAM data with A | (A) ← (A) XOR ((Rr)) | r = 0, 1 | |
| XRL A,#data | D3 data | 2/2 | 'XOR' immediate data with A | (A) ← (A) XOR data | | |
| INC A | 17 | 1/1 | Increment A by 1 | (A) ← (A)+1 | | |
| DEC A | 07 | 1/1 | Decrement A by 1 | (A) ← (A)–1 | | |
| CLR A | 27 | 1/1 | Clear A to zero | (A) ← 0 | | |
| CPL A | 37 | 1/1 | One's complement A | (A) ← NOT(A) | | |
| RL A | E7 | 1/1 | Rotate A left | $(A_{n+1}) \leftarrow (A_n)$<br>$(A_0) \leftarrow (A_7)$ | n = 0 to 6 | |
| RLC A | F7 | 1/1 | Rotate A left through carry | $(A_{n+1}) \leftarrow A_n$<br>$(A_0) \leftarrow$ (C)<br>(C) ← $(A_7)$ | n = 0 to 6 | 2 |
| RR A | 77 | 1/1 | Rotate A right | $(A_n) \leftarrow (A_{n+1})$<br>$(A_7) \leftarrow (A_0)$ | n = 0 to 6 | |
| RRC A | 67 | 1/1 | Rotate A right through carry | $(A_n) \leftarrow (A_{n+1})$<br>$(A_7) \leftarrow$ (C)<br>(C) ← $(A_0)$ | n = 0 to 6 | 2 |
| DA A | 57 | 1/1 | Decimal adjust A | If AC = 1 or $(A_{0-3}) > 9$, then (A) ← (A)+06H<br>If C = 1 or $(A_{4-7}) > 9$, then (A) ← (A)+60H | | 2 |
| SWAP A | 47 | 1/1 | Swap nibbles of A | $(A_{4-7}) \leftrightarrow (A_{0-3})$ | | 2 |

| MNEMONIC | OPCODE (HEX) | BYTES/ CYCLES | DESCRIPTION | FUNCTION | NOTES | |
|---|---|---|---|---|---|---|
| **DATA MOVES** | | | | | | |
| MOV A,Rr | F[8+r] | 1/1 | Move register contents to A | (A) ← (Rr) | r = 0 to 7 | |
| MOV A,@Rr | Fr | 1/1 | Move RAM data to A | (A) ← ((Rr)) | r = 0, 1 | |
| MOV A,#data | 23 data | 2/2 | Move immediate data to A | (A) ← data | | |
| MOV Rr,A | A[8+r] | 1/1 | Move A contents to register | (Rr) ← (A) | r = 0 to 7 | |
| MOV @Rr,A | Ar | 1/1 | Move A contents to RAM | ((Rr)) ← (A) | r = 0, 1 | |
| MOV Rr,#data | B[8+r] data | 2/2 | Move immediate data to Rr | (Rr) ← data | r = 0 to 7 | |
| MOV @Rr,#data | Br data | 2/2 | Move immediate data to RAM | ((Rr)) ← data | r = 0, 1 | |
| XCH A,Rr | 2[8+r] | 1/1 | Exchange A contents with Rr | (A) ↔ (Rr) | r = 0 to 7 | |
| XCH A,@Rr | 2r | 1/1 | Exchange A contents with RAM data | (A) ↔ ((Rr)) | r = 0, 1 | |
| XCHD A,@Rr | 3r | 1/1 | Exchange lower nibbles of A and RAM data | $(A_{0-3}) \leftrightarrow ((Rr_{0-3}))$ | r = 0, 1 | |
| MOV A,PSW | C7 | 1/1 | Move PSW contents to A | (A) ← (PSW) | | |
| MOV PSW,A | D7 | 1/1 | Move A contents to PSW | (PSW) ← (A) | | a, 10 |
| MOV PSW,A | D7 | 1/1 | Move A bit 3 content to PSW bit 3 | $(PS) \leftarrow (A_3)$ | | b, c, d, 3 |
| MOVP A,@A | A3 | 1/2 | Move indirectly addressed data in current page to A | $(PC_{0-7}) \leftarrow (A)$, (A) ← ((PC)) | | |
| MOVX A,@Rr | 8r | 1/2 | Move external RAM data to A | (A) ← ((Rr)) | r = 0, 1 | a |
| MOVX @Rr,A | 9r | 1/2 | Move A contents to external RAM location | ((Rr)) ← (A) | r = 0, 1 | a |
| MOVP3 A,@A | E3 | 1/2 | Move page 3 data to A | $(PC_{0-7}) \leftarrow (A)$ $(PC_{8-11}) \leftarrow 1100$ (A) ← ((PC)) | | a |
| **FLAGS** | | | | | | |
| CLR C | 97 | 1/1 | Clear carry bit | (C) ← 0 | | 2 |
| CPL C | A7 | 1/1 | Complement carry bit | (C) ← NOT(C) | | 2 |
| CLR F0 | 85 | 1/1 | Clear flag 0 | (F0) ← 0 | | a |
| CPL F0 | 95 | 1/1 | Complement flag 0 | (F0) ← NOT(F0) | | a |
| CLR F1 | A5 | 1/1 | Clear flag 1 | (F1) ← 0 | | a |
| CPL F1 | B5 | 1/1 | Complement flag 1 | (F1) ← NOT(F1) | | a |
| **REGISTER** | | | | | | |
| INC Rr | 1[8+r] | 1/1 | Increment register by 1 | (Rr) ← (Rr)+1 | r = 0 to 7 | |
| INC @Rr | 1r | 1/1 | Increment RAM data by 1 | ((Rr)) ← ((Rr))+1 | r = 0, 1 | |
| DEC Rr | C[8+r] | 1/1 | Decrement register by 1 | (Rr) ← (Rr)–1 | r = 0 to 7 | |
| DEC @Rr | Cr | 1/1 | Decrement RAM data by 1 | ((Rr)) ← ((Rr))–1 | r = 0, 1 | b, c, d |

| MNEMONIC | OPCODE (HEX) | BYTES/ CYCLES | DESCRIPTION | FUNCTION | NOTES |
|---|---|---|---|---|---|
| **BRANCH** | | | | | |
| JMP addr | [2n]4 addr | 2/2 | Unconditional jump | $(PC_{8-10}) \leftarrow n$<br>$(PC_{0-7}) \leftarrow addr$<br>$(PC_{11}) \leftarrow (MBFF)$ | n = 0 to 7 a |
| JMP addr | [2n]4 addr | 2/2 | Unconditional jump | $(PC_{8-10}) \leftarrow n$<br>$(PC_{0-7}) \leftarrow addr$<br>$(PC_{11-12}) \leftarrow$ (MBFF0-1) | n = 0 to 7 b, c, d |
| JMPP @A | B3 | 1/2 | Indirect jump within a page | $(PC_{0-7}) \leftarrow ((A))$ | |
| DJNZ Rr,addr | E[8+r] addr | 2/2 | Decrement Rr by 1 and jump if not zero to addr | (Rr) ← (Rr)–1;<br>if (Rr) not zero, then<br>$(PC_{0-7}) \leftarrow addr$ | r = 0 to 7 |
| DJNZ @Rr,addr | Er | 2/2 | Decrement RAM data by 1 and jump if not zero to addr | ((Rr)) ← ((Rr))–1;<br>if ((Rr)) not zero,<br>then $(PC_{0-7}) \leftarrow addr$ | r = 0, 1 b, c, d |
| JBb addr | [2b+1]2 addr | 2/2 | Jump to addr if A bit b = 1 | If $(A_b) = 1$, then<br>$(PC_{0-7}) \leftarrow addr$ | b = 0 to 7 |
| JC addr | F6 addr | 2/2 | Jump to addr if C = 1 | If (C) = 1, then<br>$(PC_{0-7}) \leftarrow addr$ | |
| JNC addr | E6 addr | 2/2 | Jump to addr if C = 0 | If (C) = 0, then<br>$(PC_{0-7}) \leftarrow addr$ | |
| JZ addr | C6 addr | 2/2 | Jump to addr if A = 0 | If (A) = 0, then<br>$(PC_{0-7}) \leftarrow addr$ | |
| JNZ addr | 96 addr | 2/2 | Jump to addr if A is NOT zero | If $(A) \neq 0$, then<br>$(PC_{0-7}) \leftarrow addr$ | |
| JT0 addr | 36 addr | 2/2 | Jump to addr if T0 = 1 | If T0 = 1, then<br>$(PC_{0-7}) \leftarrow addr$ | a, b, c |
| JT0 addr | 36 addr | 2/2 | Jump to addr if T0 = 0 | If T0 = 0, then<br>$(PC_{0-7}) \leftarrow addr$ | d |
| JNT0 addr | 26 addr | 2/2 | Jump to addr if T0 = 0 | If T0 = 0, then<br>$(PC_{0-7}) \leftarrow addr$ | a, b, c |
| JNT0 addr | 26 addr | 2/2 | Jump to addr if T0 = 1 | If T0 = 1, then<br>$(PC_{0-7}) \leftarrow addr$ | d |
| JT1 addr | 56 addr | 2/2 | Jump to addr if T1 = 1 | If T1 = 1, then<br>$(PC_{0-7}) \leftarrow addr$ | |
| JNT1 addr | 46 addr | 2/2 | Jump to addr if T1 = 0 | If T1 = 0, then<br>$(PC_{0-7}) \leftarrow addr$ | |
| JTF addr | 16 addr | 2/2 | Jump to addr if Timer Flag = 1 | If TF = 1, then<br>$(PC_{0-7}) \leftarrow addr$ | 4 |
| JNTF addr | 06 addr | 2/2 | Jump to addr if Timer Flag = 0 | If TF = 0, then<br>$(PC_{0-7}) \leftarrow addr$ | b, c, d, 4 |
| JF0 addr | B6 addr | 2/2 | Jump to addr if flag 0 = 1 | If F0 = 1, then<br>$(PC_{0-7}) \leftarrow addr$ | a |
| JF1 addr | 76 addr | 2/2 | Jump to addr if flag 1 = 1 | If F1 = 1, then<br>$(PC_{0-7}) \leftarrow addr$ | a |
| JNI addr | 86 addr | 2/2 | Jump to addr if interrupt input is LOW | If I = 0, then<br>$(PC_{0-7}) \leftarrow addr$ | a |

| MNEMONIC | OPCODE (HEX) | BYTES/ CYCLES | DESCRIPTION | FUNCTION | NOTES |
|---|---|---|---|---|---|
| **TIMER/EVENT COUNTER** | | | | | |
| MOV A,T | 42 | 1/1 | Move timer/event counter contents to A | (A) ← (T) | |
| MOV T,A | 62 | 1/1 | Move A contents to timer/ event counter | (T) ← (A) | |
| STRT CNT | 45 | 1/1 | Start event counter | | |
| STRT T | 55 | 1/1 | Start timer | | |
| STOP TCNT | 65 | 1/1 | Stop timer/event counter | | |
| EN TCNTI | 25 | 1/1 | Enable timer/event counter interrupt | | |
| DIS TCNTI | 35 | 1/1 | Disable timer/event counter interrupt | | |
| **CONTROL** | | | | | |
| ENT0 CLK | 75 | 1/1 | Enable clock output | | a |
| EN I | 05 | 1/1 | Enable external interrupt | | |
| DIS I | 15 | 1/1 | Disable external interrupt | | |
| SEL RB0 | C5 | 1/1 | Select register bank 0 | (RBS) ← 0 | 5 |
| SEL RB1 | D5 | 1/1 | Select register bank 1 | (RBS) ← 1 | 5 |
| SEL MB0 | E5 | 1/1 | Select program memory bank 0 | (MBFF) ← 0 | a, 9 |
| SEL MB0 | E5 | 1/1 | Select program memory bank 0 | (MBFF0) ← 0, (MBFF1) ← 0 | b, c, d, 9 |
| SEL MB1 | F5 | 1/1 | Select program memory bank 1 | (MBFF) ← 1 | a, 9 |
| SEL MB1 | F5 | 1/1 | Select program memory bank 1 | (MBFF0) ← 1, (MBFF1) ← 0 | b, c, d, 9 |
| SEL MB2 | A5 | 1/1 | Select program memory bank 2 | (MBFF0) ← 0, (MBFF1) ← 1 | b, c, d, 9 |
| SEL MB3 | B5 | 1/1 | Select program memory bank 3 | (MBFF0) ← 1, (MBFF1) ← 1 | b, c, d, 9 |
| STOP | 22 | 1/1 | Enter Stop mode | | c, d |
| IDLE | 01 | 1/1 | Enter Idle mode | | c, d |
| IDLE | 01 | 1/2 | Enter Idle mode | | e |
| **SUBROUTINE** | | | | | |
| CALL addr | [2n+1]4 addr | 2/2 | Jump to subroutine | ((SP)) ← (PC), ($PSW_{4-7}$) (SP) ← (SP)+1 ($PC_{8-10}$) ← n ($PC_{0-7}$) ← addr ($PC_{11}$) ← (MBFF) | n = 0 to 7 a, 6 |
| CALL addr | [2n+1]4 addr | 2/2 | Jump to subroutine | ((SP)) ← (PC), ($PSW_{4,6,7}$) (SP) ← (SP)+1 ($PC_{8-10}$) ← n ($PC_{0-7}$) ← addr ($PC_{11-12}$) ← (MBFF0-1) | n = 0 to 7 b, c, d, 6 |
| RET | 83 | 1/2 | Return from subroutine | (SP) ← (SP)–1 (PC) ← ((SP)) | 6 |
| RETR | 93 | 1/2 | Return from interrupt and restore bits 4-7 of PSW | (SP) ← (SP)–1 ($PSW_{4-7}$)+(PC) ← ((SP)) | a, 6 |
| RETR | 93 | 1/2 | Return from interrupt and restore bits 4, 6, 7 of PSW | (SP) ← (SP)–1 ($PSW_{4,6,7}$)+(PC) ← ((SP)) | b, c, d, 6 |

| MNEMONIC | OPCODE (HEX) | BYTES/ CYCLES | DESCRIPTION | FUNCTION | NOTES |
|---|---|---|---|---|---|
| **INPUT/OUTPUT** | | | | | |
| IN A,P0 | 08 | 1/2 | Input port 0 data to A | (A) ← (P0) | b,c,d |
| IN A,P1 | 09 | 1/2 | Input port 1 data to A | (A) ← (P1) | |
| IN A,P2 | 0A | 1/2 | Input port 2 data to A | (A) ← (P2) | 7 |
| OUTL P0,A | 38 | 1/2 | Output A data to port 0 | (P0) ← (A) | b,c,d |
| OUTL P1,A | 39 | 1/2 | Output A data to port 1 | (P1) ← (A) | |
| OUTL P2,A | 3A | 1/2 | Output A data to port 2 | (P2) ← (A) | |
| ANL P0, #data | 98 data | 2/2 | AND port 0 with immediate data | (P0) ← (P0) AND data | b,c,d |
| ANL P1, #data | 99 data | 2/2 | AND port 1 with immediate data | (P1) ← (P1) AND data | |
| ANL P2, #data | 9A data | 2/2 | AND port 2 with immediate data | (P2) ← (P2) AND data | |
| ORL P0, #data | 88 data | 2/2 | OR port 0 data with immediate data | (P0) ← (P0) OR data | b,c,d |
| ORL P1, #data | 89 data | 2/2 | OR port 1 data with immediate data | (P1) ← (P1) OR data | |
| ORL P2, #data | 8A data | 2/2 | OR port 2 data with immediate data | (P2) ← (P2) OR data | |
| INS A,BUS | 08 | 1/2 | Strobed input of BUS data to A | (A) ← (BUS) | a |
| OUTL BUS,A | 02 | 1/2 | Output A data to BUS | (BUS) ← (A) | a |
| ANL BUS, #data | 98 | 2/2 | AND BUS port with immediate data | (BUS) ← (BUS) AND data | a |
| ORL BUS, #data | 88 | 2/2 | OR BUS with immediate data | (BUS) ← (BUS) OR data | a |
| MOVD A,Pp | 0C | 1/2 | Move port 4-7 data to A | $A_{(0-3)}$ ← (P4); $A_{(4-7)}$ ← 0 | a |
| | 0D | | | $A_{(0-3)}$ ← (P5); $A_{(4-7)}$ ← 0 | a |
| | 0E | | | $A_{(0-3)}$ ← (P6); $A_{(4-7)}$ ← 0 | a |
| | 0F | | | $A_{(0-3)}$ ← (P7); $A_{(4-7)}$ ← 0 | a |
| MOVD Pp,A | 3C | 1/2 | Move A to port 4-7 | P4 ← $A_{(0-3)}$ | a |
| | 3D | | | P5 ← $A_{(0-3)}$ | a |
| | 3E | | | P6 ← $A_{(0-3)}$ | a |
| | 3F | | | P7 ← $A_{(0-3)}$ | a |
| ANLD Pp,A | 9C | 1/2 | AND port 4-7 data with A | (P4) ← (P4) AND $A_{(0-3)}$ | a |
| | 9D | | | (P5) ← (P5) AND $A_{(0-3)}$ | a |
| | 9E | | | (P6) ← (P6) AND $A_{(0-3)}$ | a |
| | 9F | | | (P7) ← (P7) AND $A_{(0-3)}$ | a |
| ORLD Pp,A | 8C | 1/2 | OR port 4-7 data with A | (P4) ← (P4) OR $A_{(0-3)}$ | a |
| | 8D | | | (P5) ← (P5) OR $A_{(0-3)}$ | a |
| | 8E | | | (P6) ← (P6) OR $A_{(0-3)}$ | a |
| | 8F | | | (P7) ← (P7) OR $A_{(0-3)}$ | a |

| MNEMONIC | OPCODE (HEX) | BYTES/ CYCLES | DESCRIPTION | FUNCTION | NOTES |
|---|---|---|---|---|---|
| **DERIVATIVE INPUT/OUTPUT** | | | | | |
| MOV A,Dx | 8C direct | 2/2 | Move derivative register contents to A | (A) ← (Dx) | x = 0 to 255 c, d, 8 |
| MOV Dx,A | 8D direct | 2/2 | Move A contents to derivative register | (Dx) ← (A) | x = 0 to 255 c, d, 8 |
| ANL Dx,A | 8E direct | 2/2 | AND derivative register with A | (Dx) ← (Dx) AND (A) | x = 0 to 255 c, d, 8 |
| ORL Dx,A | 8F direct | 2/2 | OR derivative register with A | (Dx) ← (Dx) OR (A) | x = 0 to 255 c, d, 8 |
| **SERIAL INPUT/OUTPUT** | | | | | |
| MOV A,$S_n$ | 0C<br>0D | 1/2 | Move serial I/O register contents to A | (A) ← (S0)<br>(A) ← (S1) | n = 0, 1 b, c, d |
| MOV $S_n$,A | 3C<br>3D<br>3E | 1/2 | Move A contents to serial I/O register | (S0) ← (A)<br>(S1) ← (A)<br>(S2) ← (A) | n = 0, 1, 2 b, c, d |
| MOV $S_n$,#data | 9C data<br>9D data<br>9E data | 2/2 | Move immediate data to serial I/O register | (S0) ← data<br>(S1) ← data<br>(S2) ← data | n = 0, 1, 2 b, c, d |
| EN SI | 85 | 1/1 | Enable serial I/O interrupt | | b, c, d |
| DIS SI | 95 | 1/1 | Disable serial I/O interrupt | | b, c, d |
| NOP | 00 | 1/1 | No operation | ($PC_{0-10}$) ← ($PC_{0-10}$)+1 | |

**Notes**

a. 8048 only.
b. 84XX only.
c. 84CXXX only.
d. 33XX only.
e. 80C49 only.

1. PSW CY, AC affected.
2. PSW CY affected.
3. PSW PS affected.
4. Execution of a JTF or JNTF instruction resets the Timer Flag (TF).
5. PSW RBS affected.
6. PSW $SP_0$, $SP_1$, $SP_2$ affected.
7. (A) = 0000, P2.3, P2.2, P2.1, P2.0 for 84CXXX and 33XX; for 84XX, (A) = 1111, P2.3, P2.2, P2.1, P2.0.
8. For more information on the derivative input/output instructions of a particular microcontroller, consult the specific microcontroller data sheet.
9. SEL MB instructions may not be used within interrupt routines.
10. All PSW bits affected.

### INSTRUCTIONS COMMON TO ALL 8048 BASED MICROCONTROLLERS

This section describes instructions that are common to all microcontrollers based on the 8048 instruction set. For full details on the instruction set of a particular microcontroller family, this section should be read in conjunction with the section that details the additional instructions which are specific to that microcontroller family.

## ADD A,#data — Add immediate data to accumulator

Encoding | 0 0 0 0 | 0 0 1 1 | $d_7$ $d_6$ $d_5$ $d_4$ | $d_3$ $d_2$ $d_1$ $d_0$ | 03H

Description: This is a 2-cycle instruction. The specified data is added to the accumulator. The carry and half carry flags are affected.

Operation: (A) ← (A) + data

Example: ADD A,#ADDLE — ADD VALUE OF SYMBOL 'ADDLE' TO ACC

## ADD A,Rr — Add register contents to accumulator

Encoding | 0 1 1 0 | 1 r r r | 68H-6FH

Description: The contents of working register 'r' are added to the accumulator. The carry and half carry flags are affected.

Operation: (A) ← (A) + (Rr) — r = 0-7

Example: ADD A,R6 — ADD REG 6 CONTENTS TO ACC

## ADD A,@Rr — Add data memory contents to accumulator

Encoding | 0 1 1 0 | 0 0 0 r | 60H-61H

Description: The contents of the resident data memory location addressed by working register 'r' are added to the accumulator. The carry and half carry flags are affected.

Operation: (A) ← (A) + ((Rr)) — r = 0-1

Example:
```
MOV R0,#01FH    MOVE '1F' HEX TO REG 0
ADD A,@R0       ADD VALUE OF LOCATION 31 TO ACC
```

## ADDC A,#data — Add carry and immediate data to accumulator

Encoding: | 0 0 0 1 | 0 0 1 1 | | $d_7$ $d_6$ $d_5$ $d_4$ | $d_3$ $d_2$ $d_1$ $d_0$ | 13H

Description: This is a 2-cycle instruction. The content of the carry bit is added to accumulator location 0 and the carry bit is cleared. Then the specified data is added to the accumulator. The carry and half carry flags are affected.

Operation: (A) ← (A) + (C) + data

Example: ADDC A,#225 — ADD CARRY AND '255' DEC TO ACC

## ADDC A,Rr — Add carry and register contents to accumulator

Encoding: | 0 1 1 1 | 1 r r r | 78H-7FH

Description: The content of the carry bit is added to accumulator location 0 and the carry bit is cleared. The contents of working register 'r' are then added to the accumulator. The carry and half carry flags are affected.

Operation: (A) ← (A) + (C) + (Rr) — r = 0-7

Example: ADDC A,R5 — ADD CARRY AND REG 5 CONTENTS TO ACC

## ADDC A,@Rr — Add carry and data memory contents to accumulator

Encoding: | 0 1 1 1 | 0 0 0 r | 70H-71H

Description: The content of the carry bit is added to accumulator location 0 and the carry bit is cleared. Then the contents of the resident data memory location addressed by working register 'r' are added to the accumulator. The carry and half carry flags are affected.

Operation: (A) ← (A) + (C) + ((Rr)) — r = 0-1

Example:

```
MOV R1,#40      MOVE '40' DEC TO REG 1
ADDC A,@R1      ADD CARRY AND LOCATION 40 CONTENTS TO ACC
```

## ANL A,#data — Logical AND accumulator with immediate mask

Encoding: 0 1 0 1 | 0 0 1 1 — $d_7$ $d_6$ $d_5$ $d_4$ | $d_3$ $d_2$ $d_1$ $d_0$ — 53H

Description: This is a 2-cycle instruction. Data in the accumulator is logical ANDed with an immediately-specified mask.

Operation: (A) ← (A) AND data

Example:

| | |
|---|---|
| ANL A,#0AFH | 'AND' ACC CONTENTS WITH MASK 1010 1111 |
| ANL A,#3 + X/Y | 'AND' ACC CONTENTS WITH VALUE OF EXPRESSION '3 + X/Y' |

## ANL A,Rr — Logical AND accumulator with register mask

Encoding: 0 1 0 1 | 1 r r r — 58H-5FH

Description: Data in the accumulator is logically ANDed with the mask contained in working register 'r'.

Operation: (A) ← (A) AND (Rr) — r = 0-7

Example: AND A,R4 — 'AND' ACC CONTENTS WITH MASK IN REG 4

## ANL A,@Rr — Logical AND accumulator with memory mask

Encoding: 0 1 0 1 | 0 0 0 r — 50H-51H

Description: Data in the accumulator is logically ANDed with the mask contained in data memory location referenced by register 'r'.

Operation: (A) ← (A) AND ((Rr)) — r = 0-1

Example:

| | |
|---|---|
| MOV R0,#03FH | MOVE '3F' HEX TO REG 0 |
| ANL A,@R0 | 'AND' ACC CONTENTS WITH MASK IN LOCATIONS 63 |

## ANL Pp,#data — Logical AND port with immediate mask

Encoding: 1 0 0 1 | 1 0 p p — $d_7$ $d_6$ $d_5$ $d_4$ | $d_3$ $d_2$ $d_1$ $d_0$ — 98H-9AH

Description: This is a 2-cycle instruction. Data on port 'p' is logically ANDed with an immediately-specified mask.

Operation: (Pp) ← (Pp) AND data — p = 0-2; P0 is not used by 8048

Example: ANL P1,#0F0H — 'AND' PORT 1 CONTENTS WITH MASK 'F0' HEX (CLEAR P1.0-P1.3)

## CALL address — Subroutine call

Encoding

| $a_{10}$ $a_9$ $a_8$ 1 | 0 1 0 0 |
|---|---|

| $a_7$ $a_6$ $a_5$ $a_4$ | $a_3$ $a_2$ $a_1$ $a_0$ |
|---|---|

Description

This is a 2-cycle instruction. The program counter and 3 or 4 PSW bits are saved in the stack. The stack pointer is updated (PSW bits 0-2). Program control is then passed to the location specified by 'address'. The status of PC bit 11 (and bit 12, if present) is determined by the most recent SEL MB instruction.

A CALL cannot commence from locations 2046-2047, 4094-4095 or 6142-6143. Execution continues at the instruction following the CALL upon return from the subroutine.

Operation

**8048**

$((SP)) \leftarrow (PC), (PSW_{4-7})$
$(SP) \leftarrow (SP) + 1$
$(PC_{8-10}) \leftarrow addr_{8-10}$
$(PC_{0-7}) \leftarrow addr_{0-7}$
$(PC_{11}) \leftarrow MBFF$

**84XX, 84CXXX, 33XX**

$((SP)) \leftarrow (PC), (PSW_{4,6,7})$
$(SP) \leftarrow (SP) + 1$
$(PC_{8-10}) \leftarrow addr_{8-10}$
$(PC_{0-7}) \leftarrow addr_{0-7}$
$(PC_{11-12}) \leftarrow MBFF_{0-1}$

Example

Add three groups of two numbers. Put subtotals in locations 50, 51 and total in location 52.

```
        MOV R0,#50      MOVE '50' DEC TO ADDRESS REG 0
        MOV A,R1        MOVE CONTENTS OF REG 1 TO ACC
        ADD A,R2        ADD REG 2 TO ACC
        CALL SUBTOT     CALL SUBROUTINE 'SUBTOT'
        ADDC A,R3       ADD REG 3 TO ACC
        ADDC A,R4       ADD REG 4 TO ACC
        CALL SUBTOT     CALL SUBROUTINE 'SUBTOT'
        ADDC A,R5       ADD REG 5 TO ACC
        ADDC A,R6       ADD REG 6 TO ACC
        CALL SUBTOT     CALL SUBROUTINE 'SUBTOT'

SUBTOT  MOV @R0,A       MOVE CONTENTS OF ACC TO LOCATION
                        ADDRESSED BY REG 0
        INC R0          INCREMENT REG 0
        RET             RETURN TO MAIN PROGRAM
```

## CLR A — Clear accumulator

Encoding

| 0 0 1 0 | 0 1 1 1 |
|---|---|

27H

Description

The contents of the accumulator are cleared to zero.

Operation

$(A) \leftarrow 0$

## CLR C — Clear carry bit

Encoding | 1 0 0 1 | 0 1 1 1 | 97H

Description: During normal program execution, the carry bit can be set to one by the ADD, ADDC, CPL C, RLC, RRC, and DAA instructions. This instruction resets the carry bit to zero.

Operation: (C) ← 0

## CPL A — Complement accumulator

Encoding | 0 0 1 1 | 0 1 1 1 | 37H

Description: The contents of the accumulator are complemented. This is strictly a one's complement. Each zero is changed to one and vice-versa.

Operation: (A) ← NOT (A)

Example: Assume accumulator contains 1001 0101.

```
CPL A          ACC CONTENTS ARE COMPLEMENTED TO
               0110 1010
```

## CPL C — Complement carry bit

Encoding | 1 0 1 0 | 0 1 1 1 | A7H

Description: The carry bit is complemented; a zero is changed to one, and vice-versa.

Operation: (C) ← NOT (C)

Example: Set C to one; current status is unknown.

```
CLR C          C IS CLEARED TO ZERO
CPL C          C IS SET TO ONE
```

## DA A — Decimal adjust accumulator

Encoding

| 0 1 0 1 | 0 1 1 1 |
|---|---|

57H

Description

The 8-bit accumulator value is adjusted to form two 4-bit Binary Coded Decimal (BCD) digits following the binary addition of BCD numbers. The carry bit C is affected. If the contents of bits 0-3 are greater than nine, or if AC is one, the accumulator is incremented by six.

The four high-order bits are then checked. If bits 4-7 exceed nine, of if C is one, these bits are increased by six. If an overflow occurs, C is set to one.

Example

Assume accumulator contains 1001 1011.

```
DA A                          ACC ADJUSTED TO 0000 0001 WITH C SET

C  AC 7 6 5 4 3 2 1 0
0  0  1 0 0 1 1 0 1 1
             0 1 1 0          ADD SIX TO BITS 0-7
0  0  1 0 1 0 0 0 0 1
      0 1 1 0                 ADD SIX TO BITS 4-7
1  0  0 0 0 0 0 0 0 1         OVERFLOW TO C
```

## DEC A — Decrement accumulator

Encoding

| 0 0 0 0 | 0 1 1 1 |
|---|---|

07H

Description

The contents of the accumulator are decremented by one.

Operation

(A) ← (A) −1

Example

Decrement the contents of external data memory location 63.

```
MOV R0,#3FH     MOVE '3F' HEX TO REG 0
MOV A,@R0       MOVE CONTENTS OF LOCATION 63 TO ACC
DEC A           DECREMENT ACC
MOV @R0,A       MOVE CONTENTS OF ACC TO LOCATION 63 IN
                EXTERNAL MEMORY
```

## DEC Rr — Decrement register

| | | |
|---|---|---|
| Encoding | 1 1 0 0 \| 1 r r r | C8H-CFH |
| Description | The contents of working register 'r' are decremented by one. | |
| Operation | (Rr) ← (Rr) –1 | r = 0-7 |
| Example: | DEC R1 | DECREMENT CONTENTS OF REG 1 |

## DIS I — Disable external interrupt

| | | |
|---|---|---|
| Encoding | 0 0 0 1 \| 0 1 0 1 | 15H |
| Description | External interrupts are disabled. A LOW signal on the interrupt input has no effect. | |

## DIS TCNTI — Disable timer/counter interrupt

| | | |
|---|---|---|
| Encoding | 0 0 1 1 \| 0 1 0 1 | 35H |
| Description | Timer/counter interrupts are disabled. Any pending timer interrupt request is cleared. The interrupt sequence is not initiated by an overflow, but the timer flag is set and time accumulation continues. | |

## DJNZ Rr,addr — Decrement register and test

Encoding

| 1 1 1 0 | 1 r r r |
|---|---|

| $a_7$ $a_6$ $a_5$ $a_4$ | $a_3$ $a_2$ $a_1$ $a_0$ |
|---|---|

E8H-EFH

Description

This is a 2-cycle instruction. Register 'r' is decremented, then tested for zero. If the register contains all zeros, program control passes to the next instruction. If the register contents are not zero, program control jumps to the specified 'address'.

The address in this case must evaluate to 8-bits (the jump must be to a location within the current 256-location page).

Operation

(Rr) ← (Rr) –1 r = 0-7
If Rr not 0
then ($PC_{0-7}$) ← addr
else (PC) ← (PC) + 2

**Note:** A 12-bit address specification does not cause an error if the DJNZ instruction and the jump target address are on the same page. If the DJNZ instruction begins in location 255 of a page, it must jump to a target address on the following page.

Example

Increment values in data memory locations 50-54.

```
        MOV R0,#50          MOVE '50' DEC TO ADDRESS REG 0
        MOV R3,#5           MOVE '5' DEC TO COUNTER REG 3
INCRT   INC @R0             INCREMENT CONTENTS OF LOCATION
                            ADDRESSED BY REG 0
        INC R0              INCREMENT ADDRESS IN REG 0
        DJNZ R3,INCRT       INCREMENT REG 3 - JUMP TO 'INCRT' IF REG 3
                            NONZERO
        NEXT ---            'NEXT' ROUTINE EXECUTED IF R3 IS ZERO
```

## EN I — Enable external interrupt

Encoding

| 0 0 0 0 | 0 1 0 1 |
|---|---|

05H

Description

External interrupts are enabled.

## EN TCNTI — Enable timer/counter interrupt

Encoding

| 0 0 1 0 | 0 1 0 1 |
|---|---|

25H

Description

Timer/counter interrupts are enabled. An overflow of the timer/counter initiates the interrupt sequence.

## IN A,Pp — Input port or data to accumulator

Encoding | 0 0 0 0 | 1 0 p p | 08H-0AH

Description: This is a 2-cycle instruction. Data present on Port 'p' is transferred (read) to the accumulator.

Operation: (A) ← (Pp) p = 0-2; P0 is not used by 8048

Example:

```
IN A,P1     INPUT PORT 1 CONTENTS TO ACC
MOV R6,A    MOVE ACC CONTENTS TO REG 6
IN A,P2     INPUT PORT 2 CONTENTS TO ACC
MOV R7,A    MOVE ACC CONTENTS TO REG 7
```

## INC A — Increment accumulator

Encoding | 0 0 0 1 | 0 1 1 1 | 17H

Description: Accumulator contents are incremented by one. Carry is not affected.

Operation: (A) ← (A) + 1

Example: Increment contents of the accumularor by 2.

```
INC A       INCREMENT ACC
INC A       INCREMENT ACC AGAIN
```

## INC Rr — Increment register

Encoding | 0 0 0 1 | 1 r r r | 18H-1FH

Description: The contents of working register 'r' are incremented by one.

Operation: (Rr) ← (Rr) + 1 r = 0-7

Example:

```
INC R0      INCREMENT ADDRESS REG 0
```

## INC @Rr — Increment data memory location

Encoding: 0 0 0 1 | 0 0 0 r — 10H-11H

Description: The contents of the resident data memory location addressed by register 'r' are incremented by one.

Operation: ((Rr)) ← ((Rr)) + 1 — r = 0-1

Example:
```
MOV R1,#03FH    MOVE '3F' HEX TO REG 1
INC @R1         INCREMENT LOCATION 63
```

## JBb address — Jump if accumulator bit is set

Encoding: $b_2\ b_1\ b_0\ 1 \mid 0\ 0\ 1\ 0$ — $a_7\ a_6\ a_5\ a_4 \mid a_3\ a_2\ a_1\ a_0$ — 12H, 32H, 52H, 72H, 92H, B2H, D2H, F2H

Description: This is a 2-cycle instruction. Control passes to the specified address if accumulator bit 'b' is set to one.

Operation:
$(PC_{0\text{-}7}) \leftarrow$ addr — If Bb = 1
(PC) ← (PC) + 2 — If Bb = 0

Example:
```
JB4 NEXT        JUMP TO 'NEXT' ROUTINE IF ACC BIT 4 IS ONE
```

**Note:** If this instruction begins in location 255 of a page, the target address is located in the following page.

## JC address — Jump if carry is set

Encoding: 1 1 1 1 | 0 1 1 0 — $a_7\ a_6\ a_5\ a_4 \mid a_3\ a_2\ a_1\ a_0$ — F6H

Description: This is a 2-cycle instruction. Control passes to the specified address if the carry bit is set to one.

Operation:
$(PC_{0\text{-}7}) \leftarrow$ addr — If C = 1
(PC) ← (PC) + 2 — If C = 0

Example:
```
JC OVFLOW       JUMP TO 'OVFLOW' ROUTINE IF CARRY IS ONE
```

**Note:** If this instruction begins in location 255 of a page, the target address is located in the following page.

## JMP address — Direct jump within 2K block

Encoding

| $a_{10}$ $a_9$ $a_8$ 0 | 0 1 0 0 |
|---|---|

| $a_7$ $a_6$ $a_5$ $a_4$ | $a_3$ $a_2$ $a_1$ $a_0$ |
|---|---|

Description

This is a 2-cycle instruction. Bits 0-10 of the program counter are replaced with the directly-specified address. The setting of PC bit 11 (and bit 12, if present) is determined by the most recent SEL MB instruction.

Operation

| **8048** | **84XX, 84CXXX, 33XX** |
|---|---|
| $(PC_{0-7}) \leftarrow$ addr 0-7 | $(PC_{0-7}) \leftarrow$ addr 0-7 |
| $(PC_{8-10}) \leftarrow$ addr 8-10 | $(PC_{8-10}) \leftarrow$ addr 8-10 |
| $(PC_{11}) \leftarrow$ MBFF | $(PC_{11-12}) \leftarrow MBFF_{0-1}$ |

Example

| | |
|---|---|
| JMP SUBTOT | JUMP TO SUBROUTINE 'SUBTOT' |
| JMP $-6 | JUMP TO INSTRUCTION SIX LOCATIONS BEFORE CURRENT LOCATION |
| JMP 2FH | JUMP TO ADDRESS '2F' HEX |

## JMPP @A — Indirect jump within page

Encoding

| 1 0 1 1 | 0 0 1 1 |
|---|---|

B3H

Description

This is a 2-cycle instruction. The contents of the program memory location pointed to by the accumulator are substituted for the 'page' portion of the program counter (PC bits 0-7)

Operation

$(PC_{0-7}) \leftarrow ((A))$

Example

Assume accumulator contains 0FH.

| | |
|---|---|
| JMPP @A | JUMP TO ADDRESS STORED IN LOCATION 15 IN CURRENT PAGE |

## JNC address — Jump if carry is not set

Encoding

| 1 1 1 0 | 0 1 1 0 |
|---|---|

| $a_7$ $a_6$ $a_5$ $a_4$ | $a_3$ $a_2$ $a_1$ $a_0$ |
|---|---|

E6H

Description

This is a 2-cycle instruction. Control passes to the specified address if the carry bit is not set (equals zero).

Operation

| | |
|---|---|
| $(PC_{0-7}) \leftarrow$ addr | If C = 0 |
| $(PC) \leftarrow (PC) + 2$ | If C = 1 |

Example

| | |
|---|---|
| JNC NOVFLO | JUMP TO 'NOVFLO' ROUTINE IS CARRY IS ZERO |

**Note:** If this instruction begins in location 255 of a page, the target address is located in the following page.

## JNT0 address — Jump if Test 0 input is LOW

| | | | |
|---|---|---|---|
| Encoding | 0 0 1 0 \| 0 1 1 0 | $a_7$ $a_6$ $a_5$ $a_4$ \| $a_3$ $a_2$ $a_1$ $a_0$ | 26H |

Description: This is a 2-cycle instruction. Control passes to the specified address if the Test 0 input signal is LOW (HIGH for 33XX) when this instruction is executed. Otherwise, program control passes to the next instruction.

| | | |
|---|---|---|
| Operation | $(PC_{0-7}) \leftarrow$ addr | If T0 = 0 (if T0 = 1 for 33XX) |
| | (PC) $\leftarrow$ (PC) + 2 | If T0 = 1 (if T0 = 0 for 33XX) |
| Example | JNT0 60 | JUMP TO LOCATION 60 DEC IF T0 IS LOW (HIGH for 33XX) |

**Note:** If this instruction begins in location 255 of a page, the target address is located in the following page.

## JNT1 address — Jump if Test 1 input is LOW

| | | | |
|---|---|---|---|
| Encoding | 0 1 0 0 \| 0 1 1 0 | $a_7$ $a_6$ $a_5$ $a_4$ \| $a_3$ $a_2$ $a_1$ $a_0$ | 46H |

Description: This is a 2-cycle instruction. Control passes to the specified address if the Test 1 signal is LOW.

| | | |
|---|---|---|
| Operation | $(PC_{0-7}) \leftarrow$ addr | If T1 = 0 |
| | (PC) $\leftarrow$ (PC) + 2 | If T1 = 1 |
| Example | JNT1 110H | JUMP TO LOCATION '110' HEX IF T1 IS LOW |

**Note:** If this instruction begins in location 255 of a page, the target address is located in the following page.

## JNZ address — Jump if accumulator is not zero

Encoding: 1 0 0 1 | 0 1 1 0 | $a_7$ $a_6$ $a_5$ $a_4$ | $a_3$ $a_2$ $a_1$ $a_0$ — 96H

Description: This is a 2-cycle instruction. Control passes to the specified address if the accumulator contents are non-zero when this instruction is executed. Otherwise, program control passes to the next instruction.

Operation:

| | |
|---|---|
| $(PC_{0-7}) \leftarrow$ addr | If $A \neq 0$ |
| $(PC) \leftarrow (PC) + 2$ | If $A = 0$ |

Example:

| | |
|---|---|
| JNZ ACCNOT | JUMP TO ROUTINE 'ACCNOT' IF ACC VALUE IS NON-ZERO |

**Note:** If this instruction begins in location 255 of a page, the target address is located in the following page.

## JT0 address — Jump if Test 0 input is HIGH

Encoding: 0 0 1 1 | 0 1 1 0 | $a_7$ $a_6$ $a_5$ $a_4$ | $a_3$ $a_2$ $a_1$ $a_0$ — 36H

Description: This is a 2-cycle instruction. Control passes to the specified address, if the Test 0 input pin is HIGH (LOW for 33XX) when this instruction is executed. Otherwise, program control passes to the next instruction.

Operation:

| | |
|---|---|
| $(PC_{0-7}) \leftarrow$ addr | If T0 = 1 (if T0 = 0 for 33XX) |
| $(PC) \leftarrow (PC) + 2$ | If T0 = 0 (if T0 = 1 for 33XX) |

Example:

| | |
|---|---|
| JT0 COUNT | JUMP TO 'COUNT' ROUTINE IF T0 IS HIGH (LOW for 33XX) |

**Note:** If this instruction begins in location 255 of a page, the target address is located in the following page.

## JT1 address — Jump if Test 1 input is HIGH

Encoding

| 0 1 0 1 | 0 1 1 0 | $a_7$ $a_6$ $a_5$ $a_4$ | $a_3$ $a_2$ $a_1$ $a_0$ | 56H |
|---|---|---|---|---|

Description

This is a 2-cycle instruction. Control passes to the specified address, if the Test 1 input pin is HIGH when this instruction is executed. Otherwise, program control passes to the next instruction.

Operation

| | |
|---|---|
| $(PC_{0-7}) \leftarrow$ addr | If T1 = 1 |
| $(PC) \leftarrow (PC) + 2$ | If T1 = 0 |

Example

| | |
|---|---|
| JT1 COUNT | JUMP TO 'COUNT' ROUTINE IF T1 IS HIGH |

**Note:** If this instruction begins in location 255 of a page, the target address is located in the following page.

## JTF address — Jump if timer flag is set

Encoding

| 0 0 0 1 | 0 1 1 0 | $a_7$ $a_6$ $a_5$ $a_4$ | $a_3$ $a_2$ $a_1$ $a_0$ | 16H |
|---|---|---|---|---|

Description

This is a 2-cycle instruction. Control passes to the specified address if the timer flag is set to one, that is, the timer/counter has overflowed. Otherwise, program control passes to the next instruction. Testing the timer flag resets it to zero. (This overflow initiates an interrupt service sequence if the timer-overflow interrupt is enabled).

Operation

| | |
|---|---|
| $(PC_{0-7}) \leftarrow$ addr | If TF = 1 |
| $(PC) \leftarrow (PC) + 2$ | If TF = 0 |

Example

| | |
|---|---|
| JTF TIMER | JUMP TO TIMER ROUTINE IF THE TIMER HAS OVERFLOWED (TF=1) |

**Note:** If this instruction begins in location 255 of a page, the target address is located in the following page.

## JZ address — Jump if accumulator is zero

Encoding

| 1 1 0 0 | 0 1 1 0 | $a_7$ $a_6$ $a_5$ $a_4$ | $a_3$ $a_2$ $a_1$ $a_0$ | C6H |
|---|---|---|---|---|

Description

This is a 2-cycle instruction. Control passes to the specified address if the accumulator contains all zeros when this instruction is executed. Otherwise, program control passes to the next instruction.

Operation

| | |
|---|---|
| $(PC_{0-7}) \leftarrow$ addr | If A = 0 |
| $(PC) \leftarrow (PC) + 2$ | If A $\neq$ 0 |

Example

| | |
|---|---|
| JZ 0A3H | JUMP TO LOCATION 'A3' HEX IF ACC VALUE IS ZERO |

## MOV A,#data — Move immediate data to accumulator

| | | | |
|---|---|---|---|
| Encoding | 0 0 1 0 \| 0 0 1 1 | $d_7$ $d_6$ $d_5$ $d_4$ \| $d_3$ $d_2$ $d_1$ $d_0$ | 23H |
| Description | This is a 2-cycle instruction. The 8-bit value specified by 'data' is loaded in the accumulator. | | |
| Operation | (A) ← data | | |
| Example | MOV A,#0A3H | MOVE 'A3' HEX TO ACC | |

## MOV A,PSW — Move PSW contents to accumulator

| | | |
|---|---|---|
| Encoding | 1 1 0 0 \| 0 1 1 1 | C7H |
| Description | The contents of the program status word are moved to the accumulator. | |
| Operation | (A) ← (PSW) | |
| Example | Jump to the 'RBISET' routine if PSW bank switch, bit 4, is set. | |
| | MOV A,PSW | MOVE PSW CONTENTS TO ACC |
| | JB4 RBISET | JUMP TO 'RBISET' IF ACC BIT 4 = 1 |

## MOV A,Rr — Move register contents to accumulator

| | | | |
|---|---|---|---|
| Encoding | 1 1 1 1 \| 1 r r r | | F8H-FFH |
| Description | 8-bits of data are moved from working register 'r' into the accumulator. | | |
| Operation | (A) ← (Rr) | r = 0-7 | |
| Example | MOV A,R3 | MOVE CONTENTS OF REG 3 TO ACC | |

## MOV A,@Rr — Move data memory contents to accumulator

Encoding

| 1 1 1 1 | 0 0 0 r |
|---|---|

F0H-F1H

Description

The contents of the resident data memory location addressed by working register 'r' are moved to the accumulator. Register 'r' contents are unaffected.

Operation

(A) ← ((Rr)) r = 0-1

Example

Assume R1 contains 0011 0110

```
MOV A,@R1        MOVE CONTENTS OF DATA MEM LOCATION 54
                 TO ACC
```

## MOV A,T — Move timer/counter contents to accumulator

Encoding

| 0 1 0 0 | 0 0 1 0 |
|---|---|

42H

Description

The contents of the timer/event counter register are moved to the accumulator.

Operation

(A) ← (T)

Example

Jump to 'EXIT' if the timer has reached or exceeded '100'.

```
CLR C            CLEAR CARRY
MOV A,T          MOVE TIMER CONTENTS TO ACC
ADD A,#156       ADD 156 DEC TO ACC
JC EXIT          JUMP TO 'EXIT' ROUTINE IF CARRY IS SET
```

## MOV PSW,A — Move accumulator contents to PSW

Encoding | 1 1 0 1 | 0 1 1 1 | D7H

**8048**

Description: The contents of the accumulator are moved into the program status word. All condition bits and the stack pointer are affected by this move.

Operation: (PSW) ← (A)

Example: Move up stack pointer by two memory locations, i.e. increment the pointer by one.

```
MOV A,PSW      MOVE PSW CONTENTS TO ACC
INC A          INCREMENT ACC BY ONE
MOV PSW,A      MOVE ACC CONTENTS TO PSW
```

**84XX, 84CXXX, 33XX**

Description: The content of accumulator bit 3 is moved into the prescaler switch.

Operation: (PS) ← ($A_3$)

Example: Set the timer to 'module 1' mode.

```
MOV A,#08H     SET ACC BIT 3 TO LOGIC ONE
MOV PSW,A      SET PS (= PSW3) TO LOGIC 1
```

## MOV Rr,#data — Move immediate data to register

Encoding | 1 0 1 1 | 1 r r r | $d_7$ $d_6$ $d_5$ $d_4$ | $d_3$ $d_2$ $d_1$ $d_0$ | B8H-BFH

Description: This is a 2-cycle instruction. The 8-bit value specified by 'data' is moved to register 'r'.

Operation: (Rr) ← data  r = 0-7

Example:

```
MOV R4,#HEXTEN     THE VALUE OF THE SYMBOL 'HEXTEN' IS
                   MOVED INTO REG 4
MOV R5,#P1*(R*R)   THE VALUE OF THE EXPRESSION 'P1*(R*R)' IS
                   MOVED INTO REG 5
MOV R3,#0ADH       'AD' HEX IS MOVED INTO REG 3
```

## MOV @Rr,#data — Move immediate data to data memory

Encoding: | 1 0 1 1 | 0 0 0 r | | $d_7$ $d_6$ $d_5$ $d_4$ | $d_3$ $d_2$ $d_1$ $d_0$ | B0H-B1H

Description: This is a 2-cycle instruction. The 8-bit value specified by 'data' is moved to the resident data memory location addressed by register Rr.

Operation: ((Rr)) ← data — r = 0-1

Example: Move the hexadecimal value AC3F to locations 61-62.

```
MOV R0,#61        MOVE '61' DEC TO ADDR REG 0
MOV @R0,#0ACH     MOVE 'AC' HEX TO LOCATION 61
INC R0            INCREMENT REG 0 TO '62'
MOV @R0,#3FH      MOVE '3F' HEX TO LOCATION 62
```

## MOV Rr,A — Move accumulator contents to register

Encoding: | 1 0 1 0 | 1 r r r | A8H-AFH

Description: The contents of the accumulator are moved to working register 'r'.

Operation: (Rr) ← (A) — r = 0-7

Example:

```
MOV R0,A          MOVE CONTENTS OF ACC TO REG 0
```

## MOV @Rr,A — Move accumulator contents to data memory

Encoding: | 1 0 1 0 | 0 0 0 r | A0H-A1H

Description: The contents of the accumulator are moved to the resident data memory location whose address is specified by register Rr. Register Rr contents are unaffected.

Operation: ((Rr)) ← A — r = 0-1

Example: Assume R0 contains 0000 0111.

```
MOV @R0,A         MOVE CONTENTS OF ACC TO LOCATION 7
                  (REG 7)
```

## MOV T,A — Move accumulator contents to timer/counter

Encoding: 0 1 1 0 | 0 0 1 0 — 62H

Description: The contents of the accumulator are moved to the timer/event counter register.

Operation: (T) ← (A)

Example: Initialize and start event counter.

```
CLR A       CLEAR ACC TO ZEROS
MOV T,A     MOVE ZEROS TO EVENT COUNTER
STRT CNT    START COUNTER
```

## MOVP A,@A — Move current page data to accumulator

Encoding: 1 0 1 0 | 0 0 1 1 — A3H

Description: This is a 2-cycle instruction. The contents of the program memory location addressed by the accumulator are moved to the accumulator. Only bits 0-7 of the program counter are affected, limiting the program memory reference to the current page. The program counter is restored following this operation.

Operation:
$(PC_{0-7}) \leftarrow (A)$
$(A) \leftarrow ((PC))$

**Note:** This is a 1-byte, 2-cycle instruction. If it appears in location 255 of a program memory page, @A addresses a location in the following page.

Example:

```
MOV A,#128  MOVE '128' DEC TO ACC
MOVP A,@A   CONTENTS OF 129th LOCATION IN CURRENT
            PAGE ARE MOVED TO ACC
```

## NOP — The NOP instruction

Encoding: 0 0 0 0 | 0 0 0 0 — 00H

Description: No operation is performed. Execution continues with the next instruction.

## ORL A,#data — Logical OR accumulator with immediate mask

| | | | |
|---|---|---|---|
| Encoding | 0 1 0 0 \| 0 0 1 1 | $d_7$ $d_6$ $d_5$ $d_4$ \| $d_3$ $d_2$ $d_1$ $d_0$ | 43H |
| Description | This is a 2-cycle instruction. Data in the accumulator is logically ORed with an immediately-specified mask. | | |
| Operation | (A) ← (A) OR data | | |
| Example | ORL A,#'X' | 'OR' ACC CONTENTS WITH MASK 0101 1000 (ASCII VALUE OF 'X'). | |

## ORL A,Rr — Logical OR accumulator with register mask

| | | | |
|---|---|---|---|
| Encoding | 0 1 0 0 \| 1 r r r | | 48H-4FH |
| Description | Data in the accumulator is logically ORed with the mask contained in working register 'r'. | | |
| Operation | (A) ← (A) OR (Rr) | r = 0-7 | |
| Example | ORL A,R4 | 'OR' ACC CONTENTS WITH MASK IN REG 4. | |

## ORL A,@Rr — Logical OR accumulator with memory mask

| | | | |
|---|---|---|---|
| Encoding | 0 1 0 0 \| 0 0 0 r | | 40H-41H |
| Description | Data in the accumulator is logically ORed with the mask contained in the resident data memory location addressed by register Rr. | | |
| Operation | (A) ← (A) OR ((Rr)) | r = 0-1 | |
| Example | MOV RO,#3FH<br>ORL A,@R0 | MOVE '3F' HEX TO REG 0<br>'OR' ACC CONTENTS WITH MASK IN LOCATION 63. | |

## ORL Pp,#data — Logical OR port with immediate mask

| | | | |
|---|---|---|---|
| Encoding | 1 0 0 0 \| 1 0 p p | $d_7$ $d_6$ $d_5$ $d_4$ \| $d_3$ $d_2$ $d_1$ $d_0$ | 88H-8AH |
| Description | This is a 2-cycle instruction. Port 'p' data is logically ORed with an immediately-specified mask. | | |
| Operation | (Pp) ← (Pp) OR data | p = 0-2; P0 is not used by 8048 | |
| Example | ORL P1,#0FFH | 'OR' PORT 1 CONTENTS WITH MASK 'FF' HEX (SET PORT 1 TO ALL ONES) | |

## OUTL Pp,A — Output accumulator data to port

| | | | |
|---|---|---|---|
| Encoding | 0 0 1 1 \| 1 0 p p | | 38H-3AH |
| Description | This is a 2-cycle instruction. Data residing in the accumulator is transferred (written) to Port 'p' and latched. | | |
| Operation | (Pp) ← (A) | p = 0-2; P0 is not used by 8048 | |
| Example | MOV A,R7<br>OUTL P1,A<br>MOV A,R6<br>OUTL P2,A | MOV REG 7 CONTENTS TO ACC<br>OUTPUT ACC CONTENTS TO PORT 1<br>MOVE REG 6 CONTENTS TO ACC<br>OUTPUT ACC CONTENTS TO PORT 2 | |

## RET — Return without PSW restore

| | | |
|---|---|---|
| Encoding | 1 0 0 0 \| 0 0 1 1 | 83H |
| Description | This is a 2-cycle instruction. The stack pointer is decremented. The program counter is then restored from the stack. PSW bits are not restored. | |
| Operation | (SP) ← (SP)-1<br>(PC) ← ((SP)) | |

## RETR Return with PSW restore

Encoding

| 1 0 0 1 | 0 0 1 1 |
|---|---|

93H

Description

This is a 2-cycle instruction. The stack pointer is decremented. The program counter and PSW are then restored from the stack. Note that RETR should be used to return frominterrupt service routines, but should not be used within them as RETR signals the end of an interrupt routine.

Operation

| **8048** | **84XX, 84CXXX, 33XX** |
|---|---|
| (SP) ← (SP)-1 | (SP) ← (SP)-1 |
| (PC) ← ((SP)) | (PC) ← ((SP)) |
| $(PSW_{4-7})$ ← ((SP)) | $(PSW_{4,6,7})$ ← ((SP)) |

## RL A Rotate left without carry

Encoding

| 1 1 1 0 | 0 1 1 1 |
|---|---|

E7H

Description

The contents of the accumulator are rotated left one bit. Bit 7 is rotated into the bit 0 position.

Operation

(An+1) ← (An) n = 0-6
(A0) ← (A7)

Example

Assume accumulator contains 1011 0001

RL A NEW ACC CONTENTS ARE 0110 0011

## RLC A Rotate left through carry

Encoding

| 1 1 1 1 | 0 1 1 1 |
|---|---|

F7H

Description

The contents of the accumulator are rotated left one bit. Bit 7 replaces the carry bit; the carry bit is rotated into the bit 0 position.

Operation

(An+1) ← (An) n = 0-6
(A0) ← (C)
(C) ← (A7)

Example

Assume accumulator contains a 'signed' number; isolate sign without changing value.

| | |
|---|---|
| CLR C | CLEAR CARRY TO ZERO |
| RLC A | ROTATE ACC LEFT, SIGN BIT (7) IS PLACED IN CARRY |
| RR A | ROTATE ACC RIGHT - VALUE (BITS 0-6) IS RESTORED, CARRY UNCHANGED, BIT 7 IS ZERO. |

## RR A — Rotate right without carry

Encoding | 0 1 1 1 | 0 1 1 1 | 77H

Description: The contents of the accumulator are rotated right one bit. Bit 0 is rotated into the bit 7 position.

Operation:
(An) ← (An+1) n = 0-6
(A7) ← (A0)

Example: Assume accumulator contains 1011 0001.
RR A — NEW ACC CONTENTS ARE 1101 1000

## RRC A — Rotate right trough carry

Encoding | 0 1 1 0 | 0 1 1 1 | 67H

Description: The contents of the accumulator are rotated right one bit. Bit 0 replaces the carry bit; the carry bit is rotated into the bit 7 position.

Operation:
(An) ← (An+1) n = 0-6
(A7) ← (C)
(C) ← (A0)

Example: Assume carry is not set and accumulator contains 1011 0001.
RRC A — CARRY IS SET AND ACC CONTAINS 0101 1000

## SEL MB0 — Select memory bank 0

Encoding | 1 1 1 0 | 0 1 0 1 | E5H

Description: Only devices with more than 2 K bytes of program memory use this instruction. PC bit 11 (and PC bit 12, if present) is set to zero on the next JMP or CALL instruction. All references to program memory addresses fall within the range 0-2047.

Operation:

| **8048** | **84XX, 84CXXX, 33XX** |
|---|---|
| (MBFF) ← (0) | $(MBFF_{1,0}) \leftarrow (0,0)$ |

## SEL MB1 — Select memory bank 1

Encoding | 1 1 1 1 | 0 1 0 1 | F5H

Description

Only devices with more than 2 K bytes of program memory use this instruction. PC bit 11 is set to '1' (and PC bit 12, if present, is reset to '0') on the next JMP or CALL instruction. All references to program memory addresses fall within the range 2048-4095.

Operation

| **8048** | **84XX, 84CXXX, 33XX** |
|---|---|
| (MBFF) ← (1) | $(MBFF_{1,0})$ ← (0,1) |

## SEL RB0 — Select register bank 0

Encoding | 1 1 0 0 | 0 1 0 1 | C5H

Description

PSW bit 4 is reset to zero. References to working registers 0-7 address data memory locations 0-7. This is the recommended setting for normal program execution.

Operation

(RBS) ← 0

## SEL RB1 — Select register bank 1

Encoding | 1 1 0 1 | 0 1 0 1 | D5H

Description

PSW bit 4 is set to one. References to working registers 0-7 address data memory locations 24-31. This is the recommended setting for interrupt service routines, since locations 0-7 are left intact. The setting of PSW bit 4 in effect at the time of an interrupt is restored by the RETR instruction when the interrupt service routine is completed.

Operation

(RBS) ← 1

Example

Assume an external interrupt has occurred, control has passed to program memory location 3, and PSW bit 4 was zero before the interrupt.

```
LOC:3  JNI INIT          JUMP TO ROUTINE 'INIT' IF INTERRUPT INPUT IS
                         ZERO
INIT:  MOV R7,A          MOVE ACC CONTENTS TO LOCATION 7
       SEL RB1           SELECT REG BANK 1
       MOV R7,#0FAH      MOVE 'FA' HEX TO LOCATION 31
       •
       •
       •
       SEL RB0           SELECT REG BANK 0
       MOV A,R7          RESTORE ACC FROM LOCATION 7
       RETR              RETURN - RESTORE PC AND PSW
```

## STOP TCNT Stop timer/event counter

Encoding

| 0 1 1 0 | 0 1 0 1 |
|---|---|

65H

Description

This instruction is used to stop both time accumulation and event counting.

Example

Disable interrupt, but jump to interrupt routine after eight overflows and stop timer. Count overflows in register 7.

```
START:  DIS TCNTI       DISABLE TIMER INTERRUPT
        CLR A           CLEAR ACC TO ZERO
        MOV T,A         MOVE ZEROS TO TIMER
        MOV R7,A        MOVE ZEROS TO REG 7
        STRT T          START TIMER
MAIN:   JTF COUNT       JUMP TO ROUTINE 'COUNT' IF TF = 1 AND
                        CLEAR TIMER FLAG
        JMP MAIN        CLOSE LOOP
COUNT:  INC R7          INCREMENT COUNTER (REG 7)
        MOV A,R7        MOVE REG 7 CONTENTS TO ACC
        JB3 INT         JMP TO 'INT' ROUTINE IF ACC BIT 3 IS SET
                        (REG 7 = 8)
        JMP MAIN        OTHERWISE RETURN TO ROUTINE 'MAIN'
        •
        •
        •
INT:    STOP TCNT       STOP TIMER
        JMP 7H          JUMP TO LOCATION 7 (TIMER) INTERRUPT
                        ROUTINE
```

## STRT CNT Start event counter

Encoding

| 0 1 0 0 | 0 1 0 1 |
|---|---|

45H

Description

The Test 1 (T1) pin is configured as the event counter input and the counter is started. The event counter register is incremented with each LOW-to-HIGH (HIGH-to-LOW for 8048) transition on the T1 pin.

Example

Initialize and start event counter. Assume overflow is desired with first T1 transition.

```
EN TCNTI        ENABLE COUNTER INTERRUPT
MOV A,#0FFH     MOVE ALL ONES TO ACC
MOV T,A         MOVE ONES TO COUNTER
STRT CNT        ENABLE T1 AS COUNTER INPUT AND START
```

## STRT T — Start timer

Encoding | 0 1 0 1 | 0 1 0 1 | 55H

Description

Time accumulation is initiated in the timer register. The register is incremented every 32 machine cycles. The prescaler which counts the 32 cycles is cleared but the timer register is not. For the 84XX, 84CXXX and 33XX, the prescaler may be bypassed. The 8048 prescaler may not be bypassed.

Example

Initialize and start timer.

```
CLR A          CLEAR ACC TO ZEROS
MOV T,A        MOVE ZEROS TO TIMER
EN TCNTI       ENABLE TIMER INTERRUPT
STRT T         START TIMER
```

## SWAP A — Swap nibbles within accumulator

Encoding | 0 1 0 0 | 0 1 1 1 | 47H

Description

Bits 0-3 of the accumulator are swapped with bits 4-7 of the accumulator.

Operation

$(A_{4-7}) \leftrightarrow (A_{0-3})$

Example

Pack bits 0-3 of locations 50-51 into location 50.

```
MOV R0,#50     MOVE '50' DEC TO REG 0
MOV R1,#51     MOVE '51' DEC TO REG 1
XCHD A,@R0     EXCHANGE BITS 0-3 OF ACC AND LOCATION 50
SWAP A         SWAP BITS 0-3 AND 4-7 OF ACC
XCHD A,@R1     EXCHANGE BITS 0-3 OF ACC AND LOCATION 51
MOV @R0,A      MOVE CONTENTS OF ACC TO LOCATION 50
```

## XCH A,Rr — Exchange accumulator register contents

Encoding | 0 0 1 0 | 1 r r r | 28H-2FH

Description

The contents of the accumulator and the contents of working register 'r' are exchanged.

Operation

(A) ↔ (Rr) r = 0-7

Example

Move PSW contents to Reg 6 without losing accumulator contents.

```
XCH A,R6       EXCHANGE CONTENTS OF REG 6 AND ACC
MOV A,PSW      MOVE PSW CONTENTS TO ACC
XCH A,R6       EXCHANGE CONTENTS OF REG 6 AND ACC
                AGAIN
```

## XCH A,@Rr — Exchange accumulator and data memory contents

Encoding | 0 0 1 0 | 0 0 0 r | 20H-21H

Description: The contents of the accumulator and the contents of the resident data memory location addressed by register Rr are exchanged. Register Rr contents are unaffected.

Operation: (A) ↔ ((Rr)) r = 0-1

Example: Decrement contents of location 52

```
MOV R0,#52      MOVE 52 DEC TO ADDRESS REG 0
XCH A,@R0       EXCHANGE CONTENTS OF ACC AND
                LOCATION 52
DEC A           DECREMENT ACC CONTENTS
XCH A,@R0       EXCHANGE CONTENTS OF ACC AND
                LOCATION 52 AGAIN
```

## XCHD A,@Rr — Exchange accumulator and data memory 4-bit data

Encoding | 0 0 1 1 | 0 0 0 r | 30H-31H

Description: This instruction exchanges bits 0-3 of the accumulator with bits 0-3 of the data memory location addressed by register Rr. Bits 4-7 of the accumulator, bits 4-7 of the data memory location, and the contents of register Rr are unaffected.

Operation: $(A_{0-3}) \leftrightarrow ((Rr_{0-3}))$ r = 0-1

Example: Load bits 0-3 of location 32 into A0-3 and clear bits 0-3 of location 32.

```
MOV R0,#32      MOVE '32' DEC TO REG 0
CLR A           CLEAR ACC TO ZEROS
XCHD A,@R0      EXCHANGE BITS 0-3 OF ACC AND LOCATION 32
                (BITS 0-3 OF LOCATION 32 ARE ZEROED)
```

## XRL A,#data — Logical XOR accumulator with immediate mask

Encoding | 1 1 0 1 | 0 0 1 1 | $d_7$ $d_6$ $d_5$ $d_4$ | $d_3$ $d_2$ $d_1$ $d_0$ | D3H

Description: This is a 2-cycle instruction. Data in the accumulator is EXCLUSIVE ORed with an immediately-specified mask.

Operation: (A) ← (A) OR data

Example:

```
XRL A,#HEXTWO   'XOR' CONTENTS OF ACC WITH MASK EQUAL
                VALUE OF SYMBOL 'HEXTWO'
```

## XRL A,Rr — Logical X0R accumulator with register mask

Encoding | 1 1 0 1 | 1 r r r | D8H-DFH

Description: Data in the accumulator is EXCLUSIVE ORed with the mask contained in working register 'r'.

Operation: (A) ← (A) X0R (Rr) r = 0-7

Example:

```
XRL A,R4        'XOR' ACC CONTENTS WITH MASK IN REG 4
```

## XRL A,@Rr — Logical X0R accumulator with memory mask

Encoding | 1 1 0 1 | 0 0 0 r | D0H-D1H

Description: Data in the accumulator is EXCLUSIVE ORed with the mask contained in the data memory location addressed by register Rr.

Operation: (A) ← (A) X0R ((Rr)) r = 0-1

Example:

```
MOV R1,#20H     MOVE '20' HEX TO REG 1
XRL A,@R1       'XOR' ACC CONTENTS WITH MASK IN
                LOCATION 32
```

### ADDITIONAL 8048 INSTRUCTIONS

This section describes the 8048 instructions which are not included in the common section; it should be used in conjunction with the common section.

## ANL BUS,#data — Logical AND BUS with immediate mask

Encoding | 1 0 0 1 | 1 0 0 0 | $d_7$ $d_6$ $d_5$ $d_4$ | $d_3$ $d_2$ $d_1$ $d_0$ | 98H

Description: This is a 2-cycle instruction. Data on the BUS port is logical ANDed with an immediately-specified mask. This instruction assumes prior execution of an 'OUTL BUS, A' instruction.

Operation: (BUS) ← (BUS) AND data

Example: ANL BUS,#MASK — 'AND' BUS CONTENTS WITH MASK EQUAL VALUE OF SYMBOL 'MASK'

## ANLD Pp,A — Logical AND port 4-7 with accumulator mask

Encoding | 1 0 0 1 | 1 1 p p | $d_7$ $d_6$ $d_5$ $d_4$ | $d_3$ $d_2$ $d_1$ $d_0$ | 9CH-9FH

Description: This is a 2-cycle instruction. Data on 8243 port 'p' is logically ANDed with the digit mask in accumulator bits 0-3.

Operation: (Pp) ← (Pp) AND (A0-3) p = 4-7

**Note:** The mapping of port 'p' to opcode bits 0-1 is as follows:

| BIT 1 | BIT 0 | PORT |
|---|---|---|
| 0 | 0 | 4 |
| 0 | 1 | 5 |
| 1 | 0 | 6 |
| 1 | 1 | 7 |

Example: ANLD P4,A — 'AND' PORT 4 CONTENTS WITH ACC BITS 0-3

## CLR F0 — Clear flag 0

Encoding | 1 0 0 0 | 0 1 0 1 | 85H

Description: Flag 0 is cleared to zero.

Operation: (F0) ← 0

## CLR F1 — Clear flag 1

Encoding | 1 0 1 0 | 0 1 0 1 | A5H

Description: Flag 1 is cleared to zero.

Operation: (F1) ← 0

## CPL F0 — Complement flag 0

Encoding | 1 0 0 1 | 0 1 0 1 | 95H

Description: Flag 0 is complemented; a zero is changed to one and vice-versa.

Operation: (F0) ← NOT (F0)

## CPL F1 — Complement flag 1

Encoding | 1 0 1 1 | 0 1 0 1 | B5H

Description: Flag 1 is complemented; a zero is changed to one and vice-versa.

Operation: (F1) ← NOT (F1)

## ENT0 CLK — Enable clock output

Encoding | 0 1 1 1 | 0 1 0 1 | 75H

Description: The T0 test pin is enabled to act as the clock output. This function is disabled by a system reset.

Example:

```
ENT0 CLK        ENABLE T0 AS CLOCK OUTPUT
```

## IDLE — Select idle operation (80C49 only)

Encoding | 0 0 0 0 | 0 0 0 1 | 01H

Description: This 2-cycle instruction places the microcontroller in a low-power mode. The oscillator, internal timer, external interrupt and counter pins continue to function and register and RAM status is maintained. To terminate the Idle mode, a system reset must be performed or interrupts must be enabled and an interrupt signal generated.

## INS A,BUS — Strobed input of BUS data to accumulator

Encoding: 0 0 0 0 | 1 0 0 0 — 08H

Description: This is a 2-cycle instruction. Data present on the BUS port is transferred (read) to the accumulator when the RD signal goes LOW. This instruction may be used in internal program memory only.

Operation: (A) ← (BUS)

Example: INS A,BUS — INPUT BUS CONTENTS TO ACC

## JF0 address — Jump if flag 0 is set

Encoding: 1 0 1 1 | 0 1 1 0 | $a_7$ $a_6$ $a_5$ $a_4$ | $a_3$ $a_2$ $a_1$ $a_0$ — B6H

Description: This is a 2-cycle instruction. Control passes to the specified address if flag 0 is set to one.

Operation: ($PC_{0-7}$) ← addr — If F0 = 1
(PC) ← (PC) + 2 — If F0 = 0

Example: JF0 TOTAL — JUMP TO 'TOTAL' ROUTINE IF F0 = 1

## JF1 address — Jump if flag 1 is set

Encoding: 0 1 1 1 | 0 1 1 0 | $a_7$ $a_6$ $a_5$ $a_4$ | $a_3$ $a_2$ $a_1$ $a_0$ — 76H

Description: This is a 2-cycle instruction. Control passes to the specified address if flag 1 is set to one.

Operation: ($PC_{0-7}$) ← addr — If F1 = 1
(PC) ← (PC) + 2 — If F1 = 0

Example: JF1 FILBLUF — JUMP TO 'FILBLUF' ROUTINE IF F1 = 1

## JNI address — Jump if interrupt input is LOW

Encoding

| 1 0 0 0 | 0 1 1 0 | $a_7$ $a_6$ $a_5$ $a_4$ | $a_3$ $a_2$ $a_1$ $a_0$ |
|---|---|---|---|

86H

Description

This is a 2-cycle instruction. Control passes to the specified address if the interrupt input is LOW (= 0), that is, an external interrupt has been signalled. (This signal initiates an interrupt service sequence if the external interrupt is enabled).

Operation

| | |
|---|---|
| $(PC_{0-7}) \leftarrow$ addr | If I = 0 |
| $(PC) \leftarrow (PC) + 2$ | If I = 1 |

Example

| | |
|---|---|
| JNI EXTINT | JUMP TO 'EXTINT' ROUTINE IF I = 0 |

## MOVD A,Pp — Move port 4-7 data to accumulator

Encoding

| 0 0 0 0 | 1 1 p p |
|---|---|

0CH-0FH

Description

This is a 2-cycle instruction. Data on 8243 port 'p' is moved (read) to accumulator bits 0-3. Accumulator bits 4-7 are reset to zero.

Operation

| | |
|---|---|
| $(A_{0-3}) \leftarrow (Pp)$ | p = 4-7 |
| $(A_{4-7}) \leftarrow 0$ | |

Note: Bits 0-1 of the opcode are used to represent ports 4-7. If coding in binary rather than assembly language, the mapping of as follows:

| Bit 1 | Bit 0 | Port |
|---|---|---|
| 0 | 0 | 4 |
| 0 | 1 | 5 |
| 1 | 0 | 6 |
| 1 | 1 | 7 |

Example

| | |
|---|---|
| MOVD A,P5 | MOVE PORT 5 DATA TO ACC BITS 0-3, ZERO ACC BITS 4-7 |

## MOVD Pp,A — Move accumulator data to port 4-7

Encoding | 0 0 1 1 | 1 1 p p | 3CH-3FH

Description: This is a 2-cycle instruction. Data in accumulator bits 0-3 is moved (written) to 8243 port 'p'. Accumulator bits 4-7 are unaffected. (See note above regarding port mapping).

Operation: $(Pp) \leftarrow (A_{0-3})$ p = 4-7

Example: Move data in accumulator to ports 4 and 5.

```
MOVD P4,A      MOVE ACC BITS 0-3 TO PORT 4
SWAP A         EXCHANGE ACC BITS 0-3 AND 4-7
MOVD P5,A      MOVE ACC BITS 0-3 TO PORT 5
```

## MOVP3 A,@A — Move page 3 data to accumulator

Encoding | 1 1 1 0 | 0 0 1 1 | E3H

Description: This is a 2-cycle instruction. The contents of the program memory location (within page 3) addressed by the accumulator are moved to the accumulator. The program counter is restored following this operation.

Operation:
$(PC_{0-7}) \leftarrow A$
$(PC_{11,10,9,8}) \leftarrow 0011$
$(A) \leftarrow ((PC))$

Example: Look up ASCII equivalent of hexadecimal code in table contained at the beginning of page 3. Note that ASCII characters are designated by a 7-bit code; the eighth bit is always reset.

```
MOV A,#0B8H    MOVE 'B8' HEX TO ACC (1011 1000)
ANL A,#7FH     LOGICAL AND ACC TO MASK BIT 7
MOVP3 A,@A     MOVE CONTENTS OF LOCATION '38' HEX IN
               PAGE 3 TO ACC (ASCII '8')
```

## MOVX A,@Rr — Move external data memory contents to accumulator

Encoding | 1 0 0 0 | 0 0 0 r | 80H-81H

Description: This is a 2-cycle instruction. The contents of the external data memory location addressed by register 'r' are moved to the accumulator. Register 'r' contents are unaffected.

Operation: $(A) \leftarrow ((Rr))$ r = 0-1

Example:
```
Assume R1 contains 0111 0110
MOVX A,@R1     MOVE CONTENTS OF LOCATION 76H TO ACC
```

## MOVX @Rr,A — Move accumulator contents to external data memory

Encoding: 1 0 0 1 | 0 0 0 r — 90H-91H

Description: This is a 2-cycle instruction. The contents of the accumulator are moved to the external data memory location addressed by register 'r'. Register 'r' contents are unaffected.

Operation: ((Rr)) ← (A) — r = 0-1

Example:
Assume R0 contains 1100 0111
MOVX @R0,A — MOVE CONTENTS OF ACC TO LOCATION C7H IN EXPANDED DATA MEMORY

## ORL BUS,#data — Logical OR BUS with immediate mask

Encoding: 1 0 0 0 | 1 0 0 0 | $d_7$ $d_6$ $d_5$ $d_4$ | $d_3$ $d_2$ $d_1$ $d_0$ — 88H

Description: This is a 2-cycle instruction. Data on the BUS port is logically ORed with an immediately-specified mask. This instruction assumes prior execution of an 'OUTL BUS,A' instruction.

Operation: (BUS) ← (BUS) OR data

Example: ORL BUS,#HEXMSK — 'OR' BUS CONTENTS WITH MASK EQUAL VALUE OF SYMBOL 'HEXMSK'

## ORLD Pp,A — Logical OR port 4-7 with immediate mask

Encoding: 1 0 0 0 | 1 1 p p — 8CH-8FH

Description: This is a 2-cycle instruction. Data on 8243 port 'p' is logically ORed with the mask contained in accumulator bits 0-3.

Operation: (Pp) ← (Pp) AND ($A_{0-3}$) — p = 4-7

Example: ORLD P6,A — 'OR' PORT 6 CONTENTS WITH ACC BITS 0-3

## OUTL BUS,A — Output accumulator data to bus

Encoding

| 0 0 0 0 | 0 0 1 0 |
|---|---|

02H

Description

This is a 2-cycle instruction. Data residing in the accumulator is transferred (written) to the BUS port and latched. The latched data remains valid until altered by another OUTL instruction. Any other instruction requiring use of the BUS port (except INS) overwrites the contents of the BUS latch. This includes expanded memory operations (such as the MOVX instruction). Logical operations on BUS data (AND, OR) assume the OUTL BUS,A instruction has been previously executed.

Operation

(BUS) ← (A)

Example

```
OUTL BUS,A          OUTPUT ACC CONTENTS TO BUS
```

### ADDITIONAL 84XX INSTRUCTIONS

This section describes the 84XX instructions which are not included in the common section; it should be used in conjunction with the common section.

## DEC @Rr — Decrement data memory location

Encoding | 1 1 0 0 \| 0 0 0 r | C0H-C1H

Description | The contents of the data memory location addressed by register Rr are decremented by one.

Operation | ((Rr)) ← ((Rr)) –1 | r = 0-1

Example

```
MOV R1,#4FH     MOVE '4F' HEX TO REG 1
DEC @R1         DECREMENT LOCATION 4F
```

## DIS SI — Disable serial input/output interrupt

Encoding | 1 0 0 1 \| 0 1 0 1 | 95H

Description | Serial input/output interrupts are disabled. An interrupt request from the SI0 has no effect. If the microcontroller enters the IDLE mode when the SI0 interrupt is disabled, it can't be restarted by this interrupt.

Operation | (SIFF) ← 0

## DJNZ @Rr,addr — Decrement data memory location and test

Encoding

| 1 1 1 0 | 0 0 0 r | $a_7$ $a_6$ $a_5$ $a_4$ | $a_3$ $a_2$ $a_1$ $a_0$ | EOH-E1H |
|---|---|---|---|---|

Description

This is a 2-cycle instruction. The memory location addressed by register Rr is decremented, then tested for zero. If the memory location contains all zeros, program control passes to the next instruction. If the register contents are not zero, control jumps to the specified 'address'.

The address in this case must evaluate to 8 bits (the jump must be to a location within the current 256-location page).

Operation

((Rr)) ← ((Rr)) –1 r = 0-1
If ((Rr)) ≠ 0
then ($PC_{0-7}$) ← addr
else (PC) ← (PC) + 2

Example

Increment values in data memory locations 81-86.

```
        MOV R0,#32      MOVE '32' DEC TO ADDRESS REG 0
        MOV @R0,#6      MOVE '6' DEC TO COUNTER REG 32
        MOV R1,#81      MOVE '81' DEC TO ADDRESS REG 1
INCRT   INC @R1         INCREMENT CONTENTS OF MEMORY
                        LOCATION ADDRESSED BY REG 1
        INC R1          INCREMENT ADDRESS IN REG 1
        DJNZ @R0,INCRT  DECREMENT LOC. 32 - JUMP TO 'INCRT' IF
                        LOCATION 32 NONZERO
        NEXT ---        'NEXT' ROUTINE EXECUTED IF LOCATION 32
                        IS ZERO
```

## EN SI — Enable serial input/output interrupt

Encoding

| 1 0 0 0 | 0 1 0 1 | 85H |
|---|---|---|

Description

Serial input/output interrupts are enabled. An interrupt request from the serial I/O initiates the interrupt sequence, in normal mode or in Idle mode.

Operation

(SIFF) ← 1

## JNTF address — Jump if timer flag is not set

Encoding: 0 0 0 0 | 0 1 1 0 — $a_7$ $a_6$ $a_5$ $a_4$ | $a_3$ $a_2$ $a_1$ $a_0$ — 06H

Description: This is a 2-cycle instruction. Control passes to the specified address if the timer flag is not set, that is, if the timer/counter has not overflowed. Otherwise, program control passes to the next instruction. Testing the timer flag resets it to zero.

Operation:
$(PC_{0-7}) \leftarrow$ addr — If TF = 0
$(PC) \leftarrow (PC) + 2$ — If TF = 1

Example: JNTF NOTIM — JUMP TO 'NOTIM' ROUTINE IF THE TIMER HAS NOT OVERFLOWED

**Note:** if this instruction begins in location 255 of a page, the target address is located in the following page.

## MOV A,Sn — Move serial I/O register contents to accumulator

Encoding: 0 0 0 0 | 1 1 0 n — 0CH, 0DH

Description: This is a 2-cycle instruction. The contents of serial I/O register 'n' are moved to the accumulator.

Operation: $(A) \leftarrow (Sn)$ — n = 0-1

Example: MOV A,S0 — MOVE CONTENTS OF SERIAL I/O DATA REGISTER TO ACC

## MOV Sn,A — Move accumulator contents to serial I/O register

Encoding: 0 0 1 1 | 1 1 n n — 3CH-3EH

Description: This is a 2-cycle instruction. The contents of the accumulator are moved to serial I/O register 'n'.

Operation: $(Sn) \leftarrow A$ — n = 0-2

Example: MOV S0,A — MOVE CONTENTS OF ACC TO SIO DATA REGISTER

## MOV Sn,#data — Move immediate data to serial I/O register

Encoding: 1 0 0 1 | 1 1 n n | $d_7$ $d_6$ $d_5$ $d_4$ | $d_3$ $d_2$ $d_1$ $d_0$ — 9CH-9EH

Description: This is a 2-cycle instruction. The byte specified by 'data' is moved to the serial I/O register 'n'.

Operation: (Sn) ← data — n = 0-2

Example: MOV S2,#28H — LOAD SIO CLOCK REGISTER

## SEL MB2 — Select memory bank 2

Encoding: 1 0 1 0 | 0 1 0 1 — A5H

Description: Only devices with more than 4 K bytes of program memory use this instruction. PC bit 11 is set to zero and PC bit 12 is set to 1 on the next JMP or CALL instruction. All references to program memory addresses fall within the range 4092-6143.

Operation: (MBFF1,0) ← (1,0)

## SEL MB3 — Select memory bank 3

Encoding: 1 0 1 1 | 0 1 0 1 — B5H

Description: Only devices with more than 6 K bytes of program memory use this instruction. PC bits 11 and 12 are set to 1 on the next JMP or CALL instruction. All references to program memory addresses fall within the range 6144-8191.

Operation: (MBFF1,0) ← 1,1

### ADDITIONAL 84CXXX INSTRUCTIONS

This section describes the 84CXXX instructions which are not included in the common section; it should be used in conjunction with the common section.

## ANL Dx,A — Logical AND derivative register with accumulator

| | | | |
|---|---|---|---|
| Encoding | 1 0 0 0 \| 1 1 1 0 | $a_7$ $a_6$ $a_5$ $a_4$ \| $a_3$ $a_2$ $a_1$ $a_0$ | 8EH |
| Description | This is a 2-byte, 2-cycle instruction. Data in the derivative register addressed by the second byte is logically ANDed with the accumulator contents. | | |
| Operation | (Dx) ← (Dx) AND (A) | | |
| Example | ANL D7,A | 'AND' CONTENTS OF DERIVATIVE REGISTER D7 WITH ACC | |

## DEC @Rr — Decrement data memory location

| | | | |
|---|---|---|---|
| Encoding | 1 1 0 0 \| 0 0 0 r | | C0H-C1H |
| Description | The contents of the data memory location addressed by working register 'r' are decremented by one. | | |
| Operation | ((Rr)) ←((Rr)) –1 | r = 0-1 | |
| Example | MOV R1,#4FH<br>DEC @R1 | MOVE '4F' HEX TO REG 1<br>DECREMENT LOCATION 4F | |

## DIS SI — Disable serial input/output interrupt

| | | |
|---|---|---|
| Encoding | 1 0 0 1 \| 0 1 0 1 | 95H |
| Description | Serial input/output interrupts are disabled. An interrupt request from the SI0 has no effect. If the microcontroller enters the IDLE mode when the SI0 interrupt is disabled, it can't be restarted by this interrupt. | |
| Operation | (SIFF) ← 0 | |

## DJNZ @Rr,addr — Decrement data memory location and test

Encoding

| 1 1 1 0 | 0 0 0 r | $a_7$ $a_6$ $a_5$ $a_4$ | $a_3$ $a_2$ $a_1$ $a_0$ | E0H-E1H |
|---|---|---|---|---|

Description

This is a 2-cycle instruction. The memory location addressed by the working register 'r' is decremented, then tested for zero. If the memory location contains all zeros, program control passes to the next instruction. If the register contents are not zero, control jumps to the specified 'address'.

The address in this case must evaluate to 8 bits (the jump must be to a location within the current 256-location page).

Operation

((Rr)) ← ((Rr)) –1 r = 0-1
If ((Rr)) ≠ 0
then ($PC_{0-7}$) ← addr
else (PC) ← (PC) + 2

Example

Increment values in data memory locations 81-86.

```
        MOV R0,#32          MOVE '32' DEC TO ADDRESS REG 0
        MOV @R0,#6          MOVE '6' DEC TO COUNTER REG 32
        MOV R1,#81          MOVE '81' DEC TO ADDRESS REG 1
INCRT   INC @R1             INCREMENT CONTENTS OF MEMORY
                            LOCATION ADDRESSED BY REG 1
        INC R1              INCREMENT ADDRESS IN REG 1
        DJNZ @R0,INCRT      DECREMENT LOC. 32 - JUMP TO 'INCRT' IF
                            LOCATION 32 NONZERO
        NEXT ---            'NEXT' ROUTINE EXECUTED IF LOCATION 32 IS
                            ZERO
```

## EN SI — Enable serial input/output interrupt

Encoding

| 1 0 0 0 | 0 1 0 1 | 85H |
|---|---|---|

Description

Serial input/output interrupts are enabled. An interrupt request from the serial I/O initiates the interrupt sequence, in normal mode or in Idle mode.

Operation

(SIFF) ← 1

## IDLE — Select IDLE operation

Encoding | 0 0 0 0 | 0 0 0 1 | 01H

Description

This instruction places the microcontroller in the Idle mode. The oscillator, timer/counter and serial I/O continue to function. The Idle mode is terminated:
- when an enabled interrupt is activated, or
- by a RESET.

An active signal on the RESET pin always restarts the microcontroller and a normal RESET sequence is executed.

An active signal from an interrupt source (if enabled) causes the execution of the normal interrupt routine since normal interrupt scanning is still being carried out. A HIGH-to-LOW transition on the external interrupt pin reactivates the microcontroller.

## JNTF address — Jump if timer flag is not set

Encoding | 0 0 0 0 | 0 1 1 0 | $a_7$ $a_6$ $a_5$ $a_4$ | $a_3$ $a_2$ $a_1$ $a_0$ | 06H

Description

This is a 2-cycle instruction. Control passes to the specified address if the timer flag is not set, that is, if the timer/counter has not overflowed. Otherwise, program control passes to the next instruction. Testing the timer flag resets it to zero.

| | | |
|---|---|---|
| Operation | $(PC_{0-7}) \leftarrow$ addr | If TF = 0 |
| | (PC) $\leftarrow$ (PC) + 2 | If TF = 1 |
| Example | JNTF NOTIM | JUMP TO 'NOTIM' ROUTINE IF THE TIMER HAS NOT OVERFLOWED |

**Note:** If this instruction begins in location 255 of a page, the target address is located in the following page.

## MOV A,Dx — Move derivative register contents to accumulator

Encoding | 1 0 0 0 | 1 1 0 0 | $a_7$ $a_6$ $a_5$ $a_4$ | $a_3$ $a_2$ $a_1$ $a_0$ | 8CH

Description

This is a 2-cycle instruction. The contents of the derivative register (x = 0 to 255) addressed by the second byte are moved to the accumulator.

| | | |
|---|---|---|
| Operation | (A) $\leftarrow$ (Dx) | x = 0 to 255 |
| Example | MOV A,D5 | MOVE THE CONTENTS OF DERIVATIVE REGISTER 5 TO THE ACCUMULATOR |

## MOV A,Sn — Move serial I/O register contents to accumulator

Encoding | 0 0 0 0 | 1 1 0 n | 0CH, 0DH

Description: This is a 2-cycle instruction. The contents of serial I/O register 'n' are moved to the accumulator.

Operation: (A) ← (Sn) n = 0-1

Example: MOV A,S0 — MOVE CONTENTS OF SERIAL I/O DATA REGISTER TO ACC

## MOV Dx,A — Move accumulator contents to derivative register

Encoding | 1 0 0 0 | 1 1 0 1 | $a_7$ $a_6$ $a_5$ $a_4$ | $a_3$ $a_2$ $a_1$ $a_0$ | 8DH

Description: This is a 2-cycle instruction. The contents of the accumulator are moved to the derivative register addressed by the second byte.

Operation: (Dx) ← (A) x = 0 to 255

Example: MOV D2,A — MOVE CONTENTS OF ACC TO DERIVATIVE REGISTER 2

## MOV Sn,A — Move accumulator contents to serial I/O register

Encoding | 0 0 1 1 | 1 1 n n | 3CH-3EH

Description: This is a 2-cycle instruction. The contents of the accumulator are moved to serial I/O register 'n'.

Operation: (Sn) ← A n = 0-2

Example: MOV S0,A — MOVE CONTENTS OF ACC TO SIO DATA REGISTER

## MOV Sn,#data — Move immediate data to serial I/O register

| | | | |
|---|---|---|---|
| Encoding | 1 0 0 1 \| 1 1 n n | $d_7$ $d_6$ $d_5$ $d_4$ \| $d_3$ $d_2$ $d_1$ $d_0$ | 9CH-9EH |
| Description | This is a 2-cycle instruction. The byte specified by 'data' is moved to the serial I/O register 'n'. | | |
| Operation | (Sn) ← data | n = 0-2 | |
| Example | MOV S2,#28H | LOAD SIO CLOCK REGISTER | |

## ORL Dx,A — Logical OR derivative register contents with accumulator

| | | | |
|---|---|---|---|
| Encoding | 1 0 0 0 \| 1 1 1 1 | $a_7$ $a_6$ $a_5$ $a_4$ \| $a_3$ $a_2$ $a_1$ $a_0$ | 8FH |
| Description | This is a 2-cycle instruction. The contents of the accumulator are logically ORed with the contents of the derivative register addressed by the second byte, and the result placed in the same derivative register. | | |
| Operation | (Dx) ← (Dx) OR (A) | x = 0-255 | |

## SEL MB2 — Select memory bank 2

| | | |
|---|---|---|
| Encoding | 1 0 1 0 \| 0 1 0 1 | A5H |
| Description | Only devices with more than 4 K bytes of program memory use this instruction. PC bit 11 is set to zero and PC bit 12 is set to 1 on the next JMP or CALL instruction. All references to program memory addresses fall within the range 4092-6143. | |
| Operation | (MBFF1,0) ← (1,0) | |

## SEL MB3 — Select memory bank 3

| | | |
|---|---|---|
| Encoding | 1 0 1 1 \| 0 1 0 1 | B5H |
| Description | Only devices with more than 6 K bytes of program memory use this instruction. PC bits 11 and 12 are set to 1 on the next JMP or CALL instruction. All references to program memory addresses fall within the range 6144-8191. | |
| Operation | (MBFF1,0) ← 1,1 | |

## STOP Stop processor

Encoding

| 0 0 1 0 | 0 0 1 0 |
|---|---|

22H

Description

This instruction places the microcontroller in the Stop mode. The oscillator is switched off; the internal status of the CPU, RAM contents and the state of I/O ports are not affected. The Stop mode may be terminated either by an active signal at the external interrupt or by an external RESET signal. When the Stop mode is terminated, an internal delay is provided to ensure that, before restarting, all internal clocks are working correctly.

If the Stop mode is terminated with a RESET, a normal RESET sequence is executed.

If the Stop mode is terminated by pulling the external interrupt pin LOW, an interrupt sequence is only executed if the external interrupt has been enabled and the microcontroller resumes the normal program sequence after returning from the interrupt routine, as in the normal mode. If the external interrupt is not enabled, the microcontroller continues the normal program sequence, executing the instruction following the Stop instruction in the main program.

**Note:** The microcontroller is restarted by a LOW level applied at the $\overline{\text{INT}}$/TO pin, and not by a HIGH-to-LOW transition (the normal interrupt mechanism). If the $\overline{\text{INT}}$/TO pin is LOW during the STOP instruction then the STOP instruction will be ignored.

### ADDITIONAL 33XX INSTRUCTIONS

This section describes the 33XX instructions which are not included in the common section; it should be used in conjunction with the common section.

## ANL Dx,A — Logical AND derivative register with accumulator

Encoding: 1 0 0 0 | 1 1 1 0 — $a_7$ $a_6$ $a_5$ $a_4$ | $a_3$ $a_2$ $a_1$ $a_0$ — 8EH

Description: This is a 2-byte, 2-cycle instruction. Data in the derivative register addressed by the second byte is logically ANDed with the accumulator contents.

Operation: (Dx) ← (Dx) AND (A)

Example:

| | |
|---|---|
| ANL D7,A | 'AND' CONTENTS OF DERIVATIVE REGISTER D7 WITH ACC |

## DEC @Rr — Decrement data memory location

Encoding: 1 1 0 0 | 0 0 0 r — C0H-C1H

Description: The contents of the data memory location addressed by working register 'r' are decremented by one.

Operation: ((Rr)) ←((Rr)) –1 r = 0-1

Example:

| | |
|---|---|
| MOV R1,#4FH | MOVE '4F' HEX TO REG 1 |
| DEC @R1 | DECREMENT LOCATION 4F |

## DIS SI — Disable serial input/output interrupt

Encoding: 1 0 0 1 | 0 1 0 1 — 95H

Description: Serial input/output interrupts are disabled. An interrupt request from the SI0 has no effect. If the microcontroller enters the IDLE mode when the SI0 interrupt is disabled, it can't be restarted by this interrupt.

Operation: (SIFF) ← 0

## DJNZ @Rr,addr — Decrement data memory location and test

Encoding

| 1 1 1 0 | 0 0 0 r | $a_7$ $a_6$ $a_5$ $a_4$ | $a_3$ $a_2$ $a_1$ $a_0$ | E0H-E1H |
|---|---|---|---|---|

Description

This is a 2-cycle instruction. The memory location addressed by the working register 'r' is decremented, then tested for zero. If the memory location contains all zeros, program control passes to the next instruction. If the register contents are not zero, control jumps to the specified 'address'.

The address in this case must evaluate to 8 bits (the jump must be to a location within the current 256-location page).

Operation

((Rr)) ← ((Rr)) −1 r = 0-1
If ((Rr)) ≠ 0
then ($PC_{0-7}$) ← addr
else (PC) ← (PC) + 2

Example

Increment values in data memory locations 81-86.

```
        MOV R0,#32      MOVE '32' DEC TO ADDRESS REG 0
        MOV @R0,#6      MOVE '6' DEC TO COUNTER REG 32
        MOV R1,#81      MOVE '81' DEC TO ADDRESS REG 1
INCRT   INC @R1         INCREMENT CONTENTS OF MEMORY
                        LOCATION ADDRESSED BY REG 1
        INC R1          INCREMENT ADDRESS IN REG 1
        DJNZ @R0,INCRT  DECREMENT LOC. 32 - JUMP TO 'INCRT' IF
                        LOCATION 32 NONZERO
        NEXT ---        'NEXT' ROUTINE EXECUTED IF LOCATION 32 IS
                        ZERO
```

## EN SI — Enable serial input/output interrupt

Encoding

| 1 0 0 0 | 0 1 0 1 | 85H |
|---|---|---|

Description

Serial input/output interrupts are enabled. An interrupt request from the serial I/O initiates the interrupt sequence, in normal mode or in Idle mode.

Operation

(SIFF) ← 1

## IDLE — Select IDLE operation

Encoding | 0 0 0 0 | 0 0 0 1 | 01H

Description

This instruction places the microcontroller in the Idle mode. The oscillator, timer/counter and serial I/O continue to function. The Idle mode is terminated:
- when an enabled interrupt is activated, or
- by a RESET.

An active signal on the RESET pin always restarts the microcontroller and a normal RESET sequence is executed.

An active signal from an interrupt source (if enabled) causes the execution of the normal interrupt routine since normal interrupt scanning is still being carried out. A LOW-to-HIGH transition on the external interrupt pin (CE pin) reactivates the microcontroller.

## JNTF address — Jump if timer flag is not set

Encoding | 0 0 0 0 | 0 1 1 0 | $a_7$ $a_6$ $a_5$ $a_4$ | $a_3$ $a_2$ $a_1$ $a_0$ | 06H

Description

This is a 2-cycle instruction. Control passes to the specified address if the timer flag is not set, that is, if the timer/counter has not overflowed. Otherwise, program control passes to the next instruction. Testing the timer flag resets it to zero.

Operation

| | |
|---|---|
| $(PC_{0-7}) \leftarrow$ addr | If TF = 0 |
| $(PC) \leftarrow (PC) + 2$ | If TF = 1 |

Example

| | |
|---|---|
| JNTF NOTIM | JUMP TO 'NOTIM' ROUTINE IF THE TIMER HAS NOT OVERFLOWED |

**Note:** If this instruction begins in location 255 of a page, the target address is located in the following page.

## MOV A,Dx — Move derivative register contents to accumulator

Encoding | 1 0 0 0 | 1 1 0 0 | $a_7$ $a_6$ $a_5$ $a_4$ | $a_3$ $a_2$ $a_1$ $a_0$ | 8CH

Description

This is a 2-cycle instruction. The contents of the derivative register (x = 0 to 255) addressed by the second byte are moved to the accumulator.

Operation

| | |
|---|---|
| $(A) \leftarrow (Dx)$ | x = 0 to 225 |

Example

| | |
|---|---|
| MOV A,D5 | MOVE THE CONTENTS OF DERIVATIVE REGISTER 5 TO THE ACCUMULATOR |

## MOV A,Sn — Move serial I/O register contents to accumulator

| | | |
|---|---|---|
| Encoding | 0 0 0 0 \| 1 1 0 n | 0CH, 0DH |
| Description | This is a 2-cycle instruction. The contents of serial I/O register 'n' are moved to the accumulator. | |
| Operation | (A) ← (Sn) | n = 0-1 |
| Example | MOV A,S0 | MOVE CONTENTS OF SERIAL I/O DATA REGISTER TO ACC |

## MOV Dx,A — Move accumulator contents to derivative register

| | | | |
|---|---|---|---|
| Encoding | 1 0 0 0 \| 1 1 0 1 | $a_7$ $a_6$ $a_5$ $a_4$ \| $a_3$ $a_2$ $a_1$ $a_0$ | 8DH |
| Description | This is a 2-cycle instruction. The contents of the accumulator are moved to the derivative register addressed by the second byte. | | |
| Operation | (Dx) ← (A) | x = 0 to 255 | |
| Example | MOV D2,A | MOVE CONTENTS OF ACC TO DERIVATIVE REGISTER 2 | |

## MOV Sn,A — Move accumulator contents to serial I/O register

| | | |
|---|---|---|
| Encoding | 0 0 1 1 \| 1 1 n n | 3CH-3EH |
| Description | This is a 2-cycle instruction. The contents of the accumulator are moved to serial I/O register 'n'. | |
| Operation | (Sn) ← A | n = 0-2 |
| Example | MOV S0,A | MOVE CONTENTS OF ACC TO SIO DATA REGISTER |

## MOV Sn,#data — Move immediate data to serial I/O register

| | | | |
|---|---|---|---|
| Encoding | 1 0 0 1 \| 1 1 n n | $d_7$ $d_6$ $d_5$ $d_4$ \| $d_3$ $d_2$ $d_1$ $d_0$ | 9CH-9EH |
| Description | This is a 2-cycle instruction. The byte specified by 'data' is moved to the serial I/O register 'n'. | | |
| Operation | (Sn) ← data | n = 0-2 | |
| Example | MOV S2,#28H | LOAD SIO CLOCK REGISTER | |

## ORL Dx,A — Logical OR derivative register contents with accumulator

| | | | |
|---|---|---|---|
| Encoding | 1 0 0 0 \| 1 1 1 1 | $a_7$ $a_6$ $a_5$ $a_4$ \| $a_3$ $a_2$ $a_1$ $a_0$ | 8FH |
| Description | This is a 2-cycle instruction. The contents of the accumulator are logically ORed with the contents of the derivative register addressed by the second byte, and the result placed in the same derivative register. | | |
| Operation | (Dx) ← (Dx) OR (A) | x = 0-255 | |

## SEL MB2 — Select memory bank 2

| | | |
|---|---|---|
| Encoding | 1 0 1 0 \| 0 1 0 1 | A5H |
| Description | Only devices with more than 4 K bytes of program memory use this instruction. PC bit 11 is set to zero and PC bit 12 is set to 1 on the next JMP or CALL instruction. All references to program memory addresses fall within the range 4092-6143. | |
| Operation | (MBFF1,0) ← (1,0) | |

## SEL MB3 — Select memory bank 3

| | | |
|---|---|---|
| Encoding | 1 0 1 1 \| 0 1 0 1 | B5H |
| Description | Only devices with more than 6 K bytes of program memory use this instruction. PC bits 11 and 12 are set to 1 on the next JMP or CALL instruction. All references to program memory addresses fall within the range 6144-8191. | |
| Operation | (MBFF1,0) ← 1,1 | |

## STOP Stop processor

Encoding

| 0 0 1 0 | 0 0 1 0 |
|---|---|

22H

Description

This instruction places the microcontroller in the Stop mode. The oscillator is switched off; the internal status of the CPU, RAM contents and the state of I/O ports are not affected. The Stop mode may be terminated either by an active signal at the external interrupt or by an external RESET signal. When the Stop mode is terminated, an internal delay is provided to ensure that before restarting, all internal clocks are working correctly.

If the Stop mode is terminated via a RESET, a normal RESET sequence is executed.

If the Stop mode is terminated by pulling the CE pin HIGH, an interrupt sequence is only executed if the external interrupt has been enabled and the microcontroller resumes the normal program sequence after returning from the interrupt routine, as in the normal mode. If the external interrupt is not enabled, the microcontroller continues the normal program sequence, executing the instruction following the Stop instruction in the main program.

**Note:** The microcontroller is restarted by a HIGH level applied at the CE/$\overline{TO}$ pin, and not by a LOW-to-HIGH transition (the normal interrupt mechanism). If the CE/$\overline{TO}$ pin is HIGH during the STOP instruction then the STOP instruction will be ignored.

# SECTION 2 84XX DERIVATIVES

# SINGLE-CHIP 8-BIT MICROCONTROLLER

## DESCRIPTION

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

- MAB8401 – 128 bytes RAM, external program memory, with 8-bit LED-driver (10mA), emulation of MAB/F8422/42* possible

- MAB/MAF8421 – 2K bytes ROM/64 bytes RAM plus 8-bit LED-driver
- MAB/MAF8441 – 4K bytes ROM/128 bytes RAM plus 8-bit LED-driver
- MAB/MAF8461 – 6K bytes ROM/128 bytes RAM plus 8-bit LED-driver

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

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

For detailed information see the 84XX family specification.

* See data sheet on MAB/F8422/42.

### Features

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

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

## PACKAGE OUTLINES

MAB8401B: 28-lead 'Piggy-back' package (with up to 28-pin EPROM on top).
MAB8401WP: 68-lead plastic leaded chip-carrier (PLCC) (SOT188).
MAB/MAF8421/41/61P: 28-lead DIL; plastic with internal heat spreader (SOT117).
MAF84A21/41/61P: 28-lead DIL; plastic with internal heat spreader (SOT117).
MAB8421/41/61T: 28-lead mini-pack; plastic (SO28; SOT136A).

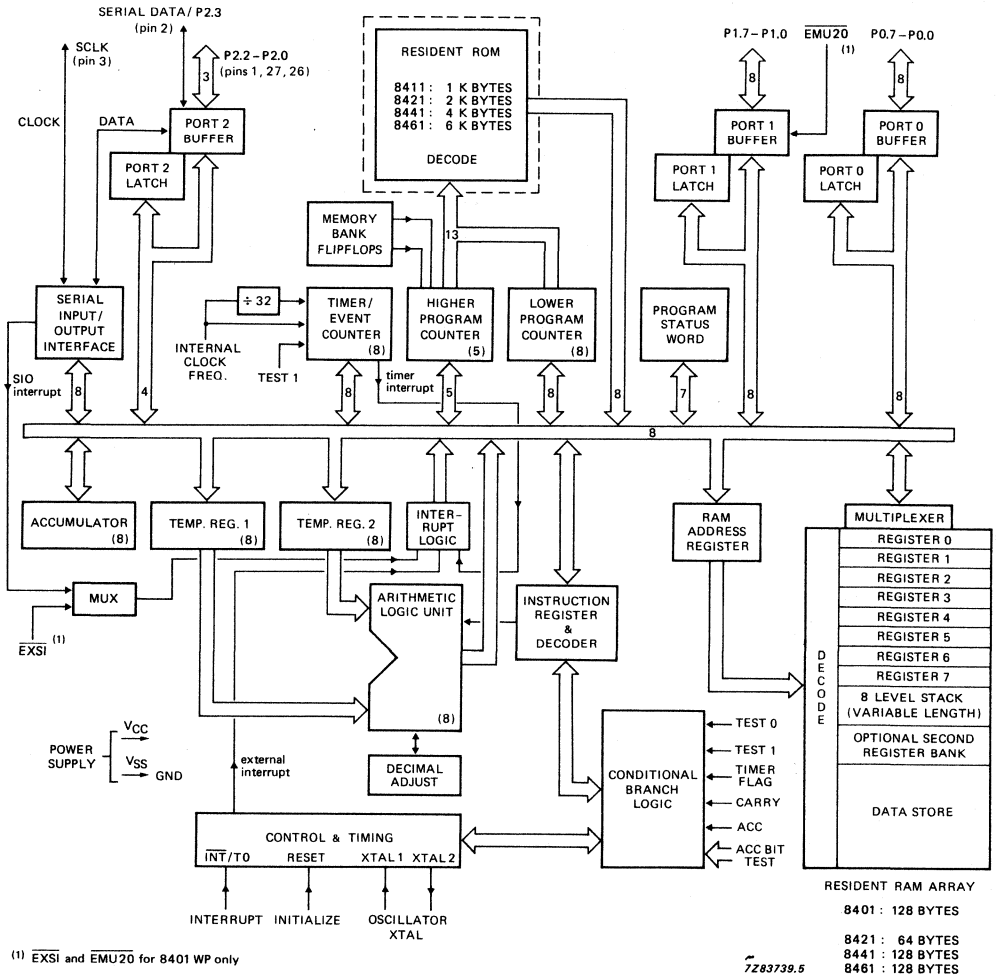



Fig. 1a Block diagram of the MAB84X1 family.

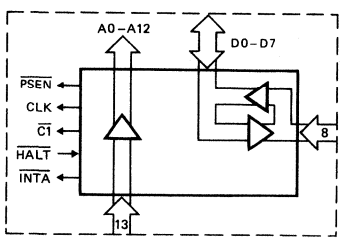



Fig. 1b Replacement for dotted part in Fig. 1a for the MAB8401WP bond-out version.

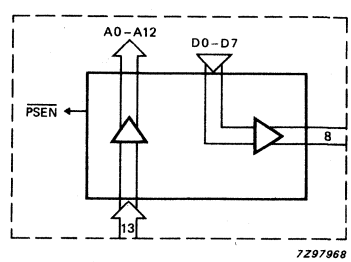



Fig. 1c Replacement of dotted part in Fig. 1a for the MAB8401B 'Piggy-back' version.

## PINNING

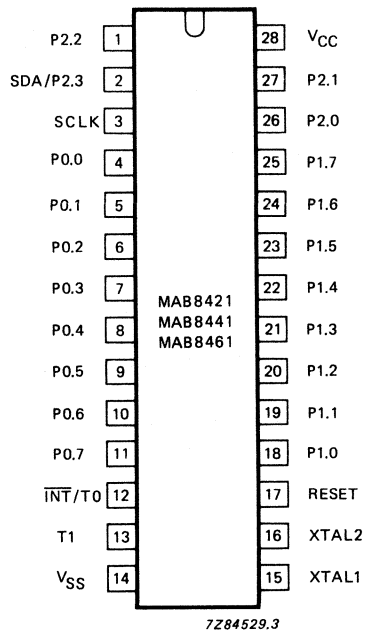
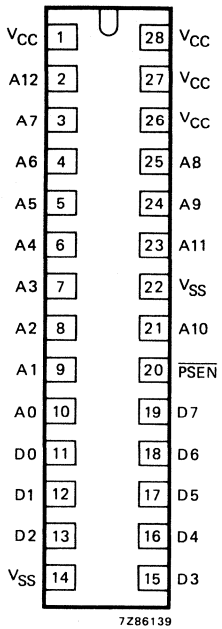



Fig. 2 Pinning diagram for mask-programmable devices MAB8421, MAB8441, MAB8461 and for MAB8401 'Piggy-back' version bottom pinning (for top pinning see Fig. 3).

## PINNING DESIGNATION

| | | |
|---|---|---|
| $V_{SS}$ | 14 | **Ground** |
| $V_{CC}$ | 28 | **Power supply,** + 5 V |
| P0.0 – P0.7 | 4 – 11 | **Port 0,** 8-bit quasi-bidirectional I/O port |
| P1.0 – P1.7 | 18 – 25 | **Port 1,** 8-bit quasi-bidirectional I/O port with 8-bit LED driver |
| P2.0 – P2.3 | 26, 27, 1, 2 | **Port 2,** 4-bit quasi-bidirectional I/O port; SDA/P2.3 is the serial data I/O in serial I/O mode |
| SCLK | 3 | Bidirectional clock for serial I/O |
| $\overline{INT}$/T0 | 12 | External interrupt input (sensitive to a negative-going edge min LOW > 7 clock pulses, min HIGH > 4 clock pulses), testable using the JT0 or JNT0 instructions. |
| T1 | 13 | Input pin, testable using the JT1 or JNT1 instructions. It can be designated as event counter input using the STRT CNT instruction. It can also be used to detect zero cross-over of slowly moving AC inputs. |
| RESET | 17 | Input to initialize the processor (active HIGH). |
| XTAL1 | 15 | Connection to timing component (crystal) that determines the frequency of the internal oscillator. It is also the input for an external clock source. |
| XTAL2 | 16 | Connection to other side of the timing component. |

**MAB8401B** (top pinning)




## PIN DESIGNATION

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

Fig. 3 Pinning diagram for MAB8401B 'Piggy-back' version top pinning (for bottom pinning see Fig. 2); to access a 2732 or 2764 EPROM.

**Note**

Access times for ROMS/EPROMS to be below 1 $\mu$s.

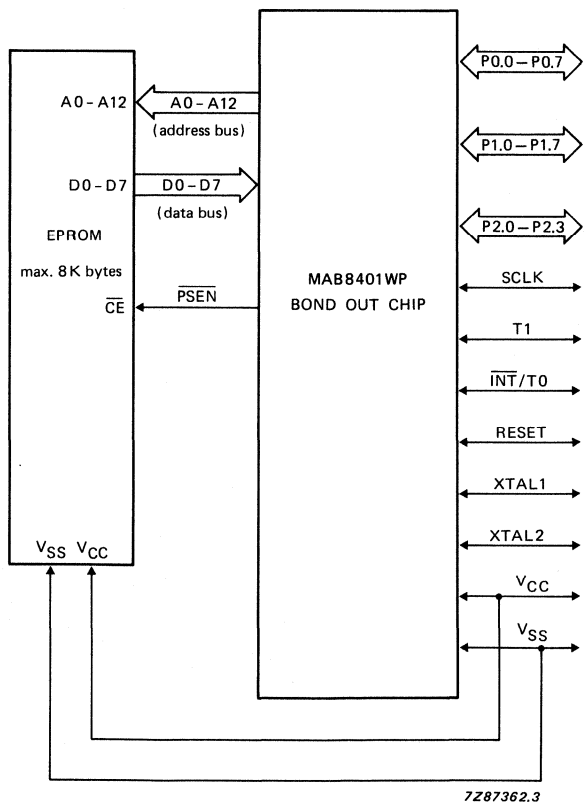



Fig. 3a Connection of EPROM to 'Piggy-back' package MAB8401B.

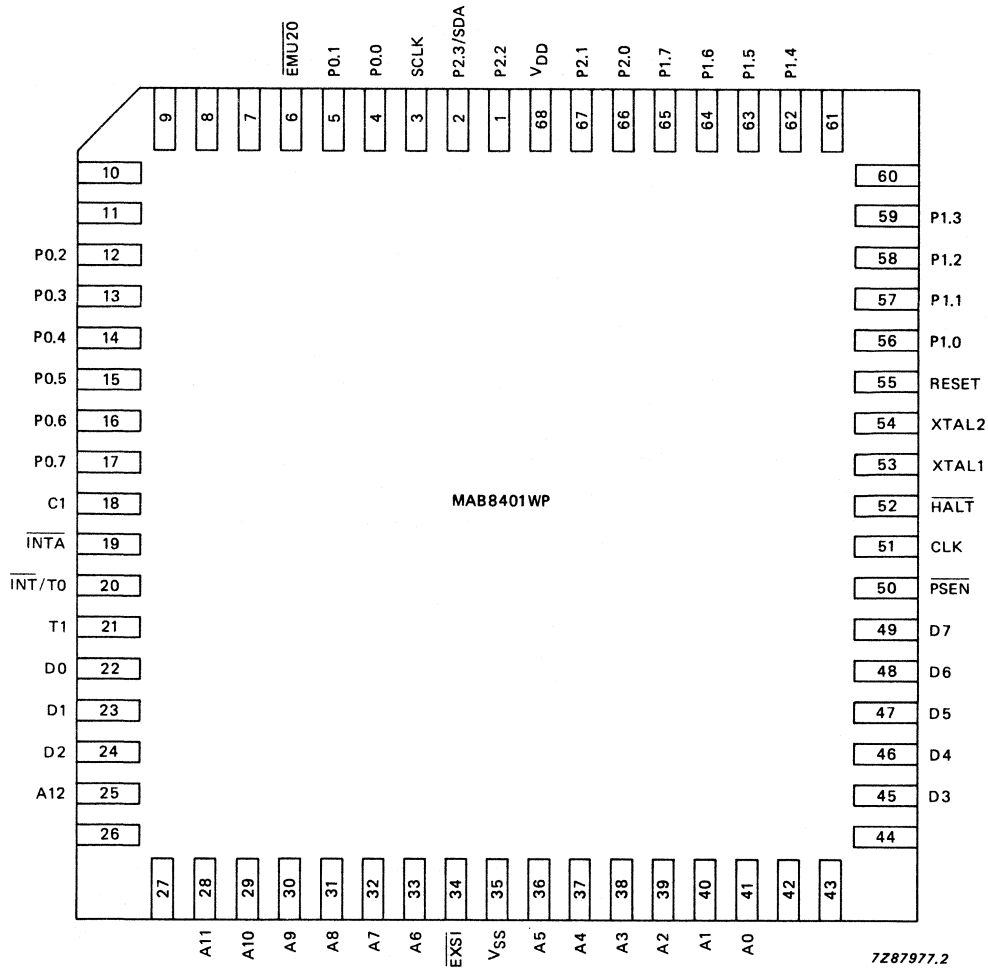



Fig. 4 Pinning diagram; PLCC.

## CHIP CARRIER DESIGNATION

| designation | pad no. | function |
|---|---|---|
| $V_{SS}$ | 35 | **Ground** |
| $V_{CC}$ | 68 | **Power supply,** + 5 V |
| P0.0 – P0.7 | 4–5, 12–17 | **Port 0,** 8-bit quasi-bidirectional I/O port |
| P1.0 – P1.7 | 56–59, 62–65 | **Port 1,** 8-bit quasi-bidirectional I/O port with 8-bit LED driver |
| P2.0 – P2.3 | 66, 67, 1, 2 | **Port 2,** 4-bit quasi-bidirectional I/O port; SDA/P2.3 is the serial data I/O in serial I/O mode |
| SCLK | 3 | Bidirectional clock for serial I/O |
| INT/T0 | 20 | External interrupt input (sensitive to a negative-going edge), testable using the JT0 or JNT0 instructions |

| | | |
|---|---|---|
| T1 | 21 | Input pin, testable using the JT1 or JNT1 instructions. It can be designated as event counter input using the STRT CNT inctruction. It can also be used to detect zero cross-over of slowly moving a.c. inputs. |
| RESET | 55 | Input to initialize the processor (active HIGH) |
| XTAL1 | 53 | Connection to timing component (e.g. crystal) that determines the frequency of the internal oscillator. It is also the input for an external clock source. |
| XTAL2 | 54 | Connection to other side of the timing component |
| $\overline{EXSI}$ | 34 | External serial I/O interrupt (active-LOW) for emulation of MAB/F8422/42. |
| A0–A12 | 41–36, 33–28 25 | Program memory address outputs (active HIGH); A0 = LSB, A12 = MSB. Address output change after begin $\phi$3 of TS8. |
| D0–D7 | 22–24, 45–49 | Data input lines (active HIGH) used for reading external program memory. D0 = LSB, D7 = MSB. |
| CLK | 51 | Clock output buffered from XTAL2. On the positive-going edge the (internal) $\phi$ clock goes HIGH. |
| $\overline{PSEN}$ | 50 | Program store enable. This signal is used for enabling the external EPROM (e.g. on the 'Piggy-back' version). For emulation, it enables the emulation memory and it indicates machine cycles. Active LOW during TS9,*TS10 of each machine cycle and TS1 of the following machine cycle. |
| $\overline{C1}$ | 18 | Cycle 1 indication output (active LOW). During emulation, this signal indicates the opcode fetch cycle (useful for external instruction decoding, real-time trace). Active from start of TS10 of the cycle preceding cycle 1, until the start of TS10 of cycle 1. |
| $\overline{HALT}$ | 52 | Halt input (active LOW). If activated, the current instruction is finished and the microcontroller stops execution (HALT mode). The next program counter address is available on the address bus. Program counter and timer/event counter are no longer updated. The serial I/O finishes the current transmit/receive action and goes into the idle state. Interrupts are *not* sampled in the HALT mode, they are only sampled when the microcontroller is running. Interrupt routines can be single-stepped as a normal program. |
| $\overline{INTA}$ | 19 | Interrupt acknowledge output (active LOW). It indicates any interrupt acception. Active from start of TS8 of the interrupted cycle, until start of TS7 of the second cycle of the (internally forced 'CALL vector address' instruction. During $\overline{INTA}$ active, the address bus shows the address that has been saved in the stack (return address); the C1 output indicates opcode fetch cycles as if a user CALL was executed. |
| $\overline{EMU20}$ | 6 | Emulate 20-pin version MAB/F8422/42 (active-LOW). |

* TS = Time slot, where 10 TS = 1 cycle.

## FUNCTIONAL DESCRIPTION (for more detail see 84XXX family specification)

### Bond-out version MAB8401WP

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

### Piggy-back version MAB8401B

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

### Program and data memory

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

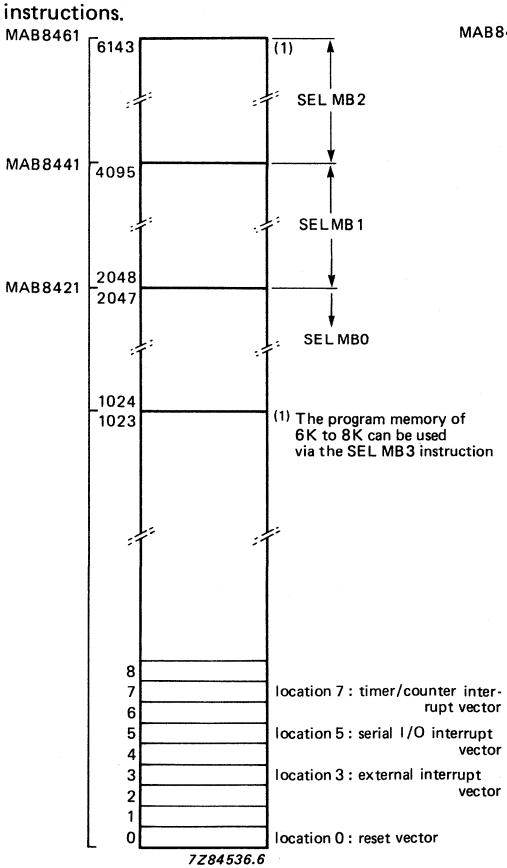



Fig. 5 The program memory map.

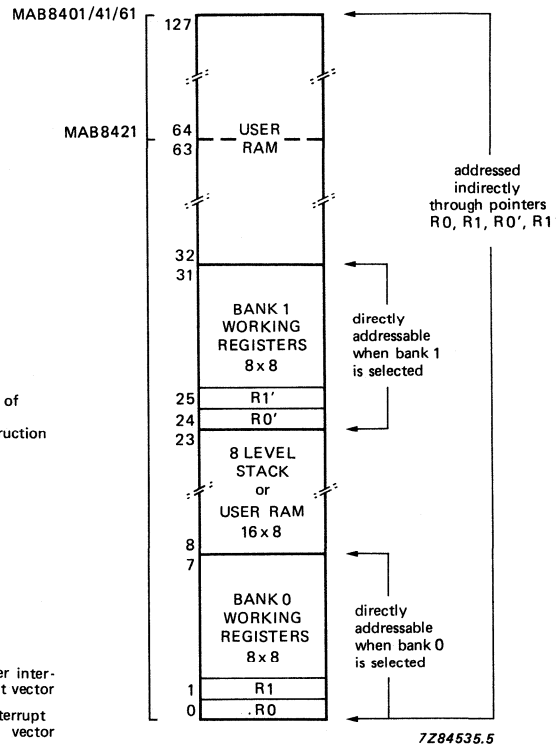



Fig. 6 The data memory map.

**FUNCTIONAL DESCRIPTION** (continued)

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

### On-chip peripheral functions

In addition to the CPU and memories, an interrupt system, I/O facilities, and an 8-bit timer/event counter are integrated on-chip to assist the CPU in repetitions, complicated or time-critical tasks. The I/O facilities include the I/O pins, parallel ports and a serial I/O port, consisting of a data line SDA shared with a parallel port line (P2.3), and a dedicated clock line SCLK.

### I/O facilities

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

- Two parallel ports of 8 lines (P0.0–P0.7, P1.0–P1.7). Each line of Port 1 can sink 10 mA.
- A parallel port of 4 lines (P2.0–P2.3).
- A serial I/O consisting of a data line shared with a parallel port line (P2.3) and a separate clock line SCLK;
- An external interrupt and test input $\overline{\text{INT}}$/T0, which when used as a test input can be tested by the conditional jump instructions JT0 or JNT0;
- A test input T1, which can alter program sequences when tested by conditional jump instructions JT1 or JNT1. T1 can also be used as an input to the timer/event counter or to detect zero cross-over of slowly moving AC signals.

All parallel port lines are available in three optional output configurations (except P2.3 – option 1 only):

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

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

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

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

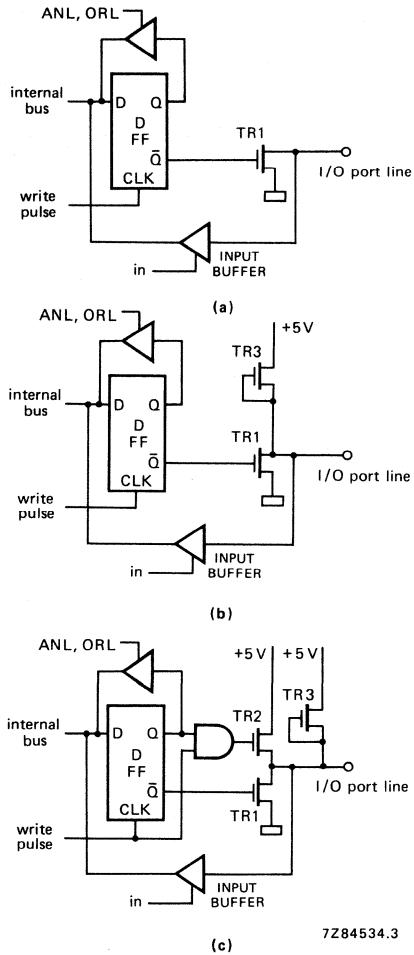



Fig. 7 Quasi-bidirectional I/O interface with (a) open drain output without pull-up transistor, (b) open drain output with pull-up transistor, (c) push-pull output with pull-up transistor.

**Serial I/O interface**

Figure 8 shows the serial I/O interface. The clock line of the serial bus has exclusive use of pin 3 (SCLK) while the data line shares pin 2 (serial data) with the I/O line P2.3 of port 2. When the serial I/O is enabled, P2.3 is disabled as a parallel port line (P2.3 and SCLK only open drain).

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

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

## FUNCTIONAL DESCRIPTION (continued)

**Serial I/O interface** (continued)

Data shift register S0

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

Serial I/O interface status word S1

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

MST and TRX

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

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

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

BB: Bus Busy

This bit indicates the status of the bus.

PIN: Pending Interrupt Not

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

ESO: Enable Serial Output

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

BC0, BC1 and BC2

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

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

AL: Arbitration Lost

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

AAS: Addressed As Slave

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

AD0: Address Zero

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

LRB: Last Received Bit

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

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

Serial clock control register S2

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

Address register S0'

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

Serial I/O interrupt logic

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

**Interrupt system**

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

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

## FUNCTIONAL DESCRIPTION (continued)

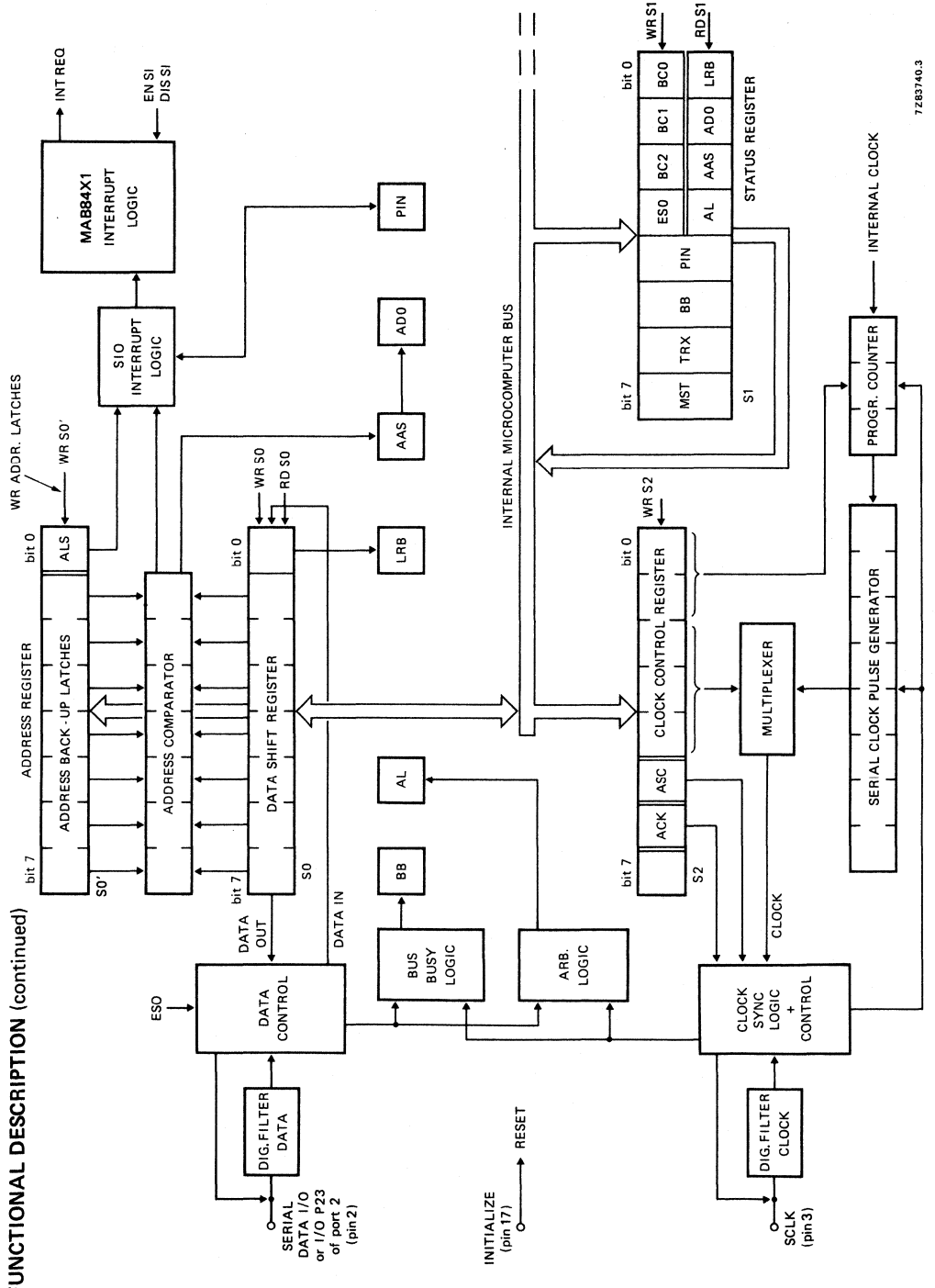



Fig. 8 The serial I/O interface.

**Test input T1**

The T1 input line can be used as:

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

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

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

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

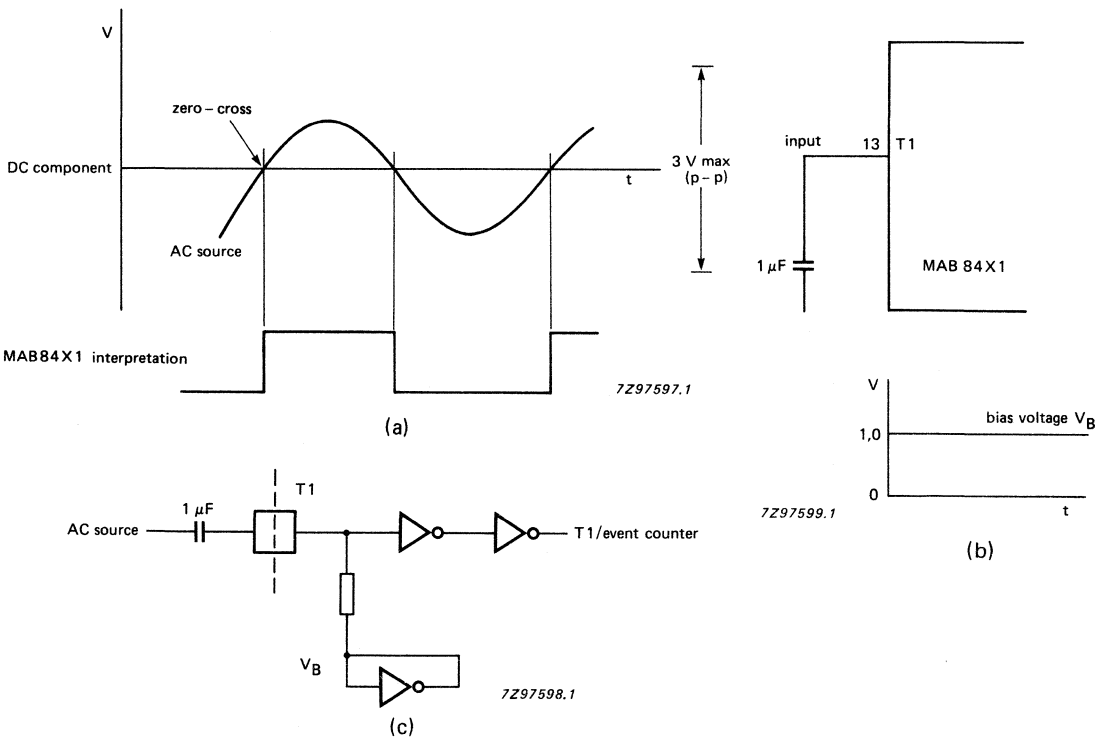



Figure 9 Zero-cross detection circuitry; (a) input waveform, (b) input diagram, (c) on-chip self-stabilized bias.

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

### High current outputs

Ten pins are provided that can sink high currents:

| | |
|---|---|
| – P2.3 (serial data), pin 2 | 5 mA at 0,45 V (open drain), |
| – SCLK, pin 3 | 5 mA at 0,45 V (open drain), |
| – P1.0 – P1.7 * | 10 mA at 1 V |

### Timer/event counter

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

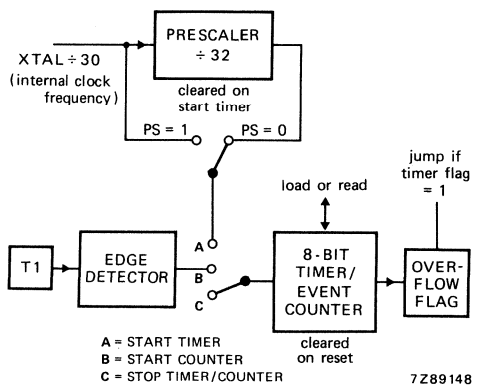



Fig. 10 The timer/event counter.

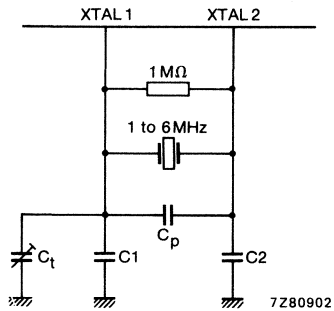
Differences between the MAB8021 and MAB8048 microcontrollers, and the MAB84X1 family.

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

* P1.0 to P1.7 may be connected in parallel if their logic outputs are always the same.

## OSCILLATOR CIRCUITRY

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




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

Fig. 11a Quartz crystal or ceramic resonator mode.

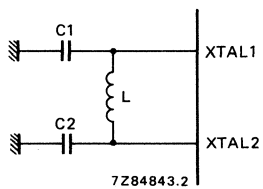



Fig. 11b LC pi-network.

**LC oscillator timing**

| frequency | C1 = C2 | L |
|---|---|---|
| 3,0 MHz | 33 pF | 100 $\mu$H |
| 4,0 MHz | 33 pF | 56 $\mu$H |
| 4,4 MHz | 33 pF | 47 $\mu$H |
| 5,0 MHz | 33 pF | 33 $\mu$H |
| 6,0 MHz | 33 pF | 22 $\mu$H |

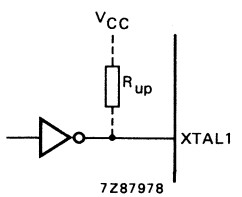



Fig. 11c External drive.

Drive XTAL1
Leave XTAL2 open

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

## PROGRAM STATUS WORD

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

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

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

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

bit 5 – not used (1);

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

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

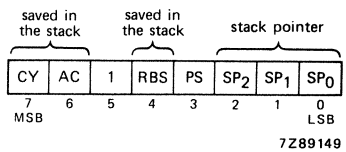



Fig. 12 Program status word.

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

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

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

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

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

**Table 2** Conditional branches

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

## RESET

A positive-going signal on the RESET input:

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

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

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

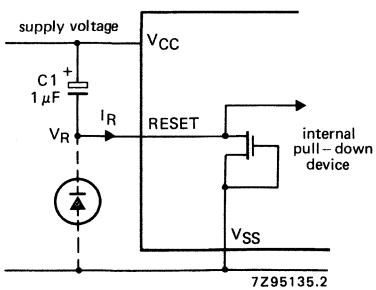



Fig. 13 Typical power-on reset circuitry.

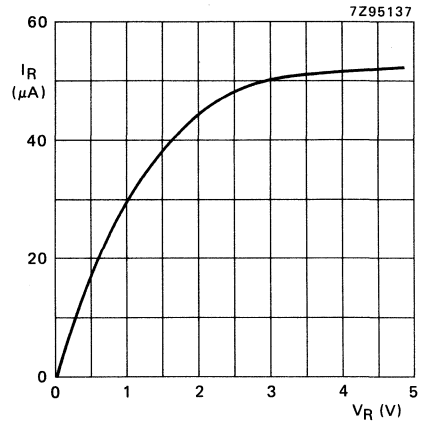



Fig. 14 Power-on reset input characteristics (typical).

## INSTRUCTION SET

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

Table 3 gives the instruction set of the MAB84X1 family and Table 4 shows the instruction map. The following symbols and abbreviations are used.

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

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

**Table 3 MAB84XX family instruction set**

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

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

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

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

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

**Notes** to Table 3.

1. PSW CY, AC affected
2. PSW CY affected
3. PSW PS affected
4. Execution of JTF and JNTF instructions resets the Timer Flag (TF).
5. PSW RBS affected
6. PSW $SP_0$, $SP_1$, $SP_2$ affected
7. (A) = 1111 P2.3, P2.2, P2,1, P2,0.
8. (S1) has a different meaning for read and write operation, see serial I/O interface.
9. Only for software-transfer from the MAB8021.

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

## Table 4 MAB84X1 family instruction set

first hexadecimal chracter of opcode ↓ / second hexadecimal chracter of opcode →

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | NOP | | | ADD A, # data | JMP page 0 | EN I | JNTF addr | DEC A | IN A,Pp 0 | 1 | 2 | | MOV A,Sn 0 | 1 | | |
| 1 | INC @Rr 0 | 1 | JB0 addr | ADDC A, # data | CALL page 0 | DIS I | JTF addr | INC A | 0 | 1 | 2 | INC Rr 3 | 4 | 5 | 6 | 7 |
| 2 | XCH A, @Rr 0 | 1 | | MOV A, # data | JMP page 1 | EN TCNTI | JNT0 addr | CLR A | 0 | 1 | 2 | XCH A,Rr 3 | 4 | 5 | 6 | 7 |
| 3 | XCHD A, @Rr 0 | 1 | JB1 addr | | CALL page 1 | DIS TCNTI | JT0 addr | CPL A | OUTL Pp,A 0 | 1 | 2 | | MOV Sn,A 0 | 1 | 2 | |
| 4 | ORL A, @Rr 0 | 1 | MOV A, T | ORL A, # data | JMP page 2 | STRT CNT | JNT1 addr | SWAP A | 0 | 1 | 2 | ORL A,Rr 3 | 4 | 5 | 6 | 7 |
| 5 | ANL A, @Rr 0 | 1 | JB2 addr | ANL A, # data | CALL page 2 | STRT T | JT1 addr | DA A | 0 | 1 | 2 | ANL A,Rr 3 | 4 | 5 | 6 | 7 |
| 6 | ADD A, @Rr 0 | 1 | MOV T, A | | JMP page 3 | STOP TCNT | | RRC A | 0 | 1 | 2 | ADD A,Rr 3 | 4 | 5 | 6 | 7 |
| 7 | ADDC A, @Rr 0 | 1 | JB3 addr | | CALL page 3 | | | RR A | 0 | 1 | 2 | ADDC A,Rr 3 | 4 | 5 | 6 | 7 |
| 8 | | | | RET | JMP page 4 | EN SI | | | ORL Pp, # data 0 | 1 | 2 | | | | | |
| 9 | OUTL PO, A | | JB4 addr | RETR | CALL page 4 | DIS SI | JNZ addr | CLR C | ANL Pp, # data 0 | 1 | 2 | | MOV Sn, # data 0 | 1 | 2 | |
| A | MOV @Rr, A 0 | 1 | | MOVP A, @A | JMP page 5 | SEL MB2 | | CPL C | 0 | 1 | 2 | MOV Rr, A 3 | 4 | 5 | 6 | 7 |
| B | MOV @Rr, # data 0 | 1 | JB5 addr | JMPP @A | CALL page 5 | SEL MB3 | | | 0 | 1 | 2 | MOV Rr, # data 3 | 4 | 5 | 6 | 7 |
| C | DEC @Rr 0 | 1 | | | JMP page 6 | SEL RB0 | JZ addr | MOV A, PSW | 0 | 1 | 2 | DEC Rr 3 | 4 | 5 | 6 | 7 |
| D | XRL A, @Rr 0 | 1 | JB6 addr | XRL A, # data | CALL page 6 | SEL RB1 | | MOV PSW, A | 0 | 1 | 2 | XRL A, Rr 3 | 4 | 5 | 6 | 7 |
| E | DJNZ @Rr, addr 0 | 1 | | | JMP page 7 | SEL MB0 | JNC addr | RL A | 0 | 1 | 2 | DJNZ Rr, addr 3 | 4 | 5 | 6 | 7 |
| F | MOV A, @Rr 0 | 1 | JB7 addr | | CALL page 7 | SEL MB1 | JC addr | RLC A | 0 | 1 | 2 | MOV A, Rr 3 | 4 | 5 | 6 | 7 |

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

**Table 5** MAB84X1 family instructions not in the MAB8048 instruction set

| serial I/O | register | control | conditional branch |
|---|---|---|---|
| MOV A, $S_n$<br>MOV $S_n$, A<br>MOV $S_n$, #data<br>EN SI<br>DIS SI | DEC @Rr<br>DJNZ @Rr, addr | SEL MB2<br>SEL MB3 | JNTF addr |

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

**Table 6** MAB8048 instructions not in the MAB84X1 family instruction set

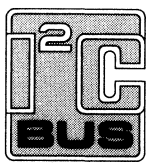
| data moves | flags | branch | control |
|---|---|---|---|
| MOVX A, @R<br>MOVX @R, A<br>MOVP3 A, @A<br>MOVD A, P<br>MOVD P, A<br>ANLD P, A<br>ORLD P, A | CLR F0<br>CPL F0<br>CLR F1<br>CPL F1 | * JNI addr<br>JF0 addr<br>JF1 addr<br><br><br>* replaced by<br>JT0 JNT0. | ENT0 CLK |

## ABSOLUTE MAXIMUM RATINGS

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

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

| parameter | symbol | min. | max. | unit |
|---|---|---|---|---|
| Input voltage on any pin with respect to ground ($V_{SS}$) | $V_I$ | −0,5 | +7 | V |
| Total power dissipation | | | | |
| SOT-117, 28-lead DIL | $P_{tot}$ | – | 1 | W |
| SOT-136, 28-lead DIL | $P_{tot}$ | – | 0,6 | W |
| Input/output current for all pins except port 1 | $I_I$, $I_O$ | – | 10 | mA |
| Input/output current for port 1 | $I_I$, $I_O$ | – | 20 | mA |
| Storage temperature | $T_{stg}$ | −65 | +150 | °C |
| Operating temperature | | | | |
| standard | $T_{amb}$ | 0 | +70 | °C |
| extended | $T_{amb}$ | −40 | +85 | °C |
| automotive | $T_{amb}$ | −40 | +110 | °C |




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

## DC CHARACTERISTICS

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

| parameter | conditions | symbol | min. | max. | unit |
|---|---|---|---|---|---|
| Supply current | | | | | |
| MAB | 0 to + 70 °C | $I_{CC}$ | – | 85 | mA |
| MAF | –40 to + 85 °C | $I_{CC}$ | – | 100 | mA |
| MAF84A | –40 to + 110 °C | $I_{CC}$ | – | 100 | mA |
| **Inputs** | | | | | |
| Input voltage LOW (except P2.3 and SCLK) | | $V_{IL}$ | –0,5 | 0,8 | V |
| Input voltage LOW (P2.3 and SCLK) | | $V_{IL1}$ | –0,5 | 1,5 | V |
| Input voltage HIGH (all inputs except XTAL1, P2.3 and SCLK) | | $V_{IH}$ | 2 | $V_{CC}$ + 0,5 | V |
| Input voltage HIGH (XTAL1, P2,3 and SCLK) | | $V_{IH1}$ | 3,0 | $V_{CC}$ + 0,5 | V |
| **Outputs** | | | | | |
| Output voltage LOW (P0.0–P0.7) | $I_{OL}$ = 1,6 mA | $V_{OL}$ | – | 0,45 | V |
| Output voltage LOW (P1.0–P1.7 for 8401/11/21/41/61) | $I_{OL12}$ = 10 mA | $V_{OL12}$ | – | 1,0 | V |
| Output voltage LOW (P2.0–P2.2) | $I_{OL2}$ = 1,6 mA | $V_{OL2}$ | – | 0,45 | V |
| Output voltage LOW (P2.3, SCLK) | $I_{OL3}$ = 5 mA | $V_{OL3}$ | – | 0,45 | V |
| Output voltage LOW (non-standard pins of bond-out versions) | $I_{OL4}$ = 0,4 mA | $V_{OL4}$ | – | 0,45 | V |
| Output voltage HIGH (all outputs unless open drain) | $I_{OH}$ = –50 μA | $V_{OH}$ | 2,4 | – | V |
| Output leakage current | $V_{SS} < V_I < V_{CC}$ | ± $I_{OL}$ | – | 10 | μA |

**AC CHARACTERISTICS** (all versions except bond-out)

$V_{CC}$ = 5 V ± 10%; $V_{SS}$ = 0 V.

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

**AC CHARACTERISTICS** (bond-out versions)

$V_{CC}$ = 5 V ± 10%; $V_{SS}$ = 0 V.

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

Note: CP = clock pulse.

**T1 ZERO-CROSS CHARACTERISTICS**

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

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

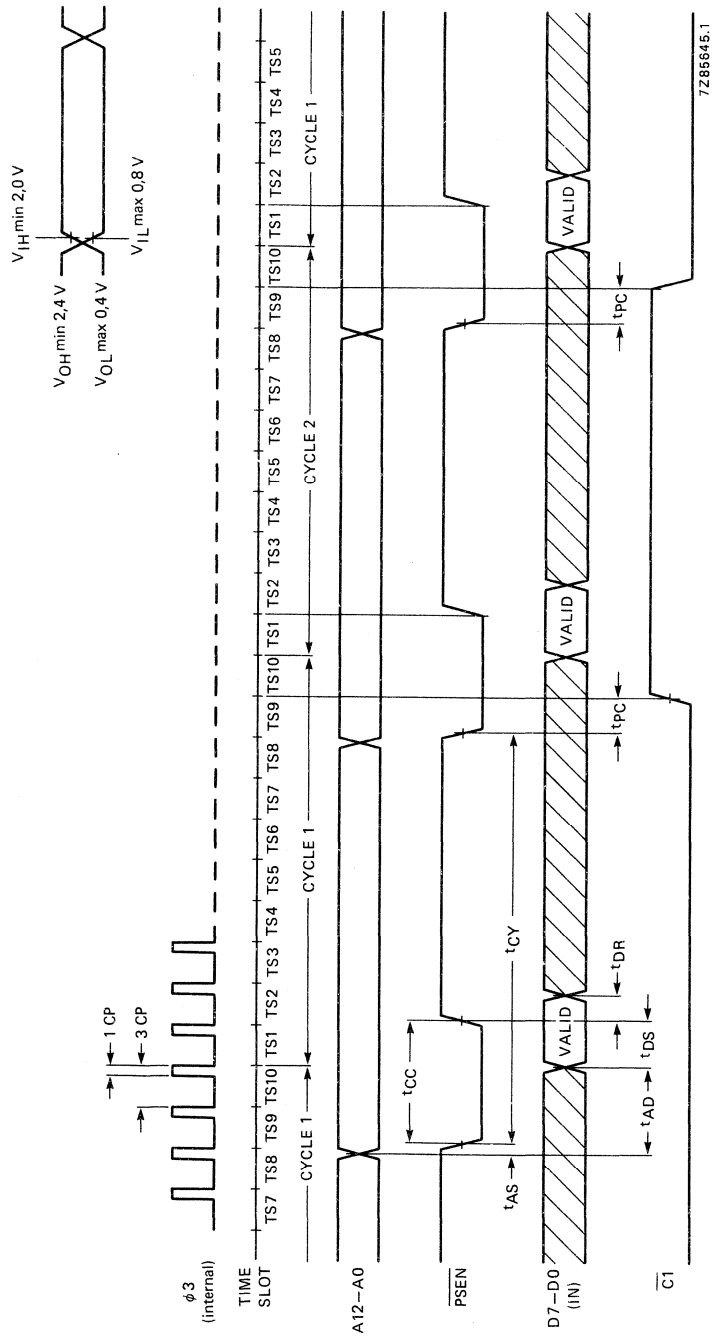



Fig. 15 Memory access timing MAB8401B/WP and I/O voltage parameters.

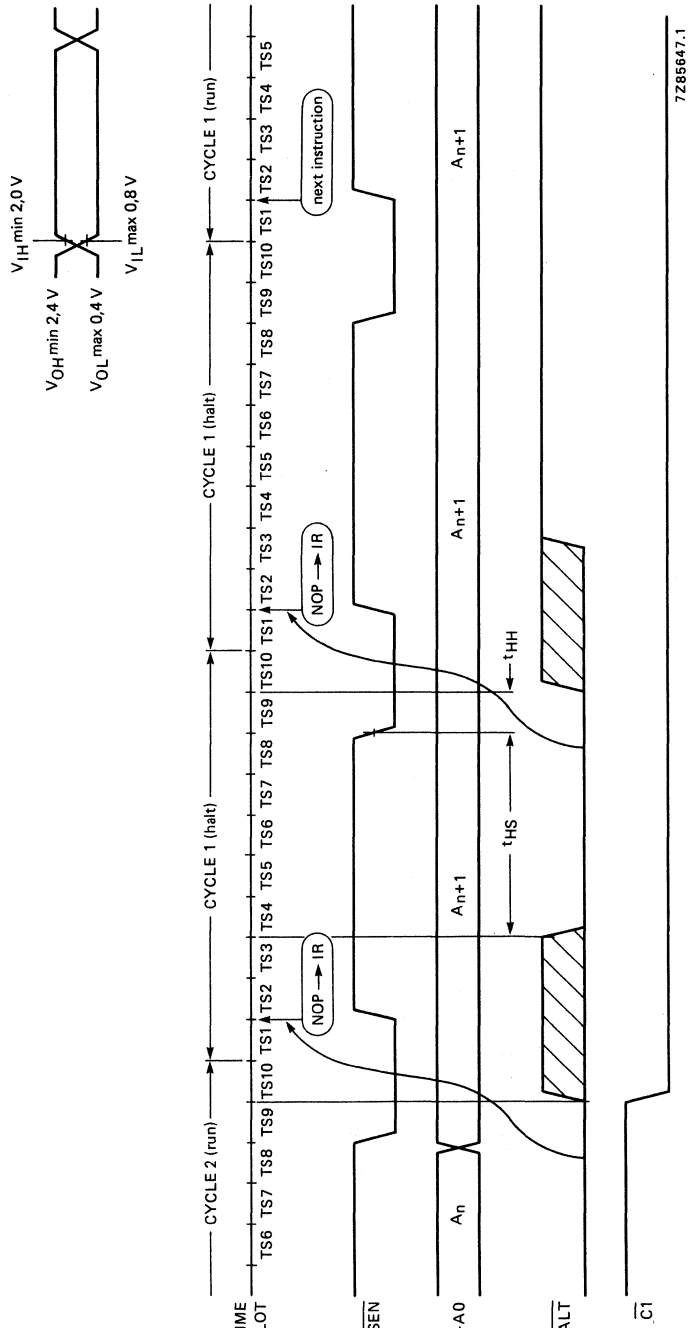



Fig. 16 $\overline{\text{HALT}}$ timing MAB8401WP and I/O voltage parameters.

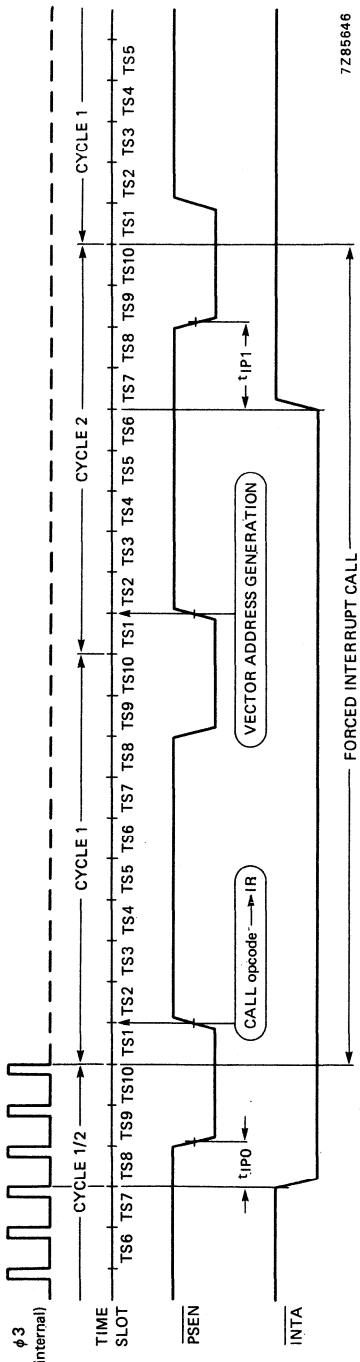



Fig. 17 $\overline{\text{INTA}}$ timing MAB8401WP.

# SECTION 3 84CXXX DERIVATIVES

# SINGLE-CHIP 8-BIT MICROCONTROLLER WITH I²C-BUS INTERFACE

## DESCRIPTION

An advanced CMOS process is used to manufacture the PCF84C00 microcontroller. The device has 20 quasi-bidirectional I/O port lines, a serial I/O interface, a single-level vectored interrupt structure, an 8-bit timer/event counter and on-chip clock oscillator and clock circuits.

This efficient controller also performs well as an arithmetic processor. It has facilities for both binary and BCD arithmetic plus bit-handling capabilities. The instruction set is similar to that of the MAB8048.

This microcontroller is a member of the 84CXXX family. For detailed information, consult the 84CXXX family specification.

### Features

- 8-bit CPU, RAM, I/O in a single 28-lead or 56-lead package
- ROM-less version, external program memory
- 256 x 8 RAM
- 20 quasi-bidirectional I/O port lines
- Two test inputs, one of which is also the external interrupt input
- Single-level vectored interrupts; external, timer/event counter and serial I/O
- I²C hardware interface for serial data transfer on two lines (serial I/O data via an existing port line and clock via a dedicated line)
- 8-bit programmable timer/event counter
- Clock frequency range: 100 kHz to 10 MHz
- Over 80 instructions (similar to those of the MAB8048) all of 1 or 2 cycles
- Single supply voltage (2.5 to 5.5 V)
- STOP and IDLE modes
- Power-on reset circuit
- Operating temperature range: −40 to +85 °C.

### For following sections see 84CXXX family specification

Program memory
Data memory
Program counter stack
IDLE and STOP modes
I/O facilities
Serial I/O
Interrupts
Oscillator
Timer/event counter
Program status word
Program counter
Central processing unit
Conditional branch logic
Test input T1
Power-on reset
Instruction set

## PACKAGE OUTLINES

PCF84C00B : 28-lead 'piggy-back' package (supports up to 28-pin EPROM).
PCF84C00T : 56-lead mini-pack; plastic (VSO56, SOT190).

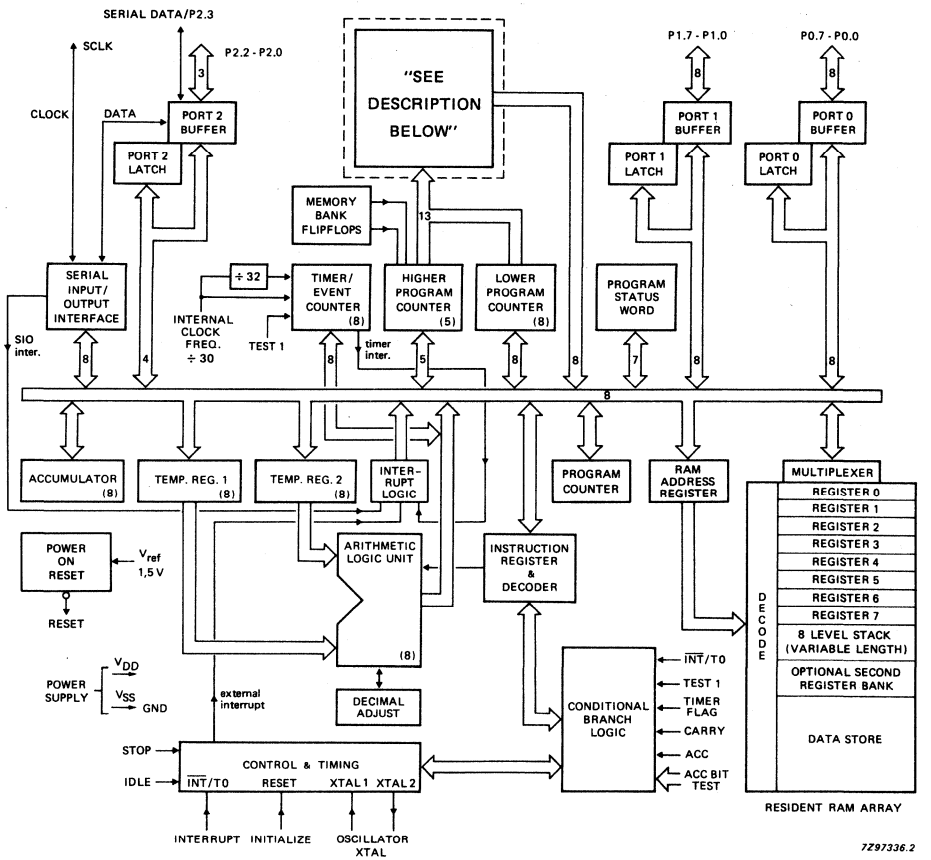



Fig. 1 Block diagram.

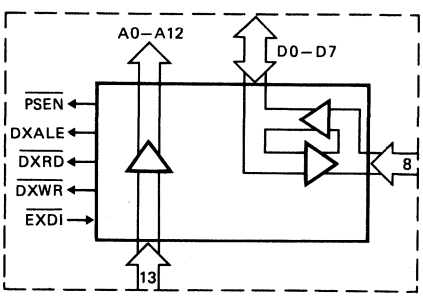



Fig. 1a Replacement of dotted section in Fig. 1, for the PCF84C00T ROM-less version.

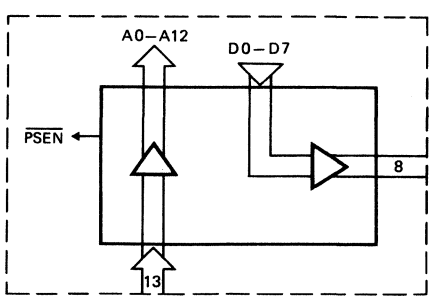



Fig. 1b Replacement of dotted section in Fig. 1, for the PCF84C00B 'piggy-back' version.

## PINNING

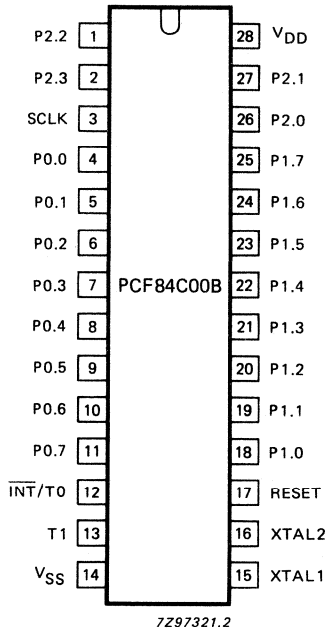



Fig.2 Bottom pinning diagram.

## PIN DESIGNATION

| *Pin* | *Symbol* | *Type* | *Function* |
|---|---|---|---|
| 3 | SCLK | I/O | **Clock:** bidirectional clock for serial I/O. |
| 4-11 | P0.0-P0.7 | I/O | **Port 0:** 8-bit quasi-bidirectional I/O port. |
| 12 | $\overline{\text{INT}}$/T0 | I | **Interrupt/Test 0:** external interrupt input (negative edge triggered)/ test input pin; when used as a test input, this pin is directly tested by conditional branch instructions JT0 and JNT0. |
| 13 | T1 | I | **Test 1:** test input pin, directly tested by conditional branch instructions JT1 and JNT1. T1 may also be selected as an input to the 8-bit timer/event counter via the STRT CNT instruction. |
| 14 | $V_{SS}$ | I | **Ground:** circuit earth potential. |
| 15 | XTAL 1 | I | **Oscillator input:** input from a crystal which determines the internal oscillator frequency or an external clock generator. |
| 16 | XTAL 2 | I/O | **Oscillator output:** output of the inverting amplifier. |
| 17 | RESET | I/O | **Reset input:** used to initialize the microcontroller (active HIGH); also output of power-on-reset circuit. |
| 18-25 | P1.0-P1.7 | I/O | **Port 1:** 8-bit quasi-bidirectional I/O port. |
| 26, 27, 1, 2 | P2.0-P2.3 | I/O | **Port 2:** 4-bit quasi-bidirectional I/O port. P2.3 is the serial data input/output in serial I/O mode. |
| 28 | $V_{DD}$ | I | **Power supply:** 2,5 V to 5,5 V. |

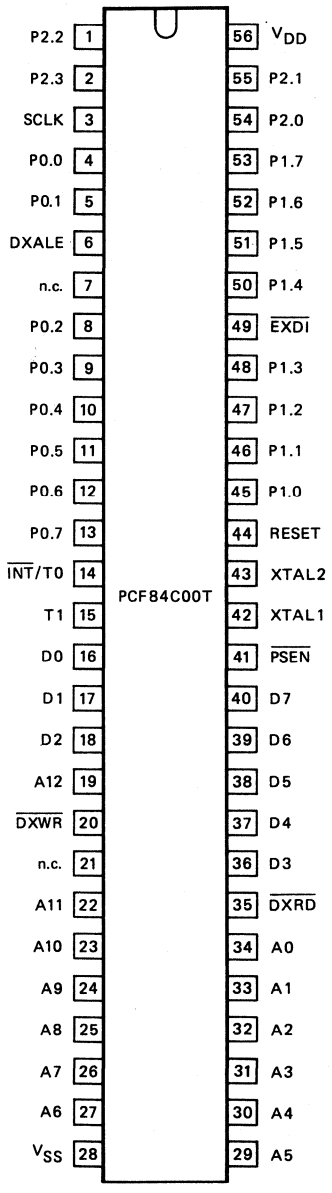



Fig. 4 Pinning diagram; ROM-less version PCF84C00T.

The PCF84C00T may be used for prototyping future derivatives of the PCF84CXX family or for low volume production applications. This device is packaged in a 56-lead VSO outline. Additional signals are available (see pinning information following) to control external program memory and derivative functions.

**PIN FUNCTION** (continued)

| *Pin* | *Symbol* | *Type* | *Function* |
|---|---|---|---|
| 34-29, 27-22, 19 | A00-A12 | O | **Address bus.** For external memory and pheripherals. |
| 16-18, 36-40 | D0-D7 | I/O | **Data bus.** For external memory and pheripherals. The specified STOP mode supply current is valid only if external pull-ups are connected to all data lines. |
| 41 | $\overline{\text{PSEN}}$ | O | **Program store enable** (active LOW). $\overline{\text{PSEN}}$ is used to enable external program memory and is active during TS9 and TS10 of each machine cycle and TS1 of each following cycle. $\overline{\text{PSEN}}$ is HIGH during the STOP mode. |
| 6 | DXALE | O | **Address latch enable.** On the falling edge of DXALE, the Dx address can be latched in an external latch. This signal occurs only during execution of the MOV Dx,A, MOV A,Dx, ANL Dx,A and ORL Dx,A instructions, with x = 0 to 255. It is active during TS10 of cycle 1 and the first half of TS1 of cycle 2. |
| 35 | $\overline{\text{DXRD}}$ | O | **Read strobe** (active LOW). When this signal is active, external registers emulating Dx registers can be read by the data bus. This signal occurs only during execution of MOV A,Dx, ANL Dx,A and ORL Dx,A instructions, with x = 0 to 255. It is active during TS3 and TS4 of cycle 2. |
| 20 | $\overline{\text{DXWR}}$ | O | **Write strobe** (active LOW). On the rising edge, data on D0-D7 may be written to external registers. This signal occurs only during MOV Dx,A, ANL Dx,A and ORL Dx,A instructions, with x = 0 to 255. It is active during TS7 of cycle 2. |
| 49 | $\overline{\text{EXDI}}$ | I | **External derivative interrupt** (active LOW). $\overline{\text{EXDI}}$ is 'OR-ed' with the internal serial interrupt and can be used to initiate an interrupt from external hardware emulating derivative functions. $\overline{\text{EXDI}}$ is pulled HIGH internally. The derivative interrupt is polled during time slot TS6*, and is only accepted if an EN SI instruction has been executed and the device is not already executing an interrupt routine. Derivative interrupts are not latched in the PCF84C00. |

* The interrupt signal must remain active until the vector address (05 H) is present on the address bus.

## FUNCTIONAL DESCRIPTION

### ROM-less version PCF84C00T

The PCF84C00T microcontroller contains no on-chip ROM, but has all address and data lines brought out to access an external ROM or EPROM. This version has more pins than the PCF84CXXX with on-chip ROM (see Fig. 1a). The PCF84C00T can address up to 8 K bytes of external program memory, and has 256 bytes of internal data RAM.

### 'Piggy-back' version PCF84C00B

The PCF84C00B package has standard pinning on the bottom to facilitate insertion as a mask-programmed device. An EPROM can be mounted on top in an additional socket. The total package height is greater than the height of a standard DIL package. The PCF84C00B can address up to 8 K bytes of external ROM/RAM, and has 256 bytes of internal data RAM.

### Data memory

Data memory consists of 256 bytes of random-access memory (RAM). All locations are indirectly addressable using RAM pointer registers; up to 16 designated locations are directly addressable. Memory also includes an 8-level program counter stack addressed by a 3-bit stack pointer.

### I/O facilities

Each device has 23 I/O lines arranged as follows:

- Port 0 8-bit parallel port (P0.0-P0.7)
- Port 1 8-bit parallel port (P1.0-P1.7)
- Port 2 4-bit parallel port (P2.0-P2.3)
- $\overline{\text{SCLK}}$ serial I/O clock line
- $\overline{\text{INT}}$/T0 external interrupt and test input. When used as a test input, it can be directly tested by conditional branch instructions JT0 and JNT0
- T1 test input which can alter program sequences when tested by conditional jump instructions JT1 and JNT1. T1 also functions as an input to the 8-bit timer/event counter

### Reset

A positive-going signal on the RESET input/output:

- Sets the program counter to zero
- Selects location 0 of memory bank 0 and register bank 0
- Sets the stack pointer to zero (000); pointing to RAM address 8
- Disables the interrupts (external, timer and serial I/O)
- Stops the timer/event counter, then sets it to zero
- Sets the timer prescaler to divide by 32
- Resets the timer flag
- Sets all ports except P2.3 to input mode
- Sets the serial I/O to slave receiver mode and disables the serial I/O
- Cancels IDLE and STOP mode

A negative-going signal on the RESET input/output:

- Sets P2.3/SDA and SCLK to HIGH after a maximum of 30 clock pulses
- Sets the serial I/O to slave receiver mode and disables the serial I/O after a maximum of 30 clock pulses
- Starts program execution after 1866 clock pulses

## RATINGS

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

| | | | | |
|---|---|---|---|---|
| Supply voltage | $V_{DD}$ | | −0,8 to +8 | V |
| All input voltages | $V_I$ | | −0,5 to $V_{DD}$ +0,5 | V |
| DC current into any input or output | $\pm I_I$, $\pm I_O$ | max. | 10 | mA |
| Total power dissipation (see note) | $P_{tot}$ | max. | 125 | mW |
| Storage temperature range | $T_{stg}$ | | −65 to +150 | °C |
| Operating ambient temperature range (if $P_{tot}$ max. = 100 mW) | $T_{amb}$ | | −40 to +70 | °C |
| Operating ambient temperature range (if $P_{tot}$ max. = 30 mW) | $T_{amb}$ | | −40 to +85 | °C |
| Operating junction temperature | $T_j$ | max. | 90 | °C |

## HANDLING

Inputs and outputs are protected against electrostatic discharge in normal handling. However, it is good practice to take normal precautions appropriate to handling MOS devices (see 'Handling MOS devices').

### Note

Thermal resistance (junction to ambient)

| | | | | |
|---|---|---|---|---|
| for SOT117 | $R_{th\ j\text{-}a}$ | max. | 120 | K/W |
| for SOT136A | $R_{th\ j\text{-}a}$ | max. | 150 | K/W |
| for SOT190 | $R_{th\ j\text{-}a}$ | max. | 110 | K/W |

## DC CHARACTERISTICS

$V_{DD}$ = 2,5 to 5,5 V; $V_{SS}$ = 0 V; $T_{amb}$ = −40 to +85 °C; all voltages with respect to $V_{SS}$; unless otherwise specified.

| parameter | symbol | min. | typ. | max. | unit |
|---|---|---|---|---|---|
| Supply voltage | | | | | |
| operating (see Fig. 9) | $V_{DD}$ | 2,5 | – | 5,5 | V |
| Supply current | | | | | |
| operating | | | | | |
| at $V_{DD}$ = 5 V; $f_{XTAL}$ = 10 MHz (note 2) | $I_{DD}$ | – | 1,6 | 3,2 | mA |
| at $V_{DD}$ = 5 V; $f_{XTAL}$ = 6 MHz (note 2) | $I_{DD}$ | – | 1 | 2 | mA |
| at $V_{DD}$ = 3 V; $f_{XTAL}$ = 3,58 MHz (note 2) | $I_{DD}$ | – | 0,3 | 0,6 | mA |
| IDLE mode | | | | | |
| at $V_{DD}$ = 5 V; $f_{XTAL}$ = 10 MHz (note 2) | $I_{DD}$ | – | 0,8 | 1,6 | mA |
| at $V_{DD}$ = 5 V; $f_{XTAL}$ = 6 MHz (note 2) | $I_{DD}$ | – | 0,5 | 1 | mA |
| at $V_{DD}$ = 3 V; $f_{XTAL}$ = 3,58 MHz (note 2) | $I_{DD}$ | – | 0,15 | 0,4 | mA |
| STOP mode (see Fig.15, note 1 and note 2) | | | | | |
| at $V_{DD}$ = 2,5 V; $T_{amb}$ = 25 °C | $I_{DD}$ | – | 1,2 | 2,5 | μA |
| at $V_{DD}$ = 2,5 V; $T_{amb}$ = 85 °C | $I_{DD}$ | – | – | 10 | μA |
| **Inputs** | | | | | |
| Input voltage LOW | $V_{IL}$ | 0 | – | $0,3V_{DD}$ | V |
| Input voltage HIGH | $V_{IH}$ | $0,7V_{DD}$ | – | $V_{DD}$ | V |
| Input leakage current | | | | | |
| at $V_{SS} < V_I < V_{DD}$ | $\pm I_{IL}$ | – | – | 1 | μA |
| **Outputs** | | | | | |
| Output sink current LOW | | | | | |
| at $V_{DD}$ = 5 V ± 10%; $V_O$ = 0,4 V | | | | | |
| except P2.3/SDA, SCLK (see Fig.11) and port 1 | $I_{OL}$ | 1,6 | 3 | – | mA |
| P2.3/SDA, SCLK (see Fig.12) | $I_{OL}$ | 3 | – | – | mA |
| Pull-up output source current HIGH (see Fig.13) | | | | | |
| at $V_{DD}$ = 5 V ± 10%; $V_O$ = $0,7V_{DD}$ | $-I_{OH}$ | 40 | – | – | μA |
| at $V_{DD}$ = 5 V ± 10%; $V_O$ = $V_{SS}$ | $-I_{OH}$ | – | – | 400 | μA |
| Push-pull output source current HIGH | | | | | |
| at $V_{DD}$ = 5 V ± 10%; $V_O$ = $V_{DD}$−0,4 V | $-I_{OH}$ | 1,6 | 3 | – | mA |

Note 1: Crystal connected between XTAL1 and XTAL2; T1 = $V_{SS}$; $\overline{INT}$ = $V_{DD}$.

Note 2: $V_{IL}$ = $V_{SS}$; $V_{IH}$ = $V_{DD}$; all outputs unloaded; all open drain outputs connected to $V_{SS}$.

## AC CHARACTERISTICS

$V_{DD}$ = 2,5 to 5,5 V; $V_{SS}$ = 0 V; $T_{amb}$ = −40 to +85 °C. All voltages with respect to $V_{SS}$ unless otherwise specified.

| parameter | symbol | min. | typ. | max. | unit |
|---|---|---|---|---|---|
| Rise time all outputs (note 1) | $t_R$ | – | 30 | – | ns |
| Fall time all outputs (note 1) | $t_F$ | – | 30 | – | ns |
| Cycle time (= 30 CP; note 2) | $t_{CY}$ | 3 | – | 30 | μs |
| **PCF84C00/non-standard pins:** | | | | | |
| Control pulse width | $t_{CC}$ | – | 9 | – | CP |
| Address to $\overline{PSEN}$ set-up | $t_{AS}$ | – | 1,5 | – | CP |
| Data to $\overline{PSEN}$ set-up | $t_{DS}$ | – | 2 | – | CP |
| Data hold time | $t_{DR}$ | 0 | – | – | ns |
| Data out to $\overline{DWXR}$ set-up | $t_{SDO}$ | – | 2 | – | CP |
| Data out to $\overline{DXWR}$ hold | $t_{HDO}$ | – | 1 | – | CP |
| Time from DXALE to $\overline{PSEN}$ | $t_{SLPH}$ | – | 1,5 | – | CP |
| Data-in to $\overline{DXRD}$ set-up | $t_{DS1}$ | – | 2,5 | – | CP |
| Data-in to $\overline{DXRD}$ hold | $t_{DR1}$ | 0 | – | – | ns |
| HIGH time of DXALE | $t_{DXALE}$ | – | 4,5 | – | CP |
| LOW time of $\overline{DXRD}$ | $t_{DXRD}$ | – | 6 | – | CP |
| LOW time of $\overline{DXWR}$ | $t_{DXWR}$ | – | 3 | – | CP |

**Notes**

1. At $V_{DD}$ = 5 V; $T_{amb}$ = + 25 °C; $C_L$ = 50 pF.
2. 1 Time slot (TS) = 3 CP, 1 clock pulse (CP) = $1/f_{XTAL}$.

**AC CHARACTERISTICS** (continued)

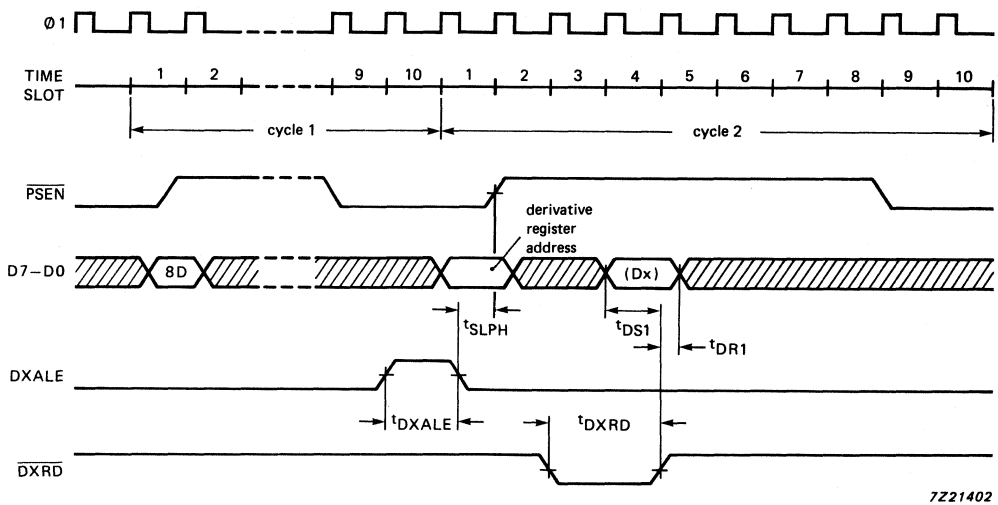



Fig. 5 MOV A,Dx timing (PCF84C00T only).

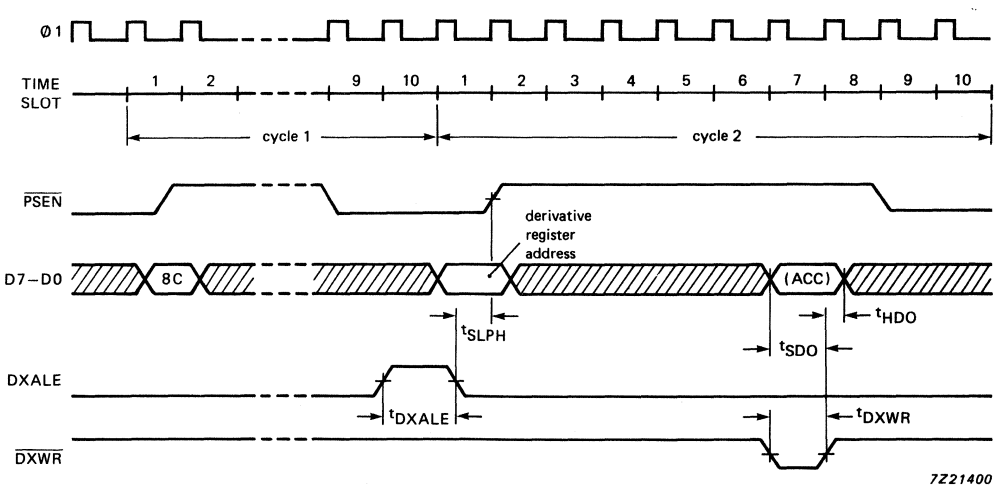



Fig. 6 MOV Dx,A timing (PCF84C00T only).

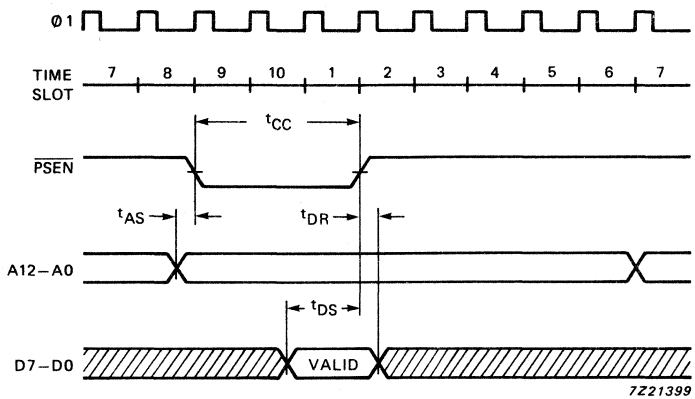



Fig. 7 External memory access timing (PCF84C00T and PCF8400B).

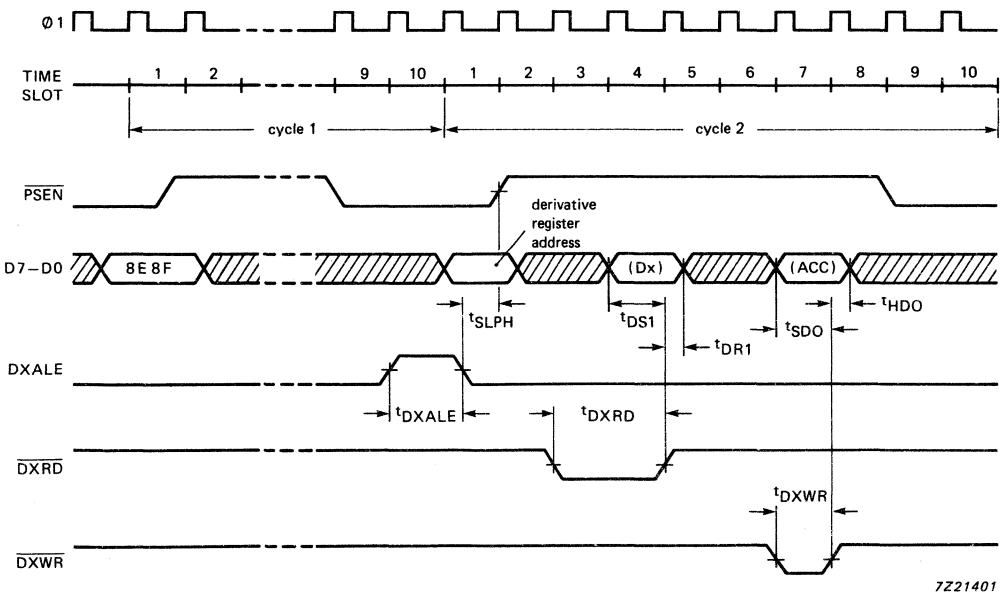



Fig. 8 ANL/ORL derivative interface timing (PCF84C00T only).

## AC CHARACTERISTICS (continued)

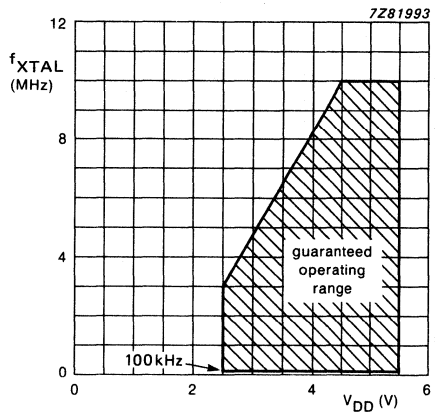



Fig.9 Maximum clock frequency ($f_{XTAL}$) as a function of the supply voltage (PCF84C00).

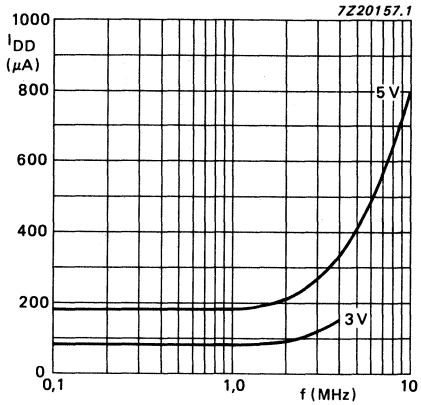



Fig.10 Typical supply current during IDLE mode as a function of frequency at $V_{DD}$ = 3 V and $V_{DD}$ = 5 V.

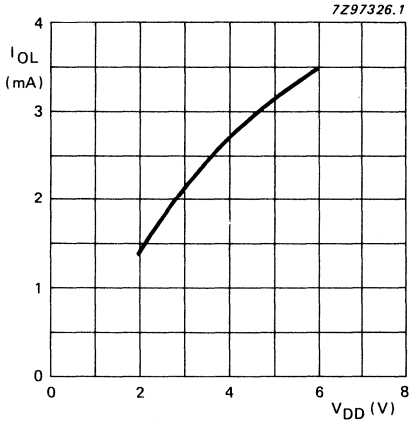



Fig.11 Typical output sink current ($I_{OL}$), outputs P0.0 to P0.7 and P2.0 to P2.2, as a function of the supply voltage ($V_{DD}$); $V_O$ = 0.4 V.

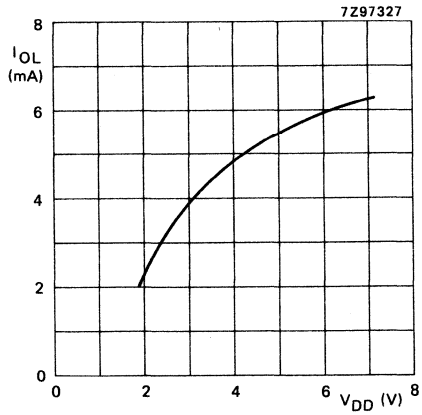



Fig.12(a) Typical output sink current ($I_{OL}$), outputs P2.3/SDA and SCLK, as a function of the supply voltage ($V_{DD}$); $V_O$ = 0.4 V.

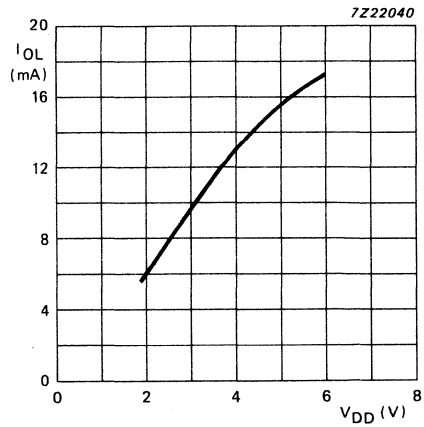
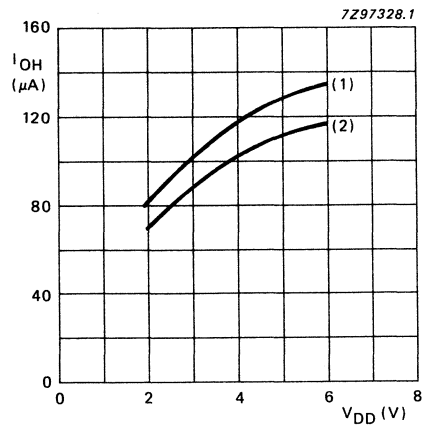



Fig.12(b) Typical output sink current ($I_{OL}$), outputs P1.0 to P1.7, as a function of the supply voltage ($V_{DD}$); $V_O$ = 1.2 V.




(1) $V_O = V_{SS}$
(2) $V_O = 0.7\ V_{DD}$

Fig.13 Typical output source current ($-I_{OH}$) as a function of the supply voltage ($V_{DD}$).

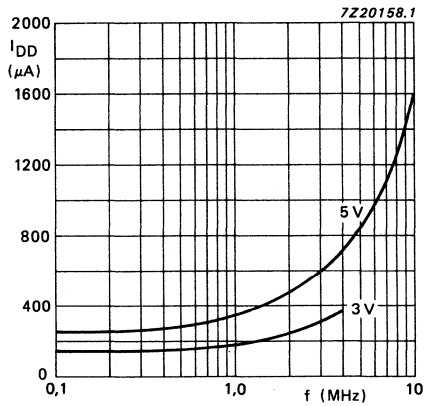
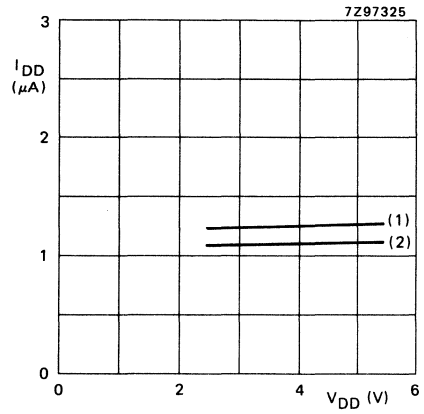



Fig. 14 Typical supply current during operating mode as a function of frequency at $V_{DD}$ = 3 V and $V_{DD}$ = 5 V.

**AC CHARACTERISTICS** (continued)




(1) $T_{amb}$ = 85 °C
(2) $T_{amb}$ = 25 °C

Fig.15 Typical supply current ($I_{DD}$) in STOP mode as a function of the supply voltage ($V_{DD}$).

**Table 4** Input timing shown in Fig.16

| symbol | timing |
|---|---|
| $t_{BUF}$ | $\geqslant 14t_{XTAL}$ |
| $t_{HD;STA}$ | $\geqslant 14t_{XTAL}$ |
| $t_{HIGH}$ | $\geqslant 17t_{XTAL}$ |
| $t_{LOW}$ | $\geqslant 17t_{XTAL}$ |
| $t_{SU;STO}$ | $\geqslant 14t_{XTAL}$ |
| $t_{HD;DAT}$ | $> 0$ |
| $t_{SU;DAT}$ | $\geqslant 250$ ns |
| $t_{RD}$ | $\leqslant 1$ μs |
| $t_{RC}$ | $\leqslant 1$ μs |
| $t_{FD}$ | $\leqslant 1$ μs |
| $t_{FC}$ | $\leqslant 0{,}3$ μs |

**Notes to Table 4**

$t_{XTAL}$ = one period of the XTAL input frequency ($f_{XTAL}$)
= 167 ns for $f_{XTAL}$ = 6 MHz
These figures apply to all modes.

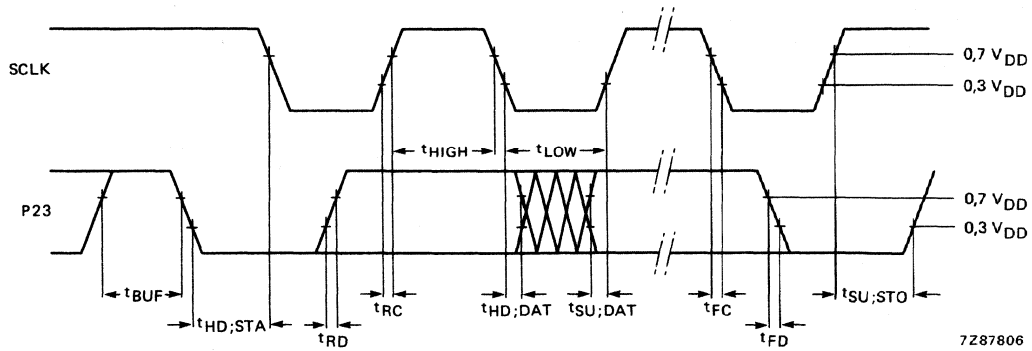



Fig.16 Timing requirements for the P2.3 and SCLK *input* signals.

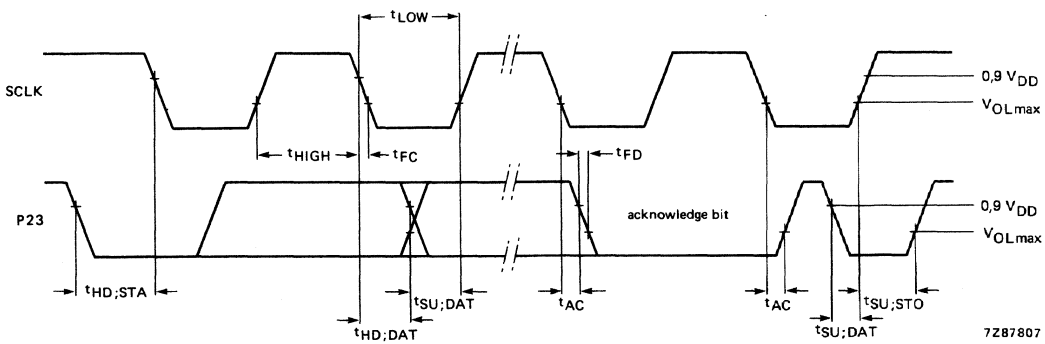



Fig.17 Timing requirements for the P2.3 and SCLK *output* signals.

**AC CHARACTERISTICS** (continued)

**Table 5** Output timing shown in Fig.17

| symbol | timing | |
|---|---|---|
| | normal mode (ASC in S2 = 0) | low-speed mode (ASC in S2 = 1) |
| $t_{HD;STA}$ | ½ (DF + 9) $t_{XTAL}$ | ¾ (DF + 9) $t_{XTAL}$ |
| $t_{HIGH}$ | ½ (DF) $t_{XTAL}$ | ¾ (DF) $t_{XTAL}$ |
| $t_{LOW}$ | ½ (DF) $t_{XTAL}$ | ¼ (DF) $t_{XTAL}$ |
| $t_{SU;STO}$ | ½ (DF − 3) $t_{XTAL}$ | ¼ (DF − 3) $t_{XTAL}$ |
| $t_{HD;DAT}$ (slave transmitter) any DF | $\geqslant 9t_{XTAL}$<br>$\leqslant 12t_{XTAL}$ | $\geqslant 9t_{XTAL}$<br>$\leqslant 12t_{XTAL}$ |
| $t_{HD;DAT}$ (master transmitter) for DF $\leqslant 51$ | $\geqslant 9t_{XTAL}$<br>$\leqslant 12t_{XTAL}$ | –<br>– |
| for DF $\leqslant 99$ | –<br>– | $\geqslant 9t_{XTAL}$<br>$\leqslant 12t_{XTAL}$ |
| $t_{SU;DAT}$ (master transmitter) for DF $> 51$ | $\geqslant 15t_{XTAL}$<br>$\leqslant 24t_{XTAL}$ | –<br>– |
| for DF $> 99$ | –<br>– | $\geqslant 15t_{XTAL}$<br>$\leqslant 24t_{XTAL}$ |
| $t_{AC}$ | $\geqslant 9t_{XTAL}$<br>$\leqslant 12t_{XTAL}$ | $\geqslant 9t_{XTAL}$<br>$\leqslant 12t_{XTAL}$ |
| $t_{FD}$, $t_{FC}$ | $\leqslant 100$ ns at $C_b = 400$ pF | $\leqslant 100$ ns at $C_b = 400$ pF |

**Notes to Table 5**

$t_{XTAL}$ = one period of the XTAL input frequency ($f_{XTAL}$)
= 167 ns for $f_{XTAL} = 6$ MHz
DF = divisor (see Table 2 Serial I/O section).
$C_b$ = the maximum bus capacitance for each line.

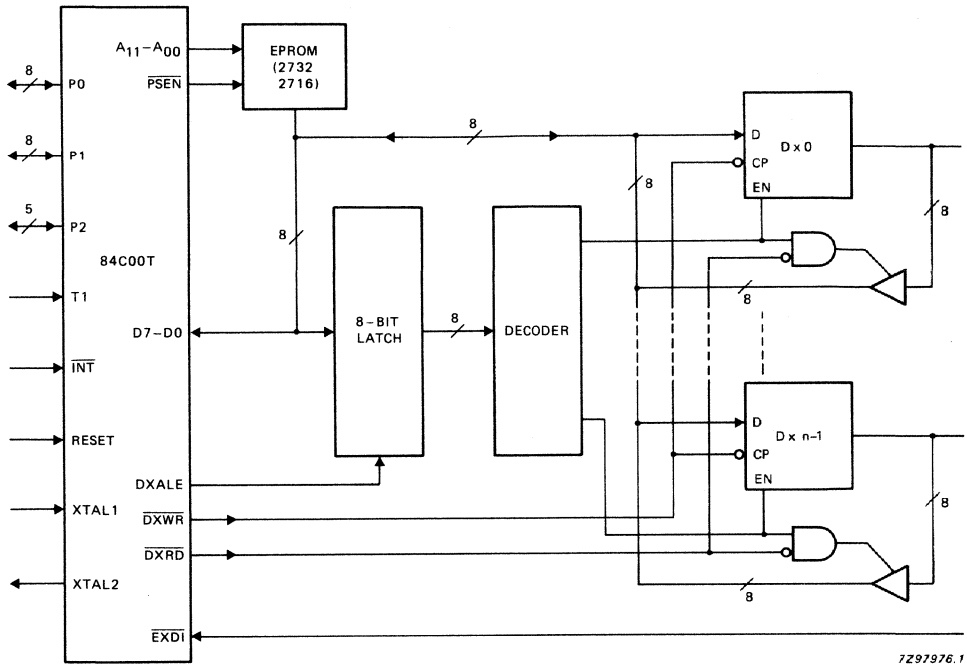



Fig. 18 Block diagram of the external Dx register interface. The Dx interface can only be used with the PCF84C00T.

# 8-Bit Microcontroller

# PCF84C12A, PCF84C22A, PCF84C42A

## FEATURES

- 8-bit CPU, ROM, RAM, I/O in a single 20-lead package
- 1k ROM bytes (PCF84C12A)
- 2k ROM bytes (PCF84C22A)
- 4k ROM bytes (PCF84C42A)
- 64 RAM bytes
- Over 100 instructions (based on MAB8048) all of 1 or 2 cycles
- 13 quasi-bidirectional I/O port lines
- 8-bit programmable timer/event counter 1
- Two single-level vectored interrupts: external, 8-bit programmable timer/event counter 1
- Two test inputs, one of which also serves as the external interrupt input
- Stop and idle modes
- Logic supply $V_{DD}$: 2.5 V to 5.5 V
- Clock frequency: 1 MHz to 16 MHz - Operating temperature range: –40 °C to 85 °C
- Manufactured in silicon gate CMOS process

## GENERAL DESCRIPTION

This data sheet details the specific properties of the PCF84C12A, PCF84C22A and PCF84C42A. The shared characteristics of the PCF84CXXXA family of microcontrollers are described in the PCF84CXXXA family data sheet, which should be read in conjunction with this publication.

The PCF84C12A, PCF84C22A and PCF84C42A are general purpose CMOS microcontrollers with 1k, 2k and 4k bytes of program memory, respectively. They include 64 bytes of RAM and 13 I/O port lines. The instruction set is based on that of the MAB8048 and is software compatible with the PCF84CXXXA family.

## ORDERING INFORMATION

| EXTENDED TYPE NUMBER | PACKAGE | | | |
|---|---|---|---|---|
| | PINS | PIN POSITION | MATERIAL | CODE |
| PCF84C12AP/22AP/42AP | 20 | DIL | plastic | SOT146 |
| PCF84C12AT/22AT/42AT | 20 | mini-pack | plastic | SOT163A |

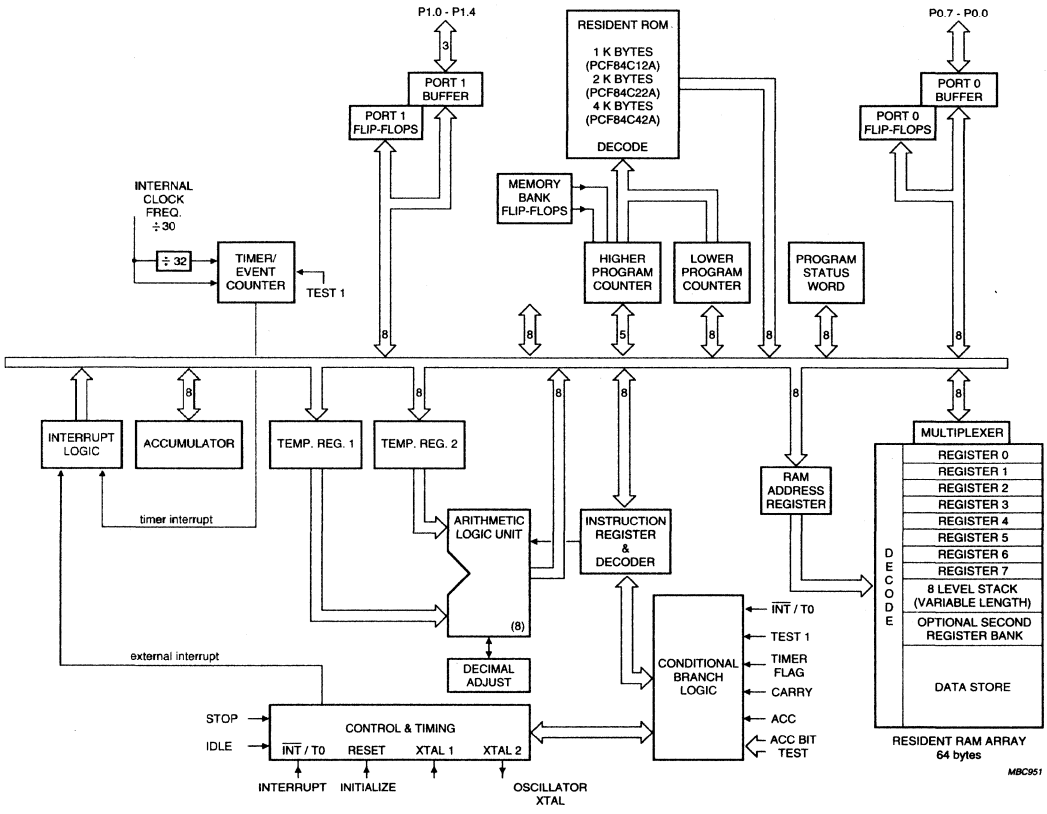



Fig.1 PCF84C12A/22A/42A Block diagram.

## PINNING

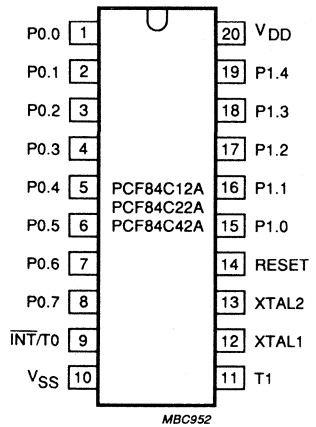



Fig.2 Pinning diagram.

**Table 1** Pin description

| SYMBOL | PIN | TYPE | FUNCTION |
|---|---|---|---|
| P0.0-P0.7 | 1-8 | I/O | Port 0: quasi-bidirectional I/O lines |
| INT/T0 | 9 | I | interrupt / test 0 |
| $V_{SS}$ | 10 | P | ground |
| T1 | 11 | I | test 1/count input of 8-bit timer/event counter 1 |
| XTAL1 | 12 | I | crystal oscillator / external clock |
| XTAL2 | 13 | O | crystal oscillator output |
| RESET | 14 | I | reset input |
| P1.0-P1.4 | 15-19 | I/O | Port 1:quasi-bidirectional I/O lines |
| $V_{DD}$ | 20 | P | positive supply |

## INSTRUCTION SET

Since serial I/O interface, port 2 and derivative logic are not provided, the serial input/output, the parallel input/output for port 2 and the derivative instructions are not available.

ROM space being restricted to 1k bytes (PCF84C12A), 2k bytes (PCF84C22A) and 4k bytes (PCF84C42A), SEL MB1/2/3 (for PCF84C12A and PCF84C22A) and SEL MB2/3 (for PCF84C42A) would define non-existing program memory banks and should therefore be avoided.

RAM space being restricted to 64 bytes, care should be taken to avoid accesses to non-existing RAM locations.

See PCF84CxxxA Family Specification for a complete description of the instruction set.

### Summary of Mask Options

**Table 2**

| ROM CODE | OPTION | | |
|---|---|---|---|
| program/data | any mix of instructions and data up to ROM size of 1k (PCF84C12A), 2k (PCF84C22A) and 4k bytes (PCF84C42A) | | |
| **Port Output Options** | | | |
| P0.0 - P0.7 | option 1 | option 2 | option 3 |
| P1.0 - P1.4 | option 1 | option 2 | option 3 |
| **Port State after reset** | | | |
| P0.0 - P0.7 | set | reset | - |
| P1.0 - P1.4 | set | reset | - |
| **Oscillator** | | | |

## HANDLING

| **Handling MOS devices** |
|---|
| Inputs and outputs are protected against electrostatic discharge in normal handling. However, it is good practice to take normal precautions appropriate to handling MOS devices. |

## LIMITING VALUES

In accordance with the Absolute Maximum System (IEC 134)

| SYMBOL | PARAMETER | MIN. | MAX. | UNIT |
|---|---|---|---|---|
| $V_{DD}$ | supply voltages | −0.5 | 7 | V |
| $V_I$ | all input voltages | −0.5 | $V_{DD}$ +0.5 | V |
| $I_I$,$I_O$ | DC input or output current | −10 | 10 | mA |
| $P_{tot}$ | total power dissipation | – | 125 | mW |
| $P_O$ | power dissipation per output | – | 30 | mW |
| $I_{DD}$ | positive supply current | −50 | 50 | mA |
| $I_{SS}$ | ground supply current | −100 | 50 | mA |
| $T_{stg}$ | storage temperature | −65 | 150 | °C |
| $t_j$ | operating junction temperature | - | 90 | °C |

## DC CHARACTERISTICS

$V_{DD}$ = 2.5 to 5.5 V; $V_{SS}$ = 0V; $T_{amb}$ = −40 °C to 85 °C; all voltages with respect to $V_{SS}$; unless otherwise specified

| SYMBOL | PARAMETER | CONDITIONS | MIN. | TYP. | MAX. | UNIT |
|---|---|---|---|---|---|---|
| $V_{DD}$ | supply voltage operating | Fig.3 | 2.5 | – | 5.5 | V |
| $I_{DD}$ | supply current operating | note 1, Figs 5 and 6<br>$V_{DD}$ = 3 V; $f_{XTAL}$ = 3.58 MHz ($g_{mL}$) | – | 0.3 | 0.6 | mA |
| | | $V_{DD}$ = 5 V; $f_{XTAL}$ = 10 MHz ($g_{mL}$) | – | 1.1 | 3.0 | mA |
| | | $V_{DD}$ = 5 V; $f_{XTAL}$ = 16 MHz ($g_{mM}$) | – | 1.7 | 5.0 | mA |
| | | $V_{DD}$ = 5 V; $f_{XTAL}$ = 16 MHz ($g_{mH}$) | – | 2.5 | 6.0 | mA |
| | idle mode | note 1, Figs 7 and 8<br>$V_{DD}$ = 3 V; $f_{XTAL}$ = 3.58 MHz ($g_{mL}$) | – | 0.2 | 0.4 | mA |
| | | $V_{DD}$ = 5 V; $f_{XTAL}$ = 10 MHz ($g_{mL}$) | – | 0.8 | 1.6 | mA |
| | | $V_{DD}$ = 5 V; $f_{XTAL}$ = 16 MHz ($g_{mM}$) | – | 1.2 | 4.0 | mA |
| | | $V_{DD}$ = 5 V; $f_{XTAL}$ = 16 MHz ($g_{mH}$) | – | 1.7 | 5.0 | mA |
| | stop mode | note 2, Fig.4<br>$V_{DD}$ = 2.5 V | – | 1.2 | 2.5 | µA |
| **Inputs** | | | | | | |
| $V_{IL}$ | LOW level input voltage | | 0 | – | 0.3 $V_{DD}$ | V |
| $V_{IH}$ | HIGH level input voltage | | 0.7 $V_{DD}$ | – | $V_{DD}$ | V |
| $I_{IL}$ | input leakage | ($V_{SS} \leq V_I \leq V_{DD}$) | −1 | – | 1 | µA |
| **Port outputs** | | | | | | |
| $I_O$ | Port sink current LOW | Fig.9 $V_{DD}$ = 5 V; $V_O$ = 0.4 V | −1.6 | −12 | – | mA |
| $I_{OH}$ | Port pullup source current HIGH | Fig.10 $V_{DD}$ = 5 V; $V_O$ = 3.5 V | 40 | 100 | – | µA |
| | | $V_{DD}$ = 5 V; $V_O$ = 0 V | – | 140 | 400 | µA |
| | Port push-pull source current HIGH | Fig.11 $V_{DD}$ = 5 V; $V_O$ = 4.6 V | 1.6 | 7 | – | mA |
| **Oscillator transconductance Fig.12** | | | | | | |
| $g_{mL}$ | option 'LOW' | $g_m$ at $V_{DD}$ = 5 V | 0.2 | 0.4 | 1.0 | mA/V |
| $g_{mM}$ | option 'MEDIUM' | $g_m$ at $V_{DD}$ = 5 V | 0.9 | 1.6 | 3.2 | mA/V |
| $g_{mH}$ | option 'HIGH' | $g_m$ at $V_{DD}$ = 5 V | 3.0 | 4.5 | 9.0 | mA/V |
| $R_f$ | feedback resistor | | 0.3 | 1.0 | 3.0 | MΩ |

### Notes

1. $V_{IL}$ = $V_{SS}$, $V_{IH}$ = $V_{DD}$; open drain outputs connected to $V_{SS}$; all other outputs, including XTAL2, open (Typical values at 25 °C with crystal connected between XTAL1 and XTAL2).
2. $V_{IL}$ = $V_{SS}$, $V_{IH}$ = $V_{DD}$; RESET and T1 at $V_{SS}$; INT/T0 at $V_{DD}$; crystal connected between XTAL1 and XTAL2; open drain outputs connected to $V_{SS}$; all other outputs open.

## AC CHARACTERISTICS

$V_{DD}$ = 2.5 to 5.5 V; $V_{SS}$ = 0 V; $T_{amb}$ = –40 °C to 85 °C; all voltages with respect to $V_{SS}$; unless otherwise specified.

| SYMBOL | PARAMETER | MIN. | TYP. | MAX. | UNIT |
|---|---|---|---|---|---|
| $t_R$ | rise time all outputs (note 1) | – | 30 | – | ns |
| $t_F$ | fall time all outputs (note 1) | – | 30 | – | ns |
| $f_{XTAL}$ | clock frequency Fig.3 | 1 | – | 16 | MHz |

**Note**

1. $V_{DD}$ = 5 V; $T_{AMB}$ = 25 °C; $C_L$ = 50 pF.

## CHARACTERISTIC CURVES

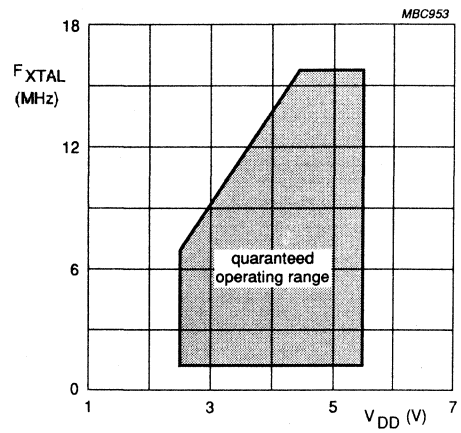



Fig.3 Maximum clock frequency as a function of supply voltage ($V_{DD}$).

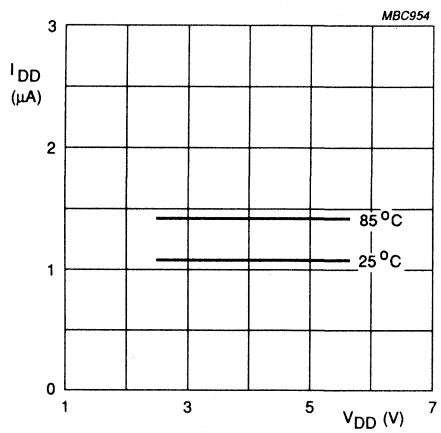
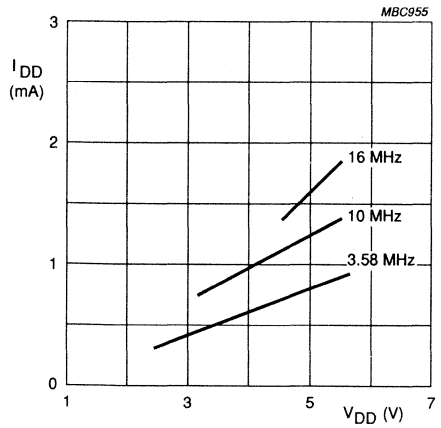
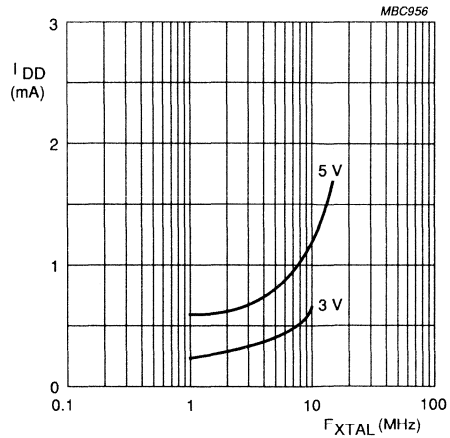



Fig.4 Typical supply current ($I_{DD}$) in stop mode as a function of supply voltage ($V_{DD}$).



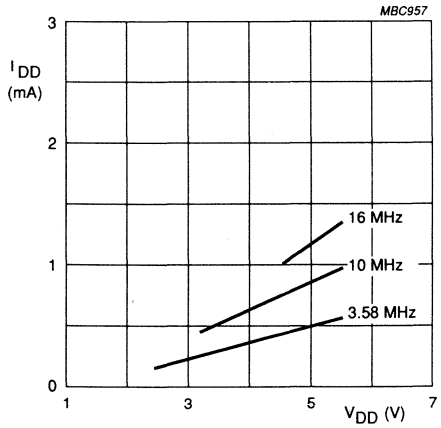

Supply current based on optimal oscillator selection. (See table 2)

Fig.5 Typical supply current ($I_{DD}$) in operation as a function of supply voltage ($V_{DD}$).



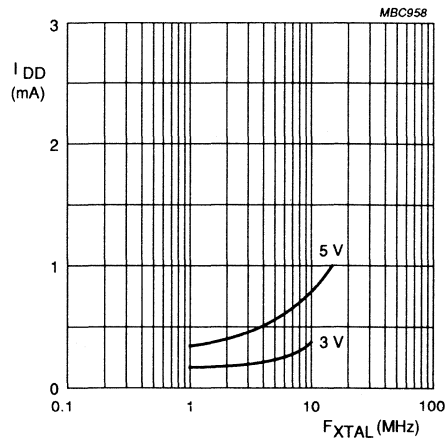

Supply current based on optimal oscillator selection. (See table 2)

Fig.6 Typical supply current ($I_{DD}$) in operation as a function of frequency ($f_{XTAL}$).




Supply current based on optimal oscillator selection. (See table 2)

Fig.7 Typical supply current ($I_{DD}$) in idle mode as a function of supply voltage ($V_{DD}$).




Supply current based on optimal oscillator selection. (See table 2)

Fig.8 Typical supply current ($I_{DD}$) in idle mode as a function of frequency ($f_{XTAL}$).

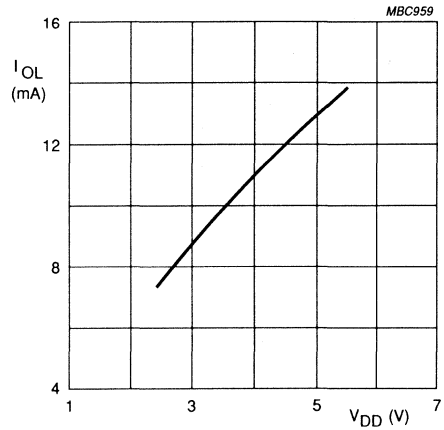



Fig.9 Typical port output sink current ($I_{OL}$) as a function of supply voltage ($V_{DD}$); $V_O$ = 0.4 V.

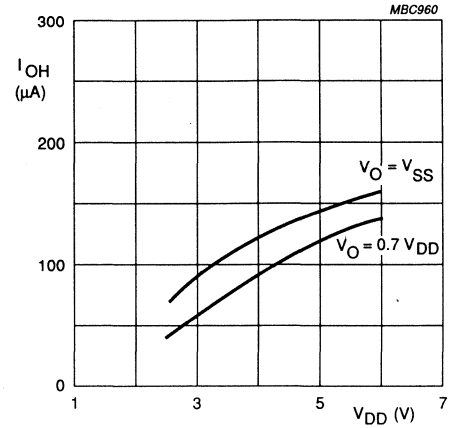



Fig.10 Typical output pullup source current ($I_{OH}$) as a function of supply voltage ($V_{DD}$).

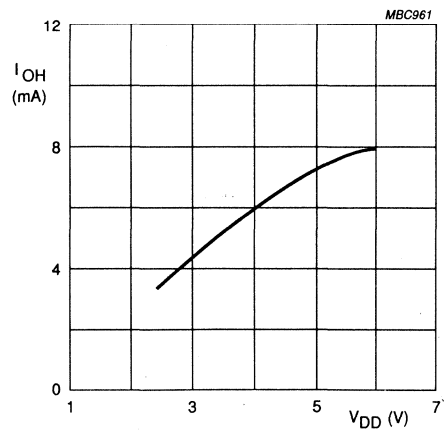



Fig.11 Typical push-pull output source current ($I_{OH}$) as a function of supply voltage ($V_{DD}$) for the options ($V_{DD}$); $V_O = V_{DD} - 0.4$ V.

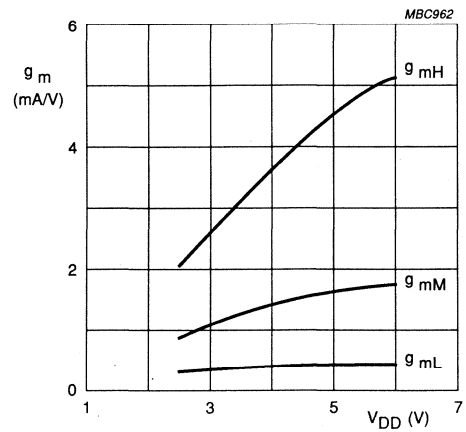



Fig.12 Typical transconductance values as a function of supply voltage ($V_{DD}$) for the options $g_{mL}$, $g_{mM}$ and $g_{mH}$.

# Telecom Microcontroller

# PCF84C21A, PCF84C41A, PCF84C81A

## FEATURES

- 8-bit CPU, ROM, RAM, I/O in a single 28-lead package
- 2k ROM bytes, 64 RAM bytes (PCF84C21A)
- 4k ROM bytes, 128 RAM bytes (PCF84C41A)
- 8k ROM bytes, 256 RAM bytes (PCF84C81A)
- Serial I/O interface with multi-master capability
- Over 100 instructions (based on MAB8048) all of 1 or 2 cycles
- 20 quasi-bidirectional I/O Port lines
- High sink current capability on the 8 lines of Port 1
- 8-bit programmable timer/event counter 1
- 3 single-level vectored interrupts: external, 8-bit programmable timer/event counter 1, SIO
- Two test inputs, one of which also serves as the external interrupt input
- Stop and Idle modes
- Positive supply $V_{DD}$: 2.5 V to 5.5 V
- Clock frequency: 1 MHz to 16 MHz - Operating temperature range: –40 °C to 85 °C
- Manufactured in silicon gate CMOS process

## GENERAL DESCRIPTION

This data sheet details the specific properties of the PCF84C21A, PCF84C41A and PCF84C81A. The shared characteristics of the PCF84CXXXA family of microcontrollers are described in the PCF84CXXXA family data sheet, which should be read in conjunction with this publication.

The PCF84C21A, PCF84C41A and PCF84C81A are general purpose CMOS microcontrollers with 2k, 4k and 8k bytes of program memory and 64, 128 and 256 bytes of RAM, respectively. In addition to 20 I/O port lines, the microcontrollers provide an on-chip serial I/O interface. This two-line serial bus extends the microcontroller capabilities when implemented with the powerful I²C bus devices of the PCF85XX, PCD33XX and "Clips" peripheral families. These include liquid crystal display drivers, telecom circuits, AD/DA converters, clock/calendar circuits, EEPROM and RAM. The instruction set is based on that of the MAB8048 and is software compatible with the PCF84CXXXA family.

## ORDERING INFORMATION

| EXTENDED TYPE NUMBER | PACKAGE | | | |
|---|---|---|---|---|
| | PINS | PIN POSITION | MATERIAL | CODE |
| PCF84C21AP | 28 | DIL | plastic | SOT117 |
| PCF84C41AP | 28 | DIL | plastic | SOT117 |
| PCF84C81AP | 28 | DIL | plastic | SOT117 |
| PCF84C21AT | 28 | mini-pack | plastic | SOT136A |
| PCF84C41AT | 28 | mini-pack | plastic | SOT136A |
| PCF84C81AT | 28 | mini-pack | plastic | SOT136A |

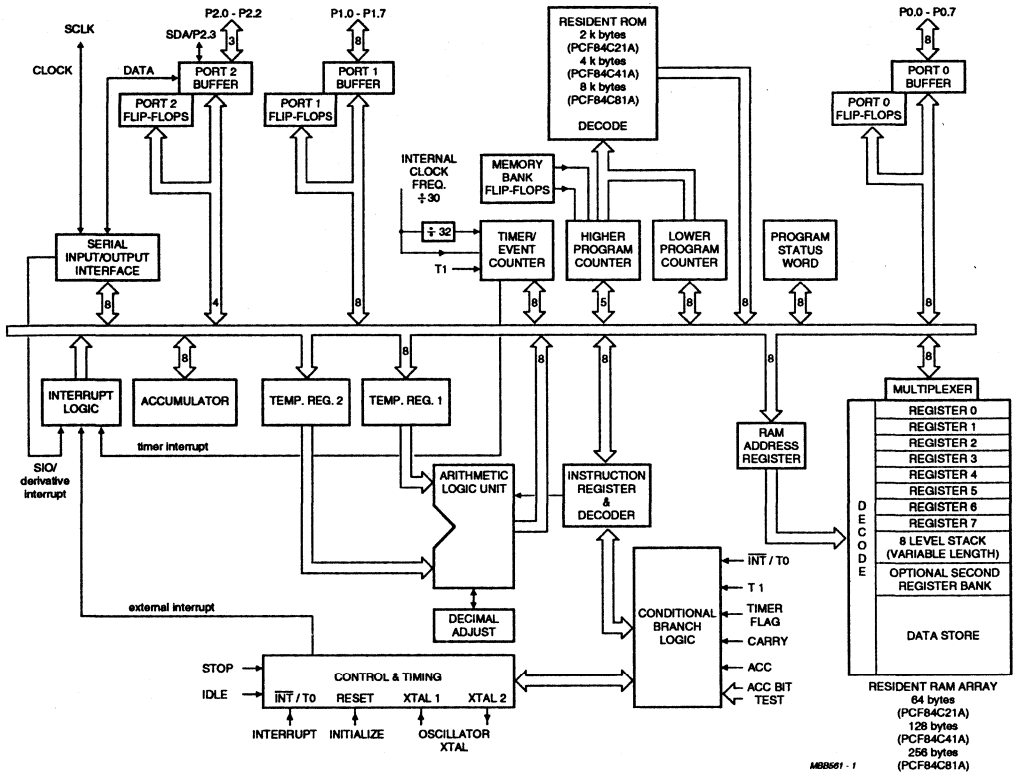



Fig.1 PCF84C21A/41A/81A block diagram.

## PINNING

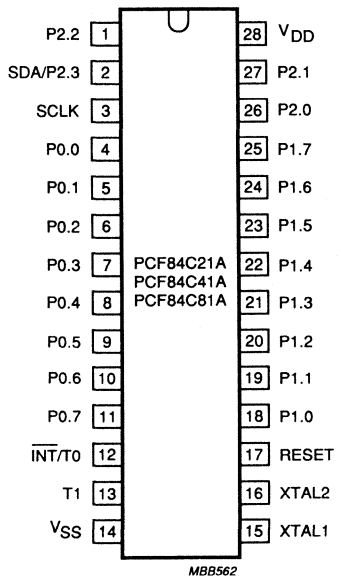



Fig.2 Pin configuration

## PIN DESCRIPTION

| SYMBOL | PIN | FUNCTION |
|---|---|---|
| P2.2 | 1 | Port 2: quasi-bidirectional I/O line |
| SDA/P2.3 | 2 | bidirectional data line of the serial I/O interface<br>Port 2: quasi-bidirectional I/O line |
| SCLK | 3 | bidirectional clock line of the serial I/O interface |
| P0.0-P0.7 | 4-11 | Port 0: quasi-bidirectional I/O lines |
| $\overline{INT}$/T0 | 12 | interrupt/test 0 |
| T1 | 13 | test 1/count input of 8-bit timer/event counter 1 |
| $V_{SS}$ | 14 | ground |
| XTAL1 | 15 | crystal oscillator/external clock |
| XTAL2 | 16 | crystal oscillator output |
| RESET | 17 | reset input |
| P1.0-P1.7 | 18-25 | Port 1: quasi-bidirectional I/O lines |
| P2.0-P2.1 | 26-27 | Port 2: quasi-bidirectional I/O lines |
| $V_{DD}$ | 28 | positive supply |

## INSTRUCTION SET

ROM space being restricted to 2k bytes (PCF84C21A) and 4k bytes (PCF84C41A), SEL MB1 (for PCF84C21A) and SEL MB2/3 (for both PCF84C21A and PCF84C41A) would define non-existing program memory banks and should therefore be avoided. RAM space being restricted to 64 bytes (PCF84C21A) and 128 bytes (PCF84C41A), care should be taken to avoid accesses to non-existing RAM locations.

See PCF84CXXXA Family Specification for a complete description of the instruction set.

### High sink current outputs

Port 1 outputs of PCF84C21A, PCF84C41A and PCF84C81A are designed for high current drive in the logic zero state. They are able to directly drive 10 mA loads and higher. Applications include drive for small relays and light-emitting diodes (LEDs). To avoid overload, care should be taken that the total Port 1 current averages less than 80 mA, i.e. an average of 10 mA per Port 1 line. Refer to table "Limiting Values", which specifies an upper limit of 100 mA for $|I_{SS}|$.

**Summary of Mask Options**

| ROM CODE | OPTION | | |
|---|---|---|---|
| program/data | any mix of instructions and data up to ROM size of 2k (PCF84C21A), 4k (PCF84C41A) and 8k bytes (PCF84C81A) | | |
| **Port Output Options** | | | |
| P0.0 – P0.7 | option 1 | option 2 | option 3 |
| P1.0 – P1.7 | option 1 | option 2 | option 3 |
| P2.0 – P2.2 | option 1 | option 2 | option 3 |
| SDA/P2.3 | – | option 2 | – |
| **Port State after reset** | | | |
| P0.0 – P0.7 | set | reset | |
| P1.0 – P1.7 | set | reset | |
| P2.0 – P2.2 | set | reset | |
| SDA/P2.3 | set | – | |
| **Oscillator** | | | |
| $g_m$ | $g_{mL}$ | $g_{mM}$ | $g_{mH}$ |

## HANDLING

| **Handling MOS devices** |
|---|
| Inputs and outputs are protected against electrostatic discharge in normal handling. However, it is good practice to take normal precautions appropriate to handling MOS devices. |

## LIMITING VALUES

In accordance with the Absolute Maximum Rating System (IEC 134)

| SYMBOL | PARAMETER | MIN. | MAX. | UNIT |
|---|---|---|---|---|
| $V_{DD}$ | supply voltages | –0.5 | +7 | V |
| $V_I$ | all input voltages | –0.5 | $V_{DD}$ +0.5 | V |
| $I_I$,$I_O$ | DC input or output current except Port 1 output LOW | –10 | +10 | mA |
| $I_O$ | Port 1 output LOW | –10 | +20 | mA |
| $P_{tot}$ | total power dissipation | – | 125 | mW |
| $P_O$ | power dissipation per output | – | 30 | mW |
| $I_{DD}$ | supply current ($V_{DD}$) | –50 | +50 | mA |
| $I_{SS}$ | ground supply current ($V_{SS}$) | –100 | +50 | mA |
| $T_{stg}$ | storage temperature range | –55 | 150 | °C |
| $T_j$ | operating junction temperature | – | 90 | °C |

## DC CHARACTERISTICS

$V_{DD}$ = 2.5 to 5.5 V; $V_{SS}$ = 0 V; $T_{amb}$ = –40 °C to 85 °C; all voltages with respect to $V_{SS}$; unless otherwise specified

| SYMBOL | PARAMETER | CONDITIONS | MIN. | TYP. | MAX. | UNIT |
|---|---|---|---|---|---|---|
| $V_{DD}$ | supply voltage operating | Fig.3 | 2.5 | – | 5.5 | V |
| $I_{DD}$ | supply current operating | note 1 and Figs 5 and 6; $V_{DD}$ = 3 V; $f_{XTAL}$ = 3.58 MHz ($g_{mL}$) | – | 0.3 | 0.6 | mA |
| | | $V_{DD}$ = 5 V; $f_{XTAL}$ = 10 MHz ($g_{mL}$) | – | 1.1 | 3.0 | mA |
| | | $V_{DD}$ = 5 V; $f_{XTAL}$ = 16 MHz ($g_{mM}$) | – | 1.7 | 5.0 | mA |
| | | $V_{DD}$ = 5 V; $f_{XTAL}$ = 16 MHz ($g_{mH}$) | – | 2.5 | 6.0 | mA |
| | Idle mode | note 1 and Fig.7 and 8; $V_{DD}$ = 3 V; $f_{XTAL}$ = 3.58 MHz ($g_{mL}$) | – | 0.2 | 0.4 | mA |
| | | $V_{DD}$ = 5 V; $f_{XTAL}$ = 10 MHz ($g_{mL}$) | – | 0.8 | 1.6 | mA |
| | | $V_{DD}$ = 5 V; $f_{XTAL}$ = 16 MHz ($g_{mM}$) | – | 1.2 | 4.0 | mA |
| | | $V_{DD}$ = 5 V; $f_{XTAL}$ = 16 MHz ($g_{mH}$) | – | 1.7 | 5.0 | mA |
| | Stop mode | note 2 and Fig.4; $V_{DD}$ = 2.5 V | – | 1.2 | 2.5 | µA |
| **Inputs** | | | | | | |
| $V_{IL}$ | LOW level input voltage | | 0 | – | 0.3 $V_{DD}$ | V |
| $V_{IH}$ | HIGH level input voltage | | 0.7 $V_{DD}$ | – | $V_{DD}$ | V |
| $I_{IL}$ | Input leakage current | ($V_{SS} \leq V_I \leq V_{DD}$) | –1 | – | +1 | µA |
| **Port outputs** | | | | | | |
| $I_{OL}$ | Port sink current LOW | Fig.9; $V_{DD}$ = 5 V; $V_O$ = 0.4 V; except Port 1, SDA/P2.3 and SCLK | 1.6 | 12 | – | mA |
| | Port 1 sink current LOW | Fig.10; $V_{DD}$ = 5 V; $V_O$ = 1.2 V | 10 | 30 | – | mA |
| | SIO sink current LOW | Fig.11; $V_{DD}$ = 5V; $V_O$ = 0.4V; on SDA/P2.3 and SCLK | 3.0 | 12 | – | mA |
| $I_{OH}$ | Port pull-up source current HIGH | Fig.12; $V_{DD}$ = 5 V; $V_O$ = 3.5 V | 40 | 100 | – | µA |
| | | $V_{DD}$ = 5V; $V_O$ = 0 V | – | 140 | 400 | µA |
| | Port push-pull source current HIGH | Fig.13; $V_{DD}$ = 5 V; $V_O$ = 4.6 V | 1.6 | 7 | – | mA |
| **Oscillator transconductance Fig.14** | | | | | | |
| $g_{mL}$ | option 'LOW' | $g_m$ at $V_{DD}$ = 5 V | 0.2 | 0.4 | 1.0 | mA/V |
| $g_{mM}$ | option 'MEDIUM' | $g_m$ at $V_{DD}$ = 5 V | 0.9 | 1.6 | 3.2 | mA/V |
| $g_{mH}$ | option 'HIGH' | $g_m$ at $V_{DD}$ = 5 V | 3.0 | 4.5 | 9.0 | mA/V |
| $R_f$ | feedback resistor | | 0.3 | 1.0 | 3.0 | MΩ |

### Notes

1. $V_{IL}$ = $V_{SS}$, $V_{IH}$ = $V_{DD}$; open drain outputs connected to $V_{SS}$; all other outputs open. Max. values: external clock at XTAL1; XTAL2 open. Typ. values: 25 °C; crystal connected between XTAL1 and XTAL2.
2. $V_{IL}$ = $V_{SS}$, $V_{IH}$ = $V_{DD}$; RESET and T1 at $V_{SS}$; $\overline{INT}$/T0 at $V_{DD}$; crystal connected between XTAL1 and XTAL2; open drain outputs connected to $V_{SS}$; all other outputs open.

## AC CHARACTERISTICS

$V_{DD}$ = 2.5 to 5.5V; $V_{SS}$ = 0 V; $T_{amb}$ = –40 °C to 85 °C; all voltages with respect to $V_{SS}$; unless otherwise specified

| SYMBOL | PARAMETER | CONDITIONS | MIN. | TYP. | MAX. | UNIT |
|---|---|---|---|---|---|---|
| $t_r$ | rise time all outputs | note 1 | – | 30 | – | ns |
| $t_f$ | fall time all outputs | note 1 | – | 30 | – | ns |
| $f_{XTAL}$ | clock frequency | see Fig. 3 | 1 | – | 16 | MHz |

**Note**

1. $V_{DD}$ = 5 V; $T_{amb}$ = 25 °C; $C_L$ = 50 pF.

## CHARACTERISTIC CURVES

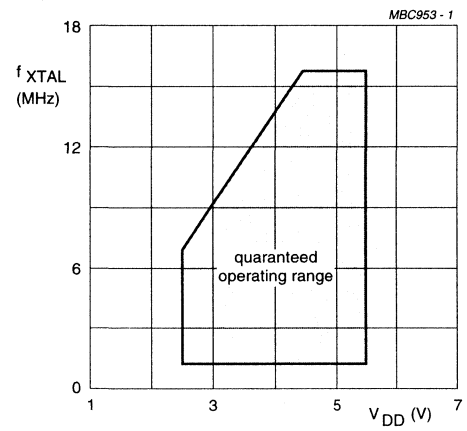



Fig.3 Maximum clock frequency as a function of supply voltage ($V_{DD}$).

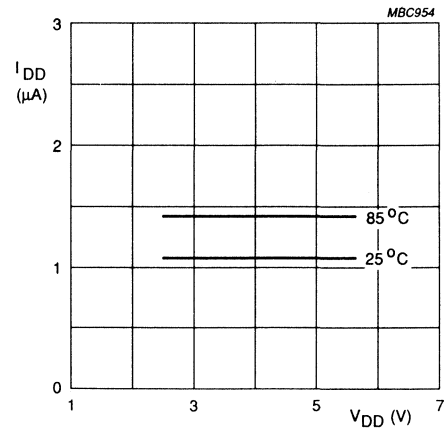



Fig.4 Typical supply current ($I_{DD}$) in stop mode as a function of supply voltage ($V_{DD}$).

**Table 1** SERIAL I/O INTERFACE CHARACTERISTICS

| SYMBOL | PARAMETER | SCLK INPUT (Fig. 15) | SCLK OUTPUT (Fig. 16, note 1) |
|---|---|---|---|
| $t_{HD;STA}$ | START condition hold time | $\geq 14/f_{XTAL}$ | $(DF + 9)/(2 \times f_{XTAL})$ |
| $t_{LOW}$ | SCLK LOW time | $\geq 17/f_{XTAL}$ | $(DF - 3)/(2 \times f_{XTAL})$; note 2 |
| $t_{HIGH}$ | SCLK HIGH time | $\geq 17/f_{XTAL}$ | $(DF + 3)/(2 \times f_{XTAL})$; note 2 |
| $t_r$ | SCLK rise time | $\leq 1$ µs | $\leq 1$ µs; note 3 |
| $t_f$ | SCLK fall time | $\leq 0.3$ µs | $\leq 0.1$ µs; note 4 |

**Notes**

1. DF stands for divisor of $f_{XTAL}$ (see PCF84CXXXA family data sheet).
2. Values given for ASC = 0; for ASC = 1: $t_{LOW} = (DF - 3)/(4 \times f_{XTAL})$, $t_{HIGH} = 3(DF + 1)/(4 \times f_{XTAL})$
3. Determined by I²C bus capacitance ($C_b$) and external pull-up resistor.
4. At maximum allowed I²C bus capacitance $C_b = 400$ pF.

**Table 2** SERIAL I/O INTERFACE CHARACTERISTICS

| SYMBOL | PARAMETER | SDA/P2.3 INPUT (Fig. 15) | SDA/P2.3 OUTPUT (Fig. 16, note 1) |
|---|---|---|---|
| $t_{BUF}$ | bus free time | $\geq 14/f_{XTAL}$ | $\geq 4.7$ µs; note 4 |
| $t_{SU;DAT}$ | DATA set-up time | $\geq 250$ ns | $\geq 15/f_{XTAL}$; note 5 |
| $t_{HD;DAT}$ | DATA hold time | $\geq 0$ | $\geq 9/f_{XTAL}$ |
| $t_{RD}$ | SDA/P2.3 rise time | $\leq 1$ µs | $\leq 1$ µs; note 2 |
| $t_{FD}$ | SDA/P2.3 fall time | $\leq 0.3$ µs | $\leq 0.1$ µs; note 3 |
| $t_{SU;STO}$ | STOP condition set-up time | $\geq 14/f_{XTAL}$ | $(DF - 3)/(2 \times f_{XTAL})$ |

**Notes**

1. DF stands for divisor of $f_{XTAL}$ (see PCF84CXXXA family data sheet).
2. Determined by I²C bus capacitance ($C_b$) and external pullup resistor.
3. At maximum allowed I²C bus capacitance $C_b = 400$ pF.
4. Determined by program.
5. If $t_{LOW} < 24/f_{XTAL}$, $t_{SU;DAT} > = t_{LOW} - 9/f_{XTAL}$, independently of ASC.

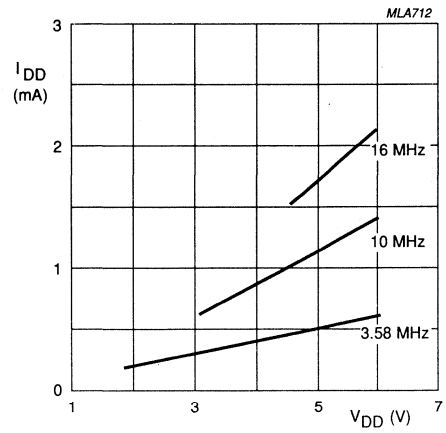



Fig. 5 Supply current based on optimal oscillator selection

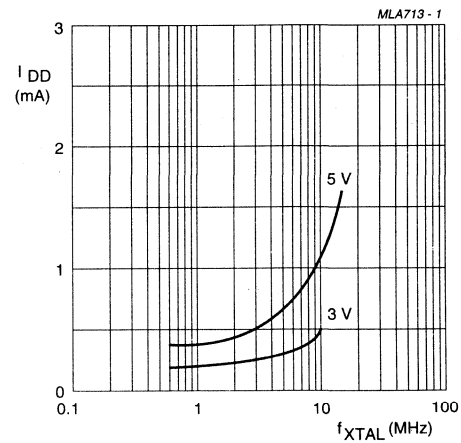



Fig. 6 Supply current based on optimal oscillator selection

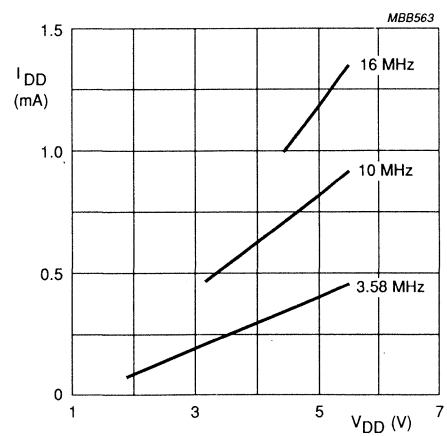



Fig. 7 Supply current based on optimal oscillator selection

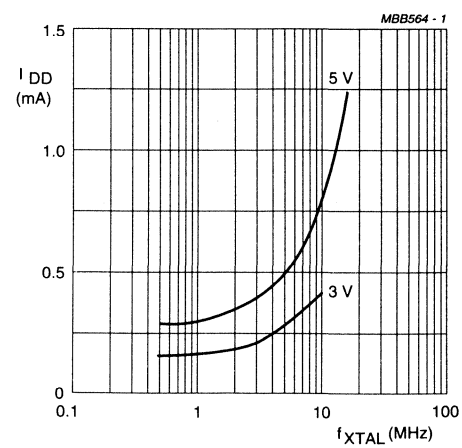



Fig. 8 Supply current based on optimal oscillator selection

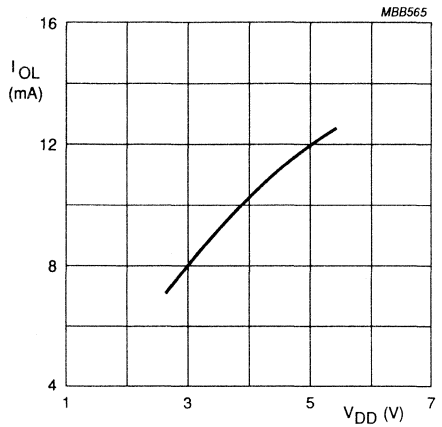



Fig.9 Typical Port output sink current ($I_{OL}$) as a function of supply voltage ($V_{DD}$); $V_O$ = 0.4 V.

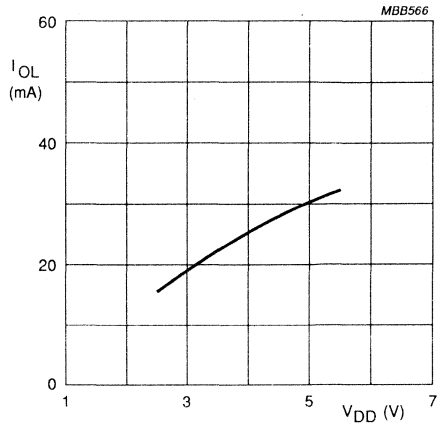



Fig.10 Typical Port output sink current ($I_{OL}$) as a function of supply voltage ($V_{DD}$); $V_O$ = 1.2 V.

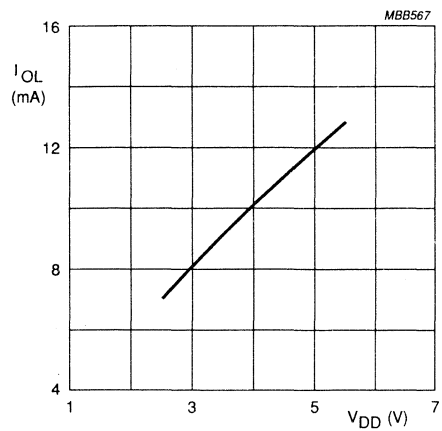



Fig.11 Typical SDA/P2.3 or SCLK output sink current ($I_{OL}$) as a function of supply voltage.

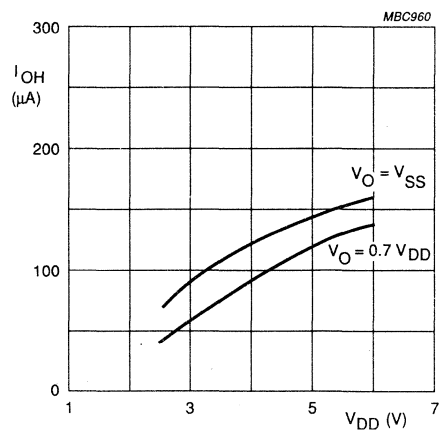



Fig.12 Typical output pullup source current ($I_{OH}$) as a function of supply voltage ($V_{DD}$).

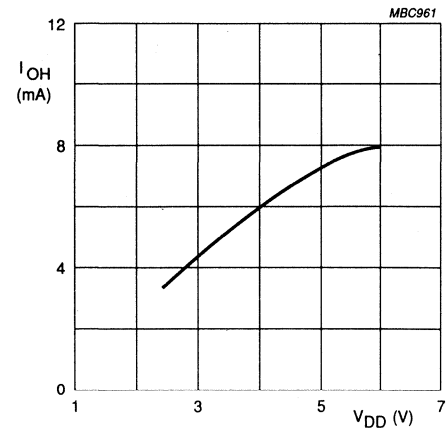



Fig.13 Typical push-pull output source current ($I_{OH}$) as a function of supply voltage ($V_{DD}$); $V_O = V_{DD} - 0.4$ V.

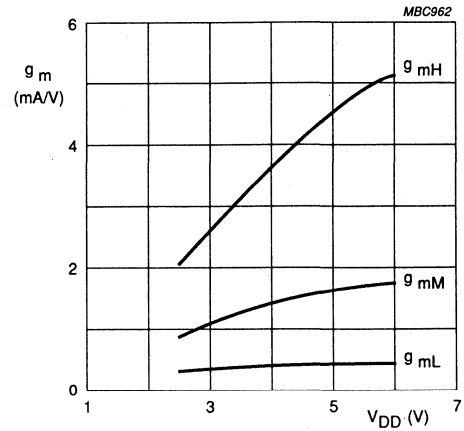



Fig.14 Typical transconductance values as a function of supply voltage ($V_{DD}$) for the options $g_{mL}$, $g_{mM}$ and $g_{mH}$.

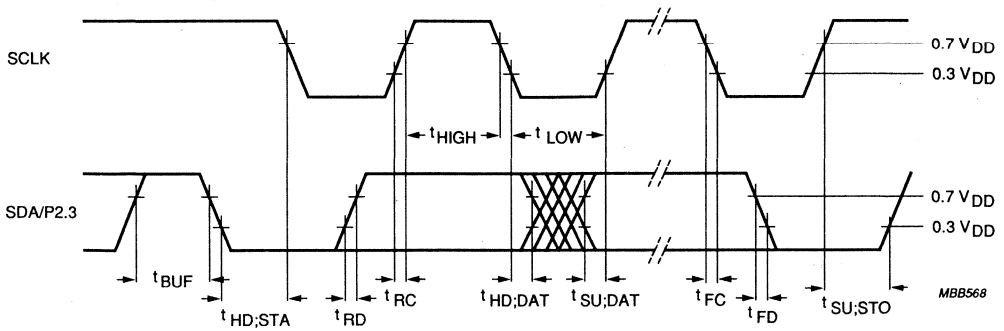



Fig.15 Slave SCLK receiver SDA/P2.3 timing (SCLK and SDA/P2.3 are inputs).

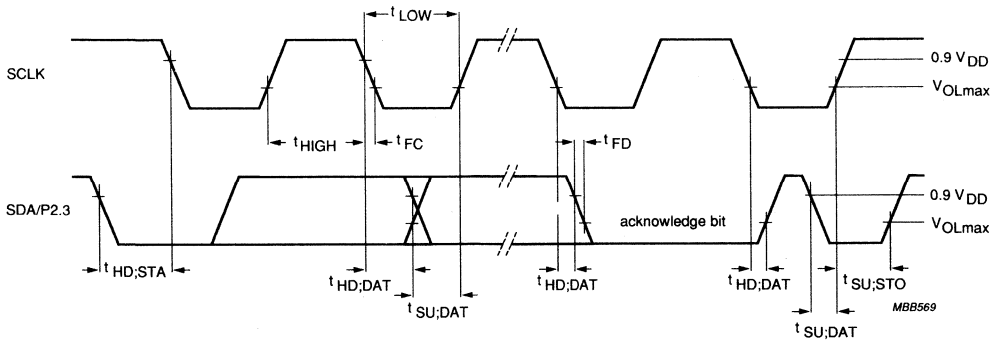



Fig.16 Master SCLK and transmitter SDA/P2.3 timing (SCLK and SDA/P2.3 are outputs).

**PURCHASE OF PHILIPS I²C COMPONENTS**

Purchase of Philips I²C components conveys a license under the Philips' I²C patent to use the components in the I²C system provided the system conforms to the I²C specification defined by Philips. This specification can be ordered using the code 9398 393 40011.

CONTENTS

# Microcontroller with extended I/O PCF84C85A

## 1 FEATURES

- 8-bit CPU, ROM, RAM, I/O in a single 40-lead package
- 8 kbytes ROM
- 256 bytes RAM
- Serial $I^2C$-bus interface with multi-master capability
- Over 100 instructions (based on MAB8048) all of 1 or 2 cycles
- 32 quasi-bidirectional I/O port lines
- 8-bit programmable timer/event counter 1
- 3 single-level vectored interrupts:
  - external
  - 8-bit programmable timer/event counter 1
  - SIO
- Two test inputs, one of which also serves as the external interrupt input
- Stop and Idle modes
- Logic supply voltage: $V_{DD}$ = 2.5 to 5.5 V
- Clock frequency: 1 to 16 MHz
- Operating temperature: –40 to +85 °C
- Manufactured in silicon gate CMOS process.

## 2 GENERAL DESCRIPTION

This data sheet details the specific properties of the PCF84C85A. The shared characteristics of the PCF84CXXXA family of microcontrollers are described in the *"PCD84XXXA family data sheet"* which should be read in conjunction with this publication.

The PCF84C85A is a general purpose CMOS microcontroller with emphasis on input/output. It provides 32 I/O port lines, 8 kbytes of program memory and 256 bytes of RAM. In addition to 20 I/O port lines, the microcontroller provides an on-chip serial I/O interface. This two-line serial bus extends the microcontroller's capabilities when implemented with the powerful $I^2C$-bus devices of the PCF85XX, PCD33XX and 'Clips' peripheral families.These include liquid crystal display drivers, telecom circuits, AD/DA converters, clock/calendar circuits, EEPROM and RAM.

The instruction set is based on that of the MAB8048 and is software compatible with the PCF84CXXXA family.

## 3 ORDERING INFORMATION

| TYPE NUMBER | PACKAGE | | | |
|---|---|---|---|---|
| | PINS | PIN POSITION | MATERIAL | CODE |
| PCF84C85AP | 40 | DIP | plastic | SOT129-1 |
| PCF84C85AT | 40 | VSO40 | plastic | SOT158-1 |

## 4 BLOCK DIAGRAM

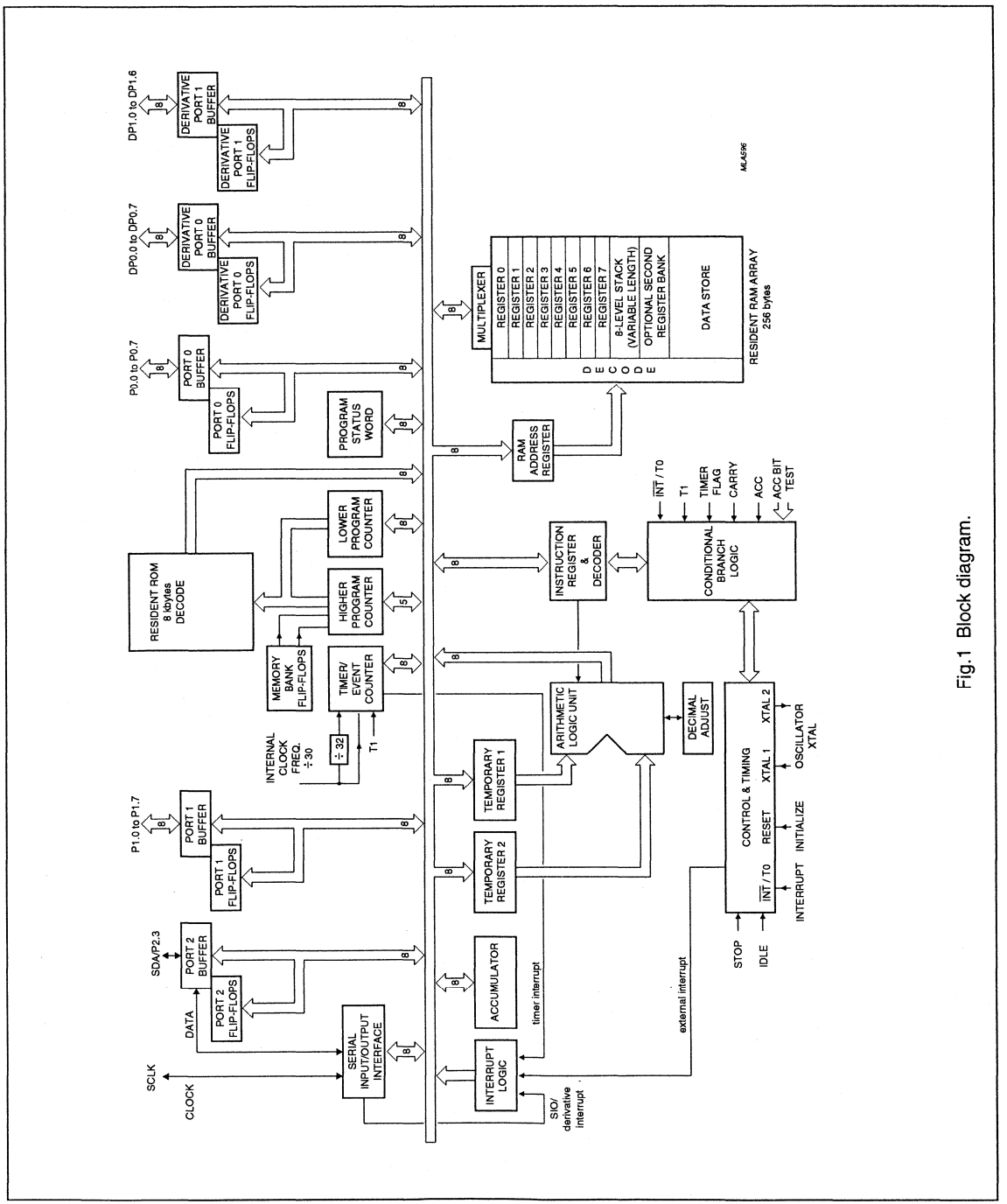



Fig.1 Block diagram.

## 5 PINNING INFORMATION

### 5.1 Pinning

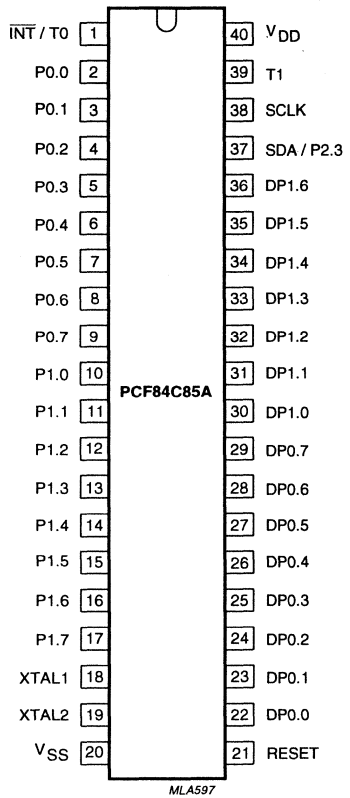



Fig.2 Pin configuration.

### 5.2 Pin description

**Table 1** DIP40 and VSO40 packages.

| SYMBOL | PIN | TYPE | DESCRIPTION |
|---|---|---|---|
| $\overline{INT}$/T0 | 1 | I | Interrupt/Test 0 |
| P0.0 to P0.7 | 2 to 9 | I/O | Port 0: quasi-bidirectional I/O lines |
| P1.0 to P1.7 | 10 to 17 | I/O | Port 1: quasi-bidirectional I/O lines |
| XTAL1 | 18 | I | XTAL input: crystal oscillator/external clock input |
| XTAL2 | 19 | O | XTAL output: crystal oscillator output |
| $V_{SS}$ | 20 | P | ground |
| RESET | 21 | I | Reset input |
| DP0.0 to DP0.7 | 22 to 29 | I/O | Derivative Port 0: quasi-bidirectional I/O lines |
| DP1.0 to DP1.6 | 30 to 36 | I/O | Derivative Port 1: quasi-bidirectional I/O lines |
| SDA/P2.3 | 37 | I/O | Port 2: bidirectional data line of the serial I/O interface/ quasi-bidirectional I/O line |
| SCLK | 38 | I/O | bidirectional clock line of the serial I/O interface |
| T1 | 39 | I | Test 1: count input of 8-bit timer/event counter 1 |
| $V_{DD}$ | 40 | P | positive supply |

## 6 PARALLEL PORTS

Of the standard quasi-bidirectional I/O ports, Port 2 is incomplete, providing only line SDA/P2.3 that is shared with the serial I/O interface. In addition to the standard ports, two derivative I/O ports are available:

- Derivative Port of 8 lines (DP0.0 to DP0.7)
- Derivative Port of 7 lines (DP1.0 to DP1.6).

Missing bits of incomplete ports, i.e. P2.0 to P2.2 and DP1.7, are fixed at zero.

## 7 INSTRUCTION SET

See *"PCF84CXXXA family data sheet"* for a complete description of the instruction set.

## 8 SUMMARY OF DERIVATIVE ADDRESSES AND REGISTERS

**Table 2** Derivative Addresses.

| Dx ADDRESS | TYPE | MNEMONIC | DESCRIPTION |
|---|---|---|---|
| 00H | R | DP0I | Derivative Port 0 lines |
| 01H | R | DP1I | Derivative Port 1 lines |
| 02H | R/W | DP0FF | Derivative Port 0 flip-flops |
| 03H | R/W | DP1FF | Derivative Port 1 flip-flops |
| 04H | – | – | |

**Table 3** Derivative Registers.

| REGISTER | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| DP0I | D0.7 | D0.6 | D0.5 | D0.4 | D0.3 | D0.2 | D0.1 | D0.0 |
| DP1I | 0 | D1.6 | D1.5 | D1.4 | D1.3 | D1.2 | D1.1 | D1.0 |
| DP0FF | F0.7 | F0.6 | F0.5 | F0.4 | F0.3 | F0.2 | F0.1 | F0.0 |
| DP1FF | 0 | F1.6 | F1.5 | F1.4 | F1.3 | F1.2 | F1.1 | F1.0 |

## 9 SUMMARY OF MASK OPTIONS

**Table 4** Port mask options (see *"PCF84CXXXA family data sheet"*).

| PORT | PORT OUTPUT[1] | | | PORT STATE AFTER RESET | |
|---|---|---|---|---|---|
| | OPTION 1 | OPTION 2 | OPTION 3 | SET | RESET |
| P0.0 to P0.7 | X | X | X | X | X |
| P1.0 to P1.7 | X | X | X | X | X |
| SDA/P2.3 | – | X | – | X | – |
| DP0.0 to DP0.7 | X | X | X | X | X |
| DP1.0 to DP1.6 | X | X | X | X | X |

**Note**

1. Option 1: normal port
   Option 2: open drain
   Option 3: push-pull.

**Table 5** Mask options.

| FEATURE | DESCRIPTION |
|---|---|
| ROM Code: program/data | Any mix of instructions and data up to ROM size of 8 kbytes. |
| Oscillator transconductance: $g_m$ | LOW transconductance: $g_{mL}$ |
| | MEDIUM transconductance: $g_{mM}$ |
| | HIGH transconductance: $g_{mH}$ |

## 10 LIMITING VALUES

In accordance with the Absolute Maximum Rating System (IEC 134).

| SYMBOL | PARAMETER | MIN. | MAX. | UNIT |
|---|---|---|---|---|
| $V_{DD}$ | supply voltage | –0.5 | +7 | V |
| $V_I$ | all input voltages | –0.5 | $V_{DD}$ + 0.5 | V |
| $I_I$ | DC input current | –10 | +10 | mA |
| $I_O$ | DC output current | –10 | +10 | mA |
| $P_{tot}$ | total power dissipation | – | 125 | mW |
| $P_O$ | power dissipation per output | – | 30 | mW |
| $I_{SS}$ | ground supply current ($V_{SS}$) | –50 | +50 | mA |
| $T_{stg}$ | storage temperature | –65 | +150 | °C |
| $T_j$ | operating junction temperature | – | 90 | °C |

## 11 HANDLING

Inputs and outputs are protected against electrostatic discharge in normal handling. However, it is good practice to take normal precautions appropriate to handling MOS devices (see *"Handling MOS devices"*).

## 12 DC CHARACTERISTICS

$V_{DD}$ = 2.5 to 5.5 V; $V_{SS}$ = 0 V; $T_{amb}$ = −40 to +85 °C; all voltages with respect to $V_{SS}$; unless otherwise specified.

| SYMBOL | PARAMETER | CONDITIONS | MIN. | TYP. | MAX. | UNIT |
|---|---|---|---|---|---|---|
| **Supply (see Figs 3 to 8)** | | | | | | |
| $V_{DD}$ | operating supply voltage | note 1 | 2.5 | – | 5.5 | V |
| $I_{DD}$ | operating supply current | $V_{DD}$ = 3 V; $f_{xtal}$ = 3.58 MHz ($g_{mL}$); note 1 | – | 0.3 | 0.6 | mA |
| | | $V_{DD}$ = 5 V; $f_{xtal}$ = 10 MHz ($g_{mL}$); note 1 | – | 1.1 | 3.0 | mA |
| | | $V_{DD}$ = 5 V; $f_{xtal}$ = 16 MHz ($g_{mM}$); note 1 | – | 1.7 | 5.0 | mA |
| | | $V_{DD}$ = 5 V; $f_{xtal}$ = 16 MHz ($g_{mH}$); note 1 | – | 2.5 | 6.0 | mA |
| $I_{DD(ID)}$ | supply current Idle mode | $V_{DD}$ = 3 V; $f_{xtal}$ = 3.58 MHz ($g_{mL}$); note 1 | – | 0.2 | 0.4 | mA |
| | | $V_{DD}$ = 5 V; $f_{xtal}$ = 10 MHz ($g_{mL}$); note 1 | – | 0.8 | 1.6 | mA |
| | | $V_{DD}$ = 5 V; $f_{xtal}$ = 16 MHz ($g_{mM}$); note 1 | – | 1.2 | 4.0 | mA |
| | | $V_{DD}$ = 5 V; $f_{xtal}$ = 16 MHz ($g_{mH}$); note 1 | – | 1.7 | 5.0 | mA |
| $I_{DD(ST)}$ | supply current Stop mode | $V_{DD}$ = 2.5 V; notes 1 and 2. | – | 1.2 | 10 | µA |
| **Inputs** | | | | | | |
| $V_{IL}$ | LOW level input voltage | | 0 | – | $0.3V_{DD}$ | V |
| $V_{IH}$ | HIGH level input voltage | | $0.7V_{DD}$ | – | $V_{DD}$ | V |
| $I_{LI}$ | input leakage current | $V_{SS} \leq V_I \leq V_{DD}$ | −1 | – | +1 | µA |
| **Outputs (see Figs 9 to 12)** | | | | | | |
| $I_{OL}$ | LOW level port sink current except SDA/P2.3 and SCLK | $V_{DD}$ = 5 V; $V_O$ = 0.4 V | 1.6 | 12 | – | mA |
| $I_{OL}$ | LOW level SIO sink current SDA/P2.3 and SCLK | $V_{DD}$ = 5 V; $V_O$ = 0.4 V | 3.0 | 12 | – | mA |
| $I_{OH}$ | HIGH level port pull-up source current | $V_{DD}$ = 5 V; $V_O$ = 3.5 V | −40 | −100 | – | µA |
| | | $V_{DD}$ = 5 V; $V_O$ = 0 V | – | −140 | −400 | µA |
| $I_{OH}$ | HIGH level port push-pull source current | $V_{DD}$ = 5 V; $V_O$ = 4.6 V | −1.6 | −7 | – | mA |

**Notes**

1. $V_{IL}$ = 0 V; $V_{IH}$ = $V_{DD}$; open drain outputs connected to $V_{SS}$; all other outputs open.
2. Crystal connected between XTAL1 and XTAL2; pin T1 at $V_{SS}$; pin $\overline{INT}$/T0 at $V_{DD}$.

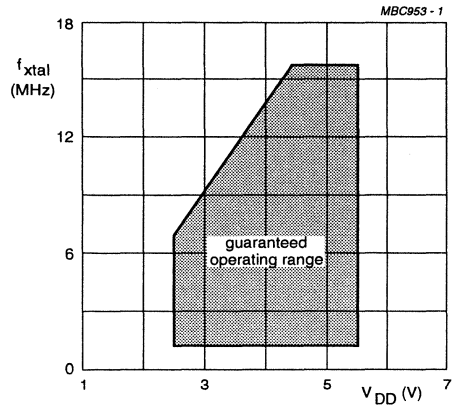



Fig.3 Maximum clock frequency ($f_{xtal}$) as a function of supply voltage ($V_{DD}$).

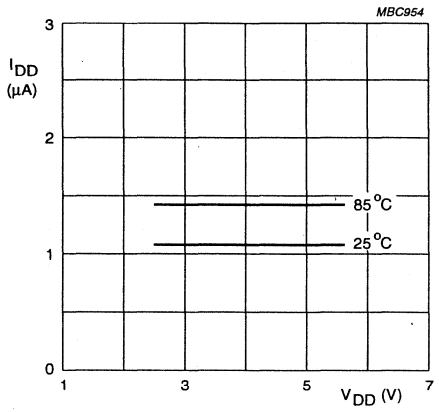



Fig.4 Typical supply current ($I_{DD}$) in Stop mode as a function of supply voltage ($V_{DD}$).

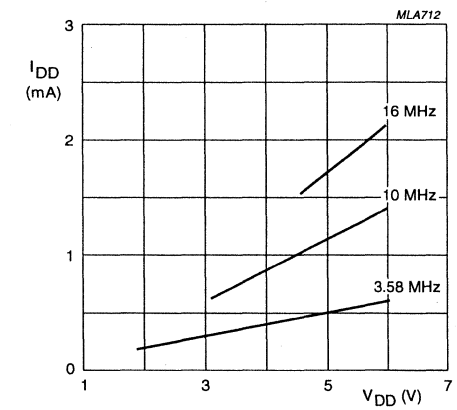



Fig.5 Typical operating supply current ($I_{DD}$) as a function of supply voltage ($V_{DD}$).

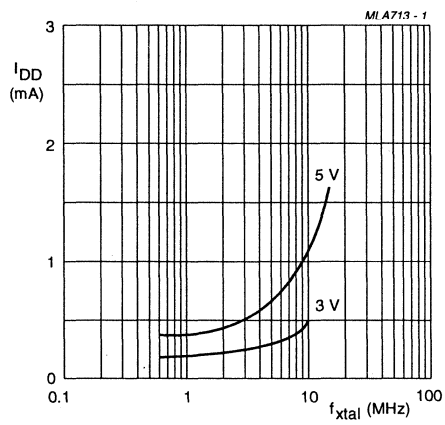



Fig.6 Typical operating supply current ($I_{DD}$) as a function of clock frequency ($f_{xtal}$).

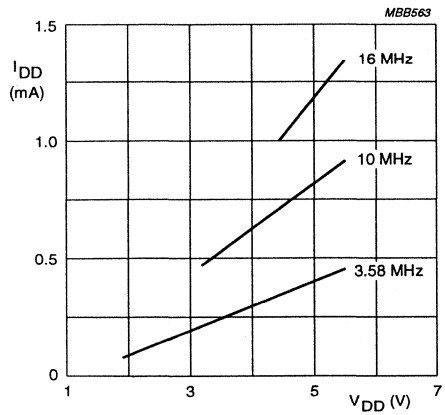



Fig.7 Typical supply current ($I_{DD}$) in Idle mode as a function of supply voltage ($V_{DD}$).

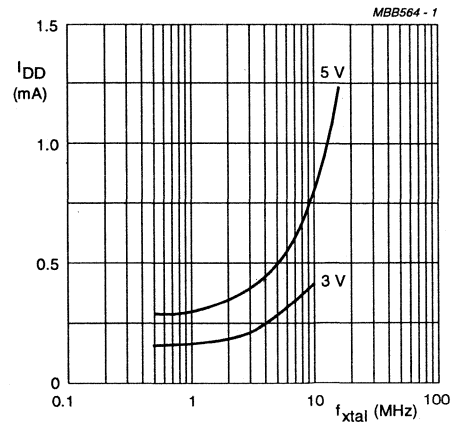
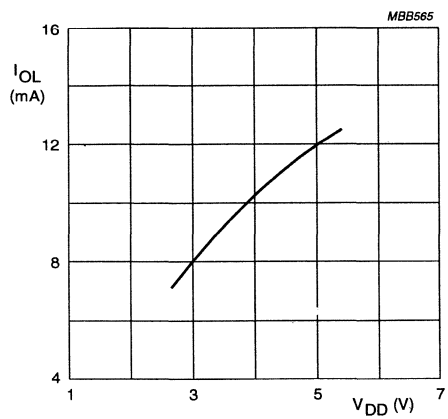
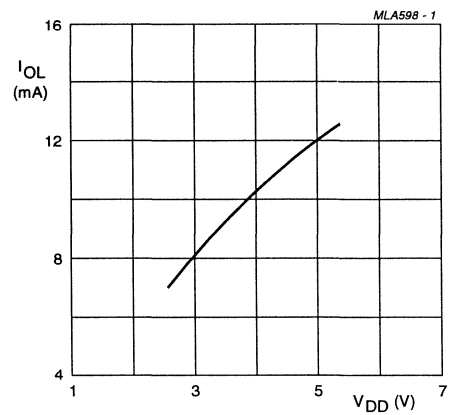



Fig.8 Typical supply current ($I_{DD}$) in Idle mode as a function of clock frequency ($f_{xtal}$).




$V_O$ = 0.4 V.

Fig.9 Typical LOW level port output sink current ($I_{OL}$) as a function of supply voltage ($V_{DD}$).




$V_O$ = 0.4 V.

Fig.10 Typical SDA/P2.3 or SCLK LOW level output sink current ($I_{OL}$) as a function of supply voltage ($V_{DD}$).

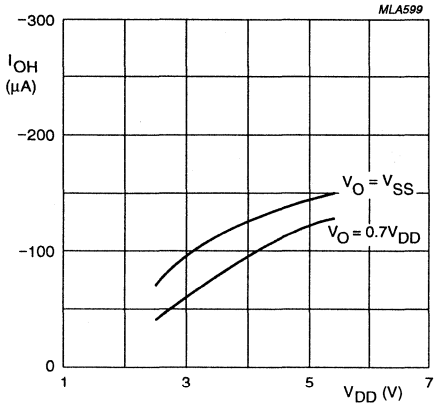
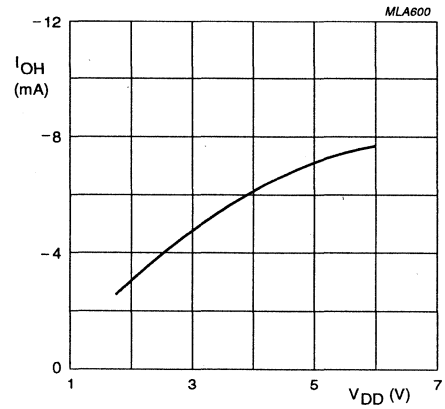



Fig.11 Typical HIGH level output pull-up source current ($I_{OH}$) as a function of supply voltage ($V_{DD}$).




$V_O = V_{DD} - 0.4$ V.

Fig.12 Typical HIGH level push-pull output source current ($I_{OH}$) as a function of supply voltage ($V_{DD}$).

## 13 AC CHARACTERISTICS

$V_{DD}$ = 2.5 to 5.5 V; $V_{SS}$ = 0 V; $T_{amb}$ = −40 to +85 °C; all voltages with respect to $V_{SS}$; unless otherwise specified.

| SYMBOL | PARAMETER | CONDITIONS | MIN. | TYP. | MAX. | UNIT |
|---|---|---|---|---|---|---|
| $t_r$ | rise time all outputs | $V_{DD}$ = 5 V; $T_{amb}$ = 25 °C; $C_L$ = 50 pF | – | 30 | – | ns |
| $t_f$ | fall time all outputs | $V_{DD}$ = 5 V; $T_{amb}$ = 25 °C; $C_L$ = 50 pF | – | 30 | – | ns |
| $f_{xtal}$ | clock frequency | see Fig.3 | 1 | – | 16 | MHz |
| **Oscillator (see Fig.13)** | | | | | | |
| $g_{mL}$ | LOW transconductance | $V_{DD}$ = 5 V | 0.2 | 0.4 | 1.0 | mS |
| $g_{mM}$ | MEDIUM transconductance | $V_{DD}$ = 5 V | 0.9 | 1.6 | 3.2 | mS |
| $g_{mH}$ | HIGH transconductance | $V_{DD}$ = 5 V | 3.0 | 4.5 | 9.0 | mS |
| $R_F$ | feedback resistor | | 0.3 | 1.0 | 3.0 | MΩ |

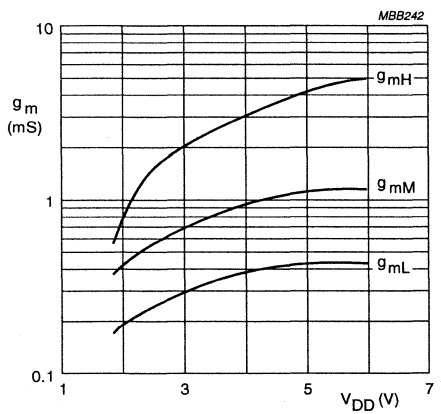



Fig.13 Typical transconductance as a function of supply voltage ($V_{DD}$) for the options $g_{mL}$, $g_{mM}$ and $g_{mH}$.

### 13.1 I²C-bus interface characteristics

**Table 6** I²C-bus timing.

| SYMBOL | PARAMETER | INPUT (see Fig.14) | OUTPUT (see Fig.15; note 1) |
|---|---|---|---|
| **SCLK** | | | |
| $t_{HD;STA}$ | START condition hold time | $\geq \frac{14}{f_{xtal}}$ | $\frac{DF+9}{2 \times f_{xtal}}$ |
| $t_{LOW}$ | SCLK LOW time | $\geq \frac{17}{f_{xtal}}$ | $\frac{DF-3}{2 \times f_{xtal}}$ ; note 2 |
| $t_{HIGH}$ | SCLK HIGH time | $\geq \frac{17}{f_{xtal}}$ | $\frac{DF+3}{2 \times f_{xtal}}$ ; note 2 |
| $t_{RC}$ | SCLK rise time | ≤ 1 µs | ≤ 1 µs; note 3 |
| $t_{FC}$ | SCLK fall time | ≤ 0.3 µs | ≤ 0.1 µs; note 4 |
| **SDA** | | | |
| $t_{BUF}$ | bus free time | $\geq \frac{14}{f_{xtal}}$ | ≥ 4.7 µs; note 5 |
| $t_{SU;DAT}$ | data set-up time | ≥ 250 ns | $\geq \frac{15}{f_{xtal}}$ ; note 6 |
| $t_{HD;DAT}$ | data hold time | ≥ 0 | $\geq \frac{9}{f_{xtal}}$ |
| $t_{RD}$ | SDA/P2.3 rise time | ≤ 1 µs | ≤ 1 µs; note 3 |
| $t_{FD}$ | SDA/P2.3 fall time | ≤ 0.3 µs | ≤ 0.1 µs; note 4 |
| $t_{SU;STO}$ | STOP condition set-up time | $\geq \frac{14}{f_{xtal}}$ | $\frac{DF-3}{2 \times f_{xtal}}$ |

**Notes**

1. DF stands for divisor of $f_{xtal}$ (see *"PCD33XXA family data sheet"*).
2. Values given for ASC = 0; for ASC =1: $t_{HIGH} = \frac{3(DF+1)}{4 \times f_{xtal}}$ ; $t_{LOW} = \frac{DF-3}{4 \times f_{xtal}}$.
3. Determined by I²C-bus capacitance ($C_b$) and external pull-up resistor.
4. At maximum allowed I²C-bus capacitance $C_b$ = 400 pF.
5. Determined by program.
6. If $t_{LOW} < \frac{24}{f_{xtal}}$, $t_{SU:DAT} \geq \frac{t_{LOW} - 9}{f_{xtal}}$, independent of ASC.

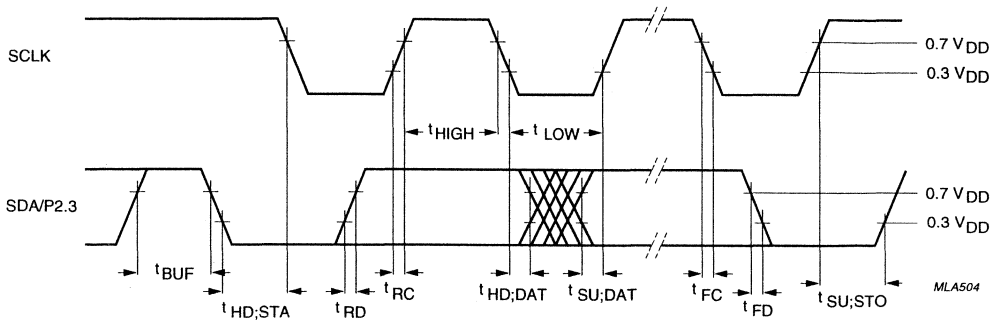



Fig.14 Slave SCLK and receiver SDA/P2.3 timing (SCLK and SDA/P2.3 are inputs).

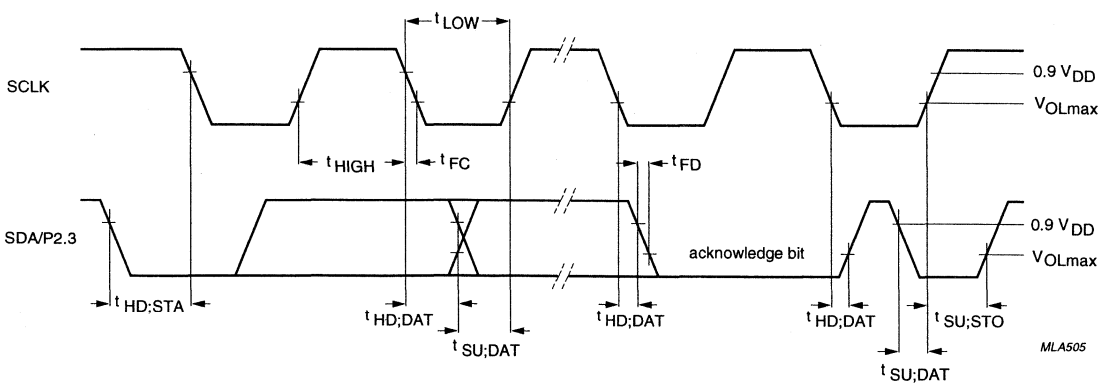



Fig.15 Master SCLK and transmitter SDA/P2.3 timing (SCLK and SDA/P2.3 are outputs).

# SINGLE-CHIP 8-BIT MICROCONTROLLER WITH 8 BYTES EEPROM

## DESCRIPTION

An advanced CMOS process is used to manufacture the PCF84C121. The PCF84C121 has 13 quasi-bidirectional I/O port lines, three single-level vectored interrupts (one external and two timer), 8 bytes of EEPROM, two 8-bit timer counters and on-chip clock oscillator and clock circuits.

This efficient microcontroller also performs well as an arithmetic processor. The PCF84C121 is pin- and instruction set compatible with the PCF84C12. The PCF84C121 has bit handling abilities and facilities for both binary and BCD arithmetic.

This microcontroller is a member of the 84CXXX family. For detailed information, consult the 84CXXX family specification.

### Features

- 8-bit CPU, ROM, RAM, EEPROM, I/O in a single 20-lead DIL or SO package
- 1 K x 8 ROM
- 64 x 8 RAM
- 8 x 8 EEPROM, designed for 10000 erase/write cycles per byte minimum
- 2 timers (8-bit programmable)
- 13 quasi-bidirectional I/O port lines
- 3 single-level, vectored interrupts: external, Timer 1 and Timer 2
- Two test inputs: one of which is also the external interrupt input
- 8-bit programmable timer/event counter
- Clock frequency range: 450 kHz to 10 MHz
- Over 80 instructions (similar to those of the MAB8048) all of 1 or 2 cycles
- Single supply voltage
- STOP and IDLE modes
- Power-on-reset circuit
- Operating temperature range: −40 to +85 °C; 0 to +55 °C for programming

## PACKAGE OUTLINES

PCF84C121P: 20-lead DIL; plastic (SOT146).
PCF84C121T: 20-lead mini-pack; plastic (SO20; SOT163A).

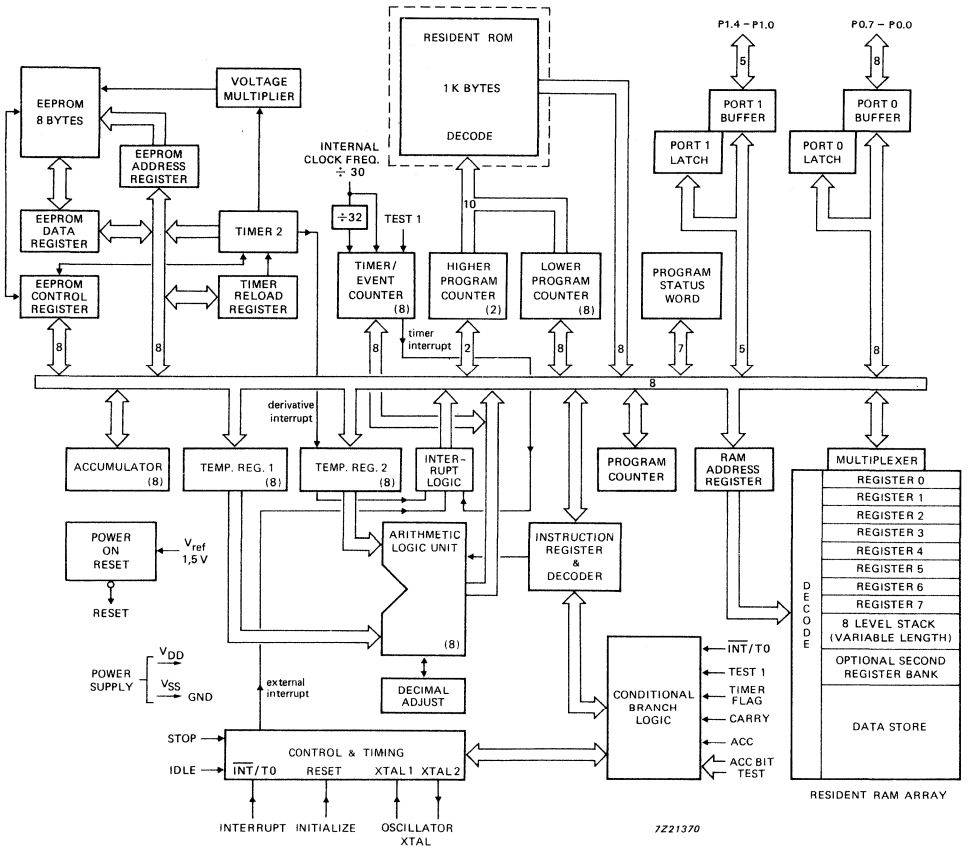



Fig. 1 Block diagram.

## PINNING

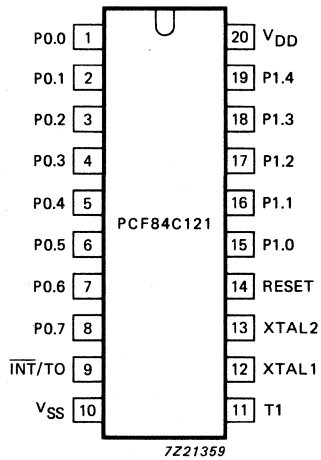



Fig. 2 Pinning diagram.

### Pin function

| *Pin* | *Symbol* | *Function* |
|---|---|---|
| 1-8 | P0.0-P0.7 | **Port 0:** 8-bit quasi-bidirectional I/O port. |
| 9 | $\overline{INT}$/T0 | **Interrupt/Test 0:** external interrupt input (negative edge triggered)/test input pin; when used as a test input, this pin is directly tested by conditional branch instructions JT0 and JNT0. |
| 10 | $V_{SS}$ | **Ground:** circuit earth potential. |
| 11 | T1 | **Test 1:** test input pin, directly tested by conditional branch instructions JT1 and JNT1. T1 may also be selected as an input to the 8-bit timer/event counter via the STRT CNT instruction. |
| 12 | XTAL 1 | **Oscillator input:** input from a crystal which determines the internal oscillator frequency or an external clock generator. |
| 13 | XTAL 2 | **Oscillator output** |
| 14 | RESET | **Reset input:** used to initialize the microcontroller (active HIGH); also output of power-on-reset circuit. |
| 15-19 | P1.0-P1.4 | **Port 1:** 5-bit quasi-bidirectional parallel I/O port. |
| 20 | $V_{DD}$ | **Power supply:** 2.5 to 5.5 V |

## FUNCTIONAL DESCRIPTION

### Program memory

The program memory consists of 1 K bytes of read-only memory (ROM). Each location is directly addressable by the program counter. The ROM is mask-programmed at the factory.

### Data memory

The data memory consists of 64 bytes of random-access data memory (RAM). All locations are indirectly addressable using RAM pointer registers; up to 16 designated locations are directly addressable. Memory also includes an 8-level program counter stack addressed by a 3-bit stack pointer.

### Derivative registers

The PCF84C121 contains 5 derivative registers (see Table 1). These registers are described below.

**Table 1** Derivative registers

| name | derivative address | function |
|---|---|---|
| ADDR | 01H | EEPROM array address (0. . . 7) |
| DATR | 03H | EEPROM read or write data |
| EPCR | 04H | EEPROM control register (status) |
| RELR | 05H | Timer 2 reload register |
| T2 | 06H | Timer 2 register |

| | MSB | | | | | | | LSB |
|---|---|---|---|---|---|---|---|---|
| ADDR (01H) | – | – | – | – | – | ADDR.2 | ADDR.1 | ADDR.0 |

Bits ADDR.7 to ADDR.3 are not used. ADDR.2, ADDR.1 and ADDR.0 are used to address one of the 8 EEPROM bytes. ADDR.2 is the most significant bit of the three bit EEPROM address.

DATR (03H)

The EEPROM data register (DATR) contains the last byte which was transferred to or from the EEPROM. When an EEPROM byte is read, it is transferred to the DATR register and then to the accumulator. The byte remains in DATR until the EEPROM is accessed again.

### I/O facilities

The PCF84C121 has 13 I/O lines arranged as:

- Port 0 — 8-bit parallel port (P0.0 to P0.7)
- Port 1 — 5-bit parallel port (P1.0 to P1.4)
- $\overline{\text{INT}}$/T0 — external interrupt and test input. When used as a test input, it can be directly tested by conditional branch instructions JT0 and JNT0
- T1 — test input which can alter program sequences when tested by conditional jump instructions JT1 and JNT1. T1 also functions as an input to the 8-bit timer/event counter.

### Timer 1

The maximum rate at which Timer 1 can be incremented is once every machine cycle. The timer flag is set when the counter overflows and can be tested and reset using the JTF (jump if timer flag is set) and JNTF (jump if timer flag is not set) instructions. An overflow also generates an interrupt by setting the Timer Interrupt Flag (TIF) when the timer/event counter interrupt is enabled.

### Timer 2

Timer 2 may be used to program the EEPROM or as a general purpose timer. It is started by setting the STT2 bit in the EPCR register. When it is started, Timer 2 is loaded with the contents of register RELR. When an overflow occurs, an interrupt flag is set and Timer 2 is automatically reloaded. Timer 2 can be read 'on the fly' by accessing the T2 derivative register.

Timer 2 may be used as a second timer as follows.

```
MOV A, #time          load A with the time to be down-counted
MOV RELR, A           load derivative register RELR
EN SI                 enable special interrupt
MOV A, #STT2 + EIT2   start timer and enable timer flag T2
ORL EPCR, A           load derivative control register with status
```

When a timer overflow occurs, a CALL to ROM address 5 is generated. The contents of the timer can be read 'on the fly' by a MOV A, T2 instruction. Execution of this instruction takes two machine cycles and it is possible for the timer to be incremented while this instruction is being executed.

### Reset (pin 14)

A positive-going signal on the RESET input:

- Sets the program counter to zero
- Selects register bank 0
- Sets the stack pointer to zero (000 points to RAM address 8)
- Disables the interrupts (external, Timer 1 and Timer 2)
- Stops the timer/event counter, then sets it to zero
- Sets the Timer 1 prescaler to modulo-32
- Resets the timer flags
- Sets all ports to input mode
- Cancels IDLE and STOP mode
- Clears all derivative registers and the Timer 2 prescaler

### Programming the EEPROM

All operations on the EEPROM are performed via the EEPROM Control Register (EPCR). EPCR is a derivative register which contains the status of the EEPROM and Timer 2.

### EEPROM Control Register EPCR

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| EPCR | STT2 | EIT2 | TF2 | EWP | MC3 | MC2 | MC1 | MC0 |

The EPCR bits are described below:

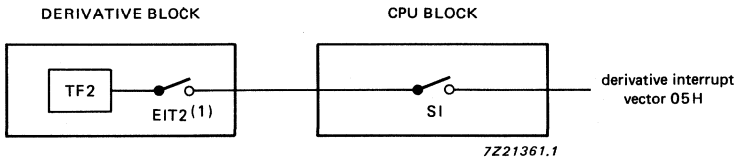
### STT2, Start Timer 2

Setting this bit clears the prescaler and initiates Timer 2 as a general purpose timer without affecting the EEPROM. Initially, Timer 2 is loaded with the contents of RELR and it is subsequently reloaded every time a timer overflow occurs. Resetting STT2 stops the timer, reloads it and clears the prescaler.

## FUNCTIONAL DESCRIPTION (continued)

### EIT2, Enable T2 Interrupt flag

When EIT2 is set and the special interrupt (SI) is enabled, the special interrupt will be requested when TF2 is set. If EIT2 is not set, no interrupt will be requested when TF2 is set (see Fig. 3).




(1) EIT2 SET: Hardware
RESET: Software

Fig. 3 Organization of the special interrupt (SI).

### TF2, Timer 2 Flag

This bit is set by hardware at the end of an erase/write or bulk erase cycle and when Timer 2 overflows while configured as a general purpose timer. A special interrupt is requested if enabled. TF2 must be cleared by software.

### EWP, Erase/write cycle in progress flag

This bit must be set by user software before an erase/write or bulk erase cycle can commence. It is cleared by hardware when the erase/write or bulk erase cycle is complete.

### Mode bits, MC0 to MC3

The mode bits and EWP determine the EEPROM operating mode as shown in Table 2.

**Table 2** EEPROM operating modes.

| mode | EWP | MC3 | MC2 | MC1 | MC0 | time (ms) |
|---|---|---|---|---|---|---|
| byte READ | 0 | 0 | 0 | 0 | 0 | – |
| byte ERASE/WRITE | 1 | 1 | 0 | 0 | 1 | 20 |
| bulk ERASE | 1 | 1 | 0 | 1 | 0 | 10 |
| byte WRITE | 1 | 0 | 1 | 1 | 0 | 10 |

Other bit combinations have no effect on the EEPROM of the PCF84C121. MC0 to MC3 can be read and written by software and are automatically cleared by hardware when an erase/write or bulk erase cycle is complete.

### The erase/write sequence

Before an erase/write sequence can commence, the RELR register must be loaded with the correct erase/write time (see Table 3). An erase/write sequence can be started if no erase or write sequence is in progress (EWP = logic 0). Six instructions are required to perform an erase/write sequence. The following performs a write operation:

```
MOV A,#data          load accumulator with EEPROM byte address
MOV ADDR, A          load ADDR with EEPROM byte address
MOV A, Rr            load accumulator with byte to be stored
MOV DATR, A          write byte to EEPROM
MOV A, #'0X011001'   erase/write cycle mode
MOV EPCR, A          commence erase/write cycle
```

When the microcontroller enters the IDLE mode, an erase/write cycle in progress will be completed. The interrupt request at the end of the erase/write cycle will cause the microcontroller to exit the IDLE mode if the interrupt is enabled. After the erase/write cycle has been completed, the EPCR register must be cleared by software.

Before entering the STOP mode, bit EWP in register EPCR should be tested. If an EEPROM write or erase cycle is in progress, this bit will be set and the STOP mode must not be entered.

**The bulk erase sequence**

Before a bulk erase sequence can commence, the RELR register must be loaded with the correct erase time (see Table 3). An erase or write sequence must not already be in progress (EWP = logic 0). The following clears all EEPROM bytes:

```
MOV A, #'0X011010'    bulk erase mode
MOV EPCR, A           commence bulk erase cycle
```

**The read sequence**

The read mode is automatically entered when an erase/write or bulk erase cycle is complete or after a RESET. The following is an example of a read operation:

```
MOV A, address    load accumulator with EEPROM address to be read
MOV ADDR, A       send address to ADDR derivative register
MOV A, DATR       load accumulator with EEPROM
MOV Rr, A         store data into register
```

The next byte may be read by repeating the above. An EEPROM byte can not be read while an erase/write or bulk erase operation is in progress.

**Adjusting the erase/write time**

Timer 2 determines the erase and write times during an erase/write cycle. Both times are equal and are of constant duration. Since the oscillator frequency may vary for different applications, a programmable timer and reload register (RELR) is required.

Timer 2 is clocked via a prescaler whose input is $f_{osc}$ /30. RELR is a derivative register and can be read or written. When an erase/write cycle starts, Timer 2 is loaded with the contents of RELR and started. When the first underflow occurs, the erase process is complete, Timer 2 is reloaded and the write process commences. When the second overflow occurs, the write process is complete, the timer is halted and an interrupt flag is set. The 8-bit timer and 4-bit prescaler can generate the correct erase and write durations for an oscillator frequency range of 450 kHz to 10 MHz.

**Table 3** Reload values for various oscillator frequencies

| $f_{XTAL}$ | reload value (Hex) |
|---|---|
| 450 kHz | 09H |
| 1 MHz | 15H |
| 2 MHz | 2AH |
| 3.58 MHz | 4BH |
| 6 MHz | 7DH |
| 10 MHz | DϕH |

Since T2 is incremented once every 480 clock cycles, the veloed value is calcilated as follows:

$$\frac{10\ ms}{480} \cdot f_{XTAL} = \text{reload value}$$

## DC CHARACTERISTICS

$V_{DD}$ = 2.5 to 5.5 V; $V_{SS}$ = 0 V; $T_{amb}$ = −40 to + 85 °C; all voltages with respect to $V_{SS}$; unless otherwise specified

| parameter | symbol | min. | typ. | max. | unit |
|---|---|---|---|---|---|
| Supply voltage | | | | | |
| operating (see Fig. 4) | $V_{DD}$ | 2.5 | – | 5,5 | V |
| Supply current | | | | | |
| operating (see Fig. 5 and note 2) | | | | | |
| at $V_{DD}$ = 5 V; $f_{XTAL}$ = 10 MHz | $I_{DD}$ | – | 2.0 | 3.5 | mA |
| at $V_{DD}$ = 5 V; $f_{XTAL}$ = 6 MHz | $I_{DD}$ | – | 1.2 | 2.4 | mA |
| at $V_{DD}$ = 3 V; $f_{XTAL}$ = 3.58 MHz | $I_{DD}$ | – | 0.4 | 0.8 | mA |
| IDLE mode (see Fig. 6 and note 2) | | | | | |
| at $V_{DD}$ = 5 V; $f_{XTAL}$ = 10 MHz | $I_{DD}$ | – | 1.0 | 2.0 | mA |
| at $V_{DD}$ = 5 V; $f_{XTAL}$ = 6 MHz | $I_{DD}$ | – | 0.7 | 1.4 | mA |
| at $V_{DD}$ = 3 V; $f_{XTAL}$ = 3.58 MHz | $I_{DD}$ | – | 0.25 | 0.6 | mA |
| STOP mode (see Fig. 9, note 1 and note 2) | | | | | |
| at $V_{DD}$ = 2.5 V; $T_{amb}$ = 85 °C | $I_{DD}$ | – | – | 10 | μA |
| Erase/write cycle | | | | | |
| time (see note 3) | $t_{e/w}$ | – | 10 | – | ms |
| data retention | Dr | 10 | – | – | years |
| **Inputs** | | | | | |
| Input voltage LOW | $V_{IL}$ | 0 | – | $0.3V_{DD}$ | V |
| Input voltage HIGH | $V_{IH}$ | $0.7V_{DD}$ | – | $V_{DD}$ | V |
| Input leakage current | | | | | |
| as $V_{SS} < V_I < V_{DD}$ | $\pm I_{IL}$ | – | – | 1 | μA |
| **Outputs** | | | | | |
| Output sink current LOW | | | | | |
| at $V_{DD}$ = 5 V ± 10%; $V_O$ = 0.4 V | $I_{OL}$ | 1.6 | 3 | – | mA |
| Pull-up output source current HIGH | | | | | |
| at $V_{DD}$ = 5 V ± 10%; $V_O$ = 0.7 $V_{DD}$ | $-I_{OH}$ | 40 | – | – | μA |
| at $V_{DD}$ = 5 V ± 10%; $V_O$ = $V_{SS}$ | $-I_{OH}$ | – | – | 400 | μA |
| Push-pull output source current HIGH | | | | | |
| at $V_{DD}$ = 5 V ± 10%; $V_O$ = $V_{DD}$−0.4 V | $-I_{OH}$ | 1.6 | 3 | – | mA |

**Notes to the DC characteristics**

1. Crystal connected between XTAL 1 and XTAL 2; T1 = $V_{SS}$; $\overline{INT}$ = $V_{DD}$.
2. $V_{IL}$ = $V_{SS}$; $V_{IH}$ = $V_{DD}$; all outputs unloaded; all open-drain outputs connected to $V_{SS}$.
3. Select reload value (Table 3) for $t_{e/w}$ = 10 ms. No shorter nor longer time is recommended.

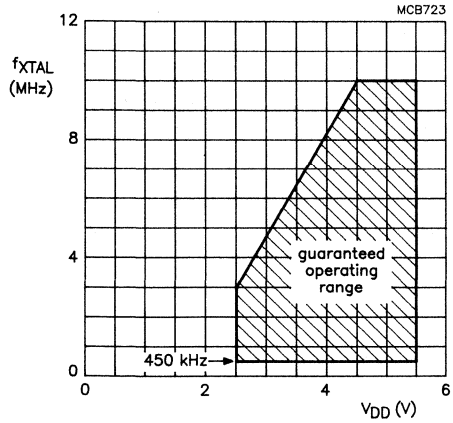
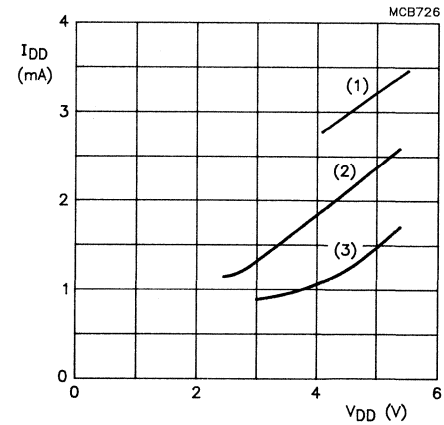
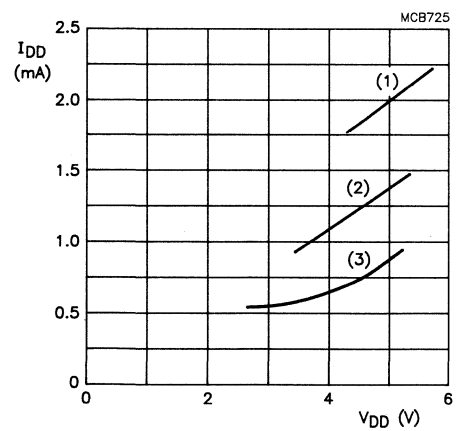



Fig. 4 Maximum clock frequency ($f_{XTAL}$) as a function of supply voltage ($V_{DD}$).




(1) $f_{XTAL}$ = 10 MHz
(2) $f_{XTAL}$ = 6 MHz
(3) $f_{XTAL}$ = 3.58 MHz

Fig. 5 Maximum supply current ($I_{DD}$) in operating mode as a function of the supply voltage.




(1) $f_{XTAL}$ = 10 MHz
(2) $f_{XTAL}$ = 6 MHz
(3) $f_{XTAL}$ = 3.58 MHz

Fig. 6 Maximum supply current ($I_{DD}$) in IDLE mode as a function of supply voltage ($V_{DD}$).

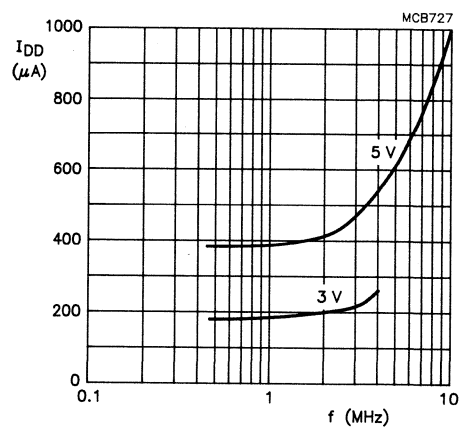



Fig. 7 Typical supply current during IDLE mode as a function of frequency at $V_{DD}$ = 3 V and $V_{DD}$ = 5 V.

**DC CHARACTERISTICS** (continued)

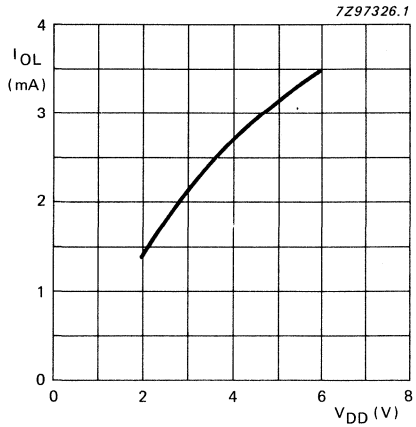
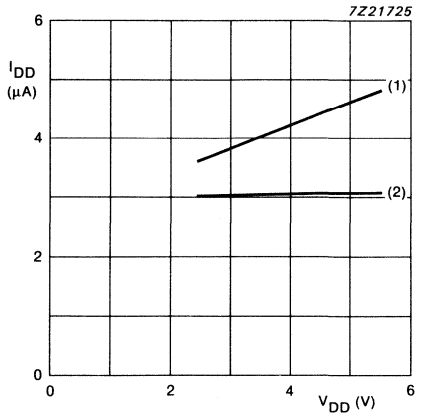
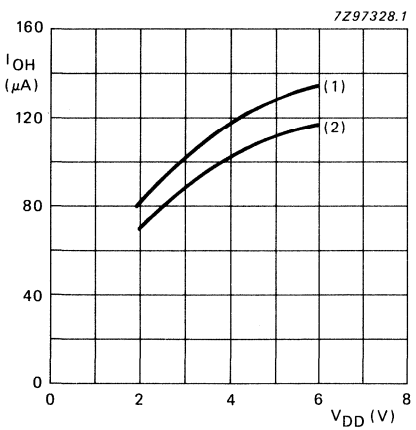



Fig. 8 Typical output sink current ($I_{OL}$), as a function of supply voltage ($V_{DD}$); $V_O$ = 0.4 V.




(1) $T_{amb}$ = 85 °C
(2) $T_{amb}$ = 25 °C

Fig. 9 Typical supply current ($I_{DD}$) in STOP mode as a function of the supply voltage ($V_{DD}$).




(1) $V_O = V_{SS}$
(2) $V_O = 0.7\ V_{DD}$

Fig. 10 Typical output source current ($-I_{OH}$) as a function of supply voltage ($V_{DD}$).

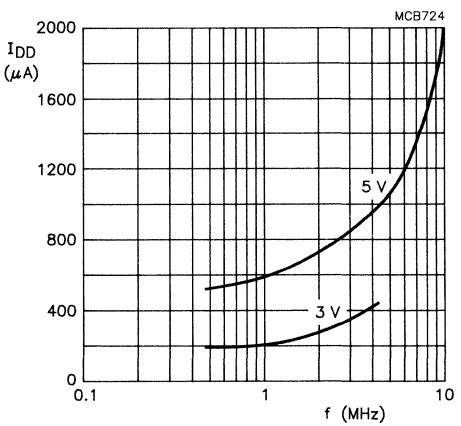



Fig. 11 Typical supply current during operating mode as a function of frequency at $V_{DD}$ = 3 V and $V_{DD}$ = 5 V.

# 8-bit microcontrollers for remote control transmitters

# PCA84C122; 222; 422; 622; 822

## CONTENTS



## 1 FEATURES

- 84CXXX CPU
- ROM, RAM and I/O are device dependent, see Chapter 3
- Two test inputs T0 (ANDed with P1 input lines), T1
- 3 single-level vectored interrupt sources: external (T0/$\overline{\text{INT}}$ and Port 1, for keypad press wake-up function), timer/counter (TI) and hardware modulator interrupt
- 8-bit programmable timer/counter with 5-bit pre-scaler
- On-board oscillator 1 MHz to 6 MHz
- Single supply voltage from 2.0 V to 5.5 V
- Operating temperature range: –20 to +50 °C
- Power saving modes: Idle and Stop modes are provided
- 'Hardware Modulator' that provides pulse bursts of which the 'on' time and 'off' time of each pulse (i.e. duty cycle) and the number of pulses are programmable
- One output line from the 'Hardware Modulator' to control the driver transistor for the IR-LED. Capable of sinking 27 mA at $V_{DD}$ = 2.0 V, $V_{\overline{OUT}}$ = 1.0 V
- Watchdog timer to keep the transmitter from being locked or malfunction. Automatic system reset is generated by the WDT if the timer is not reset before overflow from counting within a certain period of time
- Available packages: SO and DIL types (SO20L, SO24L, SO28L, SDIL24 and DIL20), see Chapter 4.

## 2 GENERAL DESCRIPTION

The 84C122 is a stand-alone microcontroller designed for use in remote control transmitters for a wide range of applications. The 84C122 for this purpose provides a number of dedicated hardware functions for remote controller applications.

These include the following additional blocks to the 84CXXX core:

- Interrupt Gate
- Hardware Modulator
- Output Driver
- Watchdog Timer.

Although the 84C122 is specifically referred to throughout this data sheet, the information applies to all the devices. The small differences between the 84C122 and the other devices are specified in the text and also highlighted in Chapter 3.

The 84CXXX core 8-bit microcontroller family specification is described in *"Section 4"* of *"Data Handbook IC14"*. With reference specifically to Chapter *"Functional description"* (interrupts, reset, stop mode etc.) and to the Chapter *"Instruction set"*.

The general block diagram of the device is shown in Fig.2. The 84CXXX core plus 8 kbytes ROM and 64 bytes RAM has the same function as described in the data sheet of the 84CXXX.

When the transmitter is not in use the microcontroller is in STOP mode and the oscillator is HALTED. The AND gate from P1 Port line provides the wake-up to end STOP mode.

The Hardware Modulator produces pulse bursts according to the required protocol. By software the 'ON-time' and the 'OFF-time' of each pulse and the number of pulses are controlled.

The Output Driver can handle sufficient current to drive a single transistor, and this can provide the required current for the LED.

The Watchdog Timer will reset the 84C122 when it has not been reloaded (reset) in time, because the program has run out of sequence (endless loop, continuous IDLE mode, etc.). During STOP mode the oscillator is halted, so then the Watchdog Timer is not running.

## 3 MEMORY AND I/O CONFIGURATIONS

| DEVICE | RAM | ROM | I/O LINES |
|---|---|---|---|
| PCA84C122A | 32 bytes | 1K | 16 |
| PCA84C122B | 32 bytes | 1K | 12 |
| PCA84C222A | 32 bytes | 2K | 16 |
| PCA84C222B | 32 bytes | 2K | 12 |
| PCA84C422A | 32 bytes | 4K | 16 |
| PCA84C422B | 32 bytes | 4K | 12 |
| PCA84C622A | 64 bytes | 6K | 16 |
| PCA84C622B | 64 bytes | 6K | 12 |
| PCA84C622C | 64 bytes | 6K | 20[1] |
| PCA84C822A | 64 bytes | 8K | 16 |
| PCA84C822B | 64 bytes | 8K | 12 |
| PCA84C822C | 64 bytes | 8K | 20[1] |

**Note**

1. 4 I/O lines with 10 mA sink capability.

## 4 ORDERING INFORMATION

| EXTENDED TYPE NUMBER | PACKAGE | | | |
|---|---|---|---|---|
| | PINS | PIN POSITION | MATERIAL | CODE |
| PCA84C122AP | 24 | SDIL24 | plastic | SOT234A |
| PCA84C122AT | 24 | SO24L | plastic | SOT137A |
| PCA84C122BP | 20 | DIL20 | plastic | SOT146E |
| PCA84C122BT | 20 | SO20L | plastic | SOT163A |
| PCA84C222AP | 24 | SDIL24 | plastic | SOT234A |
| PCA84C222AT | 24 | SO24L | plastic | SOT137A |
| PCA84C222BP | 20 | DIL20 | plastic | SOT146E |
| PCA84C222BT | 20 | SO20L | plastic | SOT163A |
| PCA84C422AP | 24 | SDIL24 | plastic | SOT234A |
| PCA84C422AT | 24 | SO24L | plastic | SOT137A |
| PCA84C422BP | 20 | DIL20 | plastic | SOT146E |
| PCA84C422BT | 20 | SO20L | plastic | SOT163A |
| PCA84C622AP | 24 | SDIL24 | plastic | SOT234A |
| PCA84C622AT | 24 | SO24L | plastic | SOT137A |
| PCA84C622BP | 20 | DIL20 | plastic | SOT146E |
| PCA84C622BT | 20 | SO20L | plastic | SOT163A |
| PCA84C622CT | 28 | SO28L | plastic | SOT136A |
| PCA84C822AP | 24 | SDIL24 | plastic | SOT234A |
| PCA84C822AT | 24 | SO24L | plastic | SOT137A |
| PCA84C822BP | 20 | DIL20 | plastic | SOT146E |
| PCA84C822BT | 20 | SO20L | plastic | SOT163A |
| PCA84C822CT | 28 | SO28L | plastic | SOT136A |

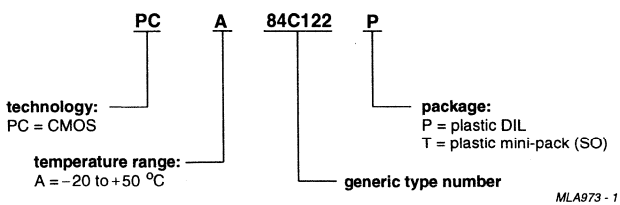



Fig.1 Numbering scheme.

## 5 BLOCK DIAGRAM

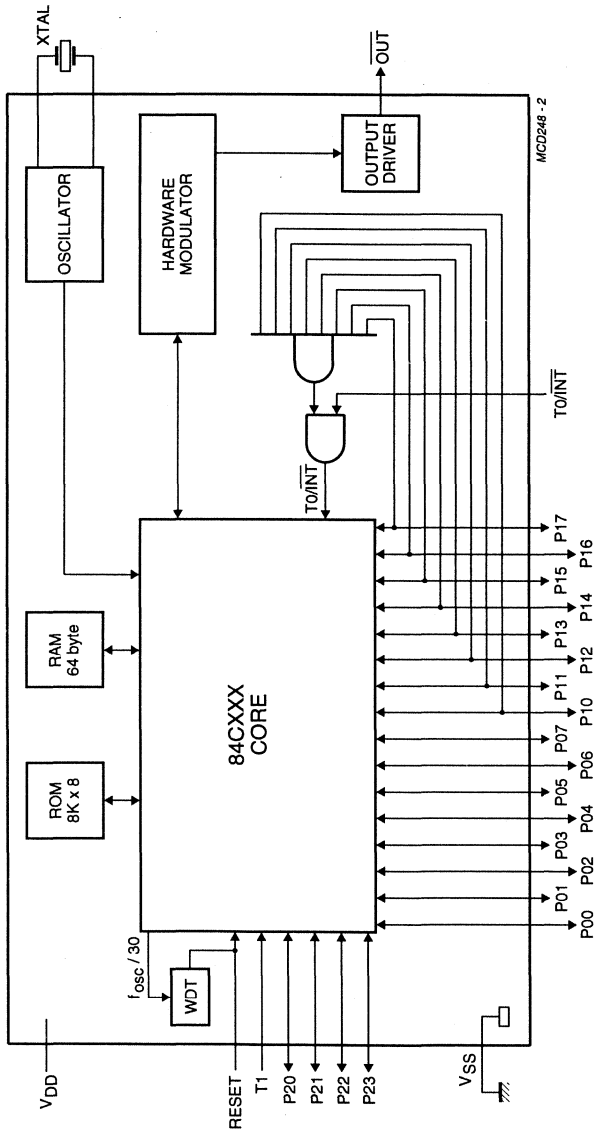



Fig.2 Block diagram.

## 6 PINNING INFORMATION

### 6.1 Pinning

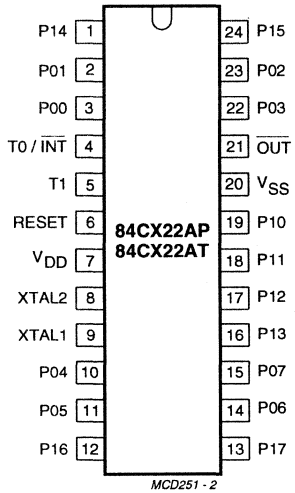



Fig.3 Pin configurations (SDIL24 and SO24L) PCA84CX22AP and PCA84CX22AT.

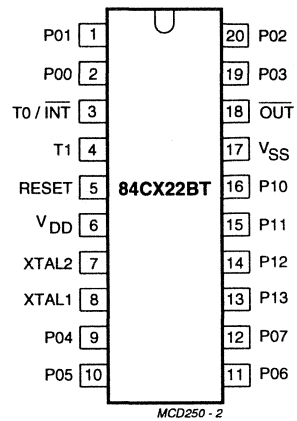



Fig.4 Pin configuration (SO20L) PCA84CX22BT.

### 6.2 Pin description

**Table 1** See Figs 3 and 4.

| SYMBOL | PIN | | DESCRIPTION |
|---|---|---|---|
| | SDIL24 SO24L | SO20L | |
| P00 – P07 | 3, 2, 23, 22 10, 11, 14, 15 | 2, 1, 20, 19 9, 10, 11, 12 | standard I/O Port lines, generally used for keypad scanning |
| P10 – P17 | 19, 18, 17, 16 1, 22, 12, 13 | 16, 15, 14, 13 | standard I/O Port lines, generally used for keypad scanning |
| T0/$\overline{INT}$ | 4 | 3 | test T0 and external interrupt input |
| T1 | 5 | 4 | test T1 input |
| RESET | 6 | 5 | active HIGH reset, normally tied to $V_{SS}$, however, see PCF84CXXX description in *"Data Handbook IC14"*. |
| XTAL1/2 | 9, 8 | 8, 7 | crystal or ceramic resonator |
| $\overline{OUT}$ | 21 | 18 | pulse train output pin, capable of sinking 27 mA |
| $V_{DD}$ | 7 | 6 | power supply |
| $V_{SS}$ | 20 | 17 | ground |

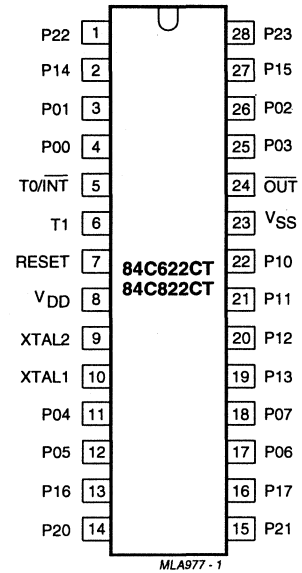



Fig.5 Pin configuration (SO28L) PCA84CX22CT.

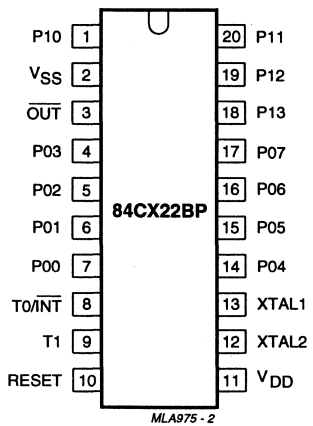



Fig.6 Pin configuration (DIL20) PCA84CX22BP.

**Table 2** See Figs 5 and 6.

| SYMBOL | PIN | | DESCRIPTION |
|---|---|---|---|
| | SO28L | DIL20 | |
| P00 – P07 | 4, 3, 26, 25<br>11, 12, 17, 18 | 7, 6, 5, 4<br>14, 15, 16, 17 | standard I/O Port lines, generally used for keypad scanning |
| P10 – P17 | 22, 21, 20, 19<br>2, 27, 13, 16 | 1, 20, 19, 18 | standard I/O Port lines, generally used for keypad scanning |
| P20 – P23 | 14, 15, 1, 28 | – | standard I/O Port lines, generally used for visible LED's |
| T0/$\overline{INT}$ | 5 | 8 | test T0 and external interrupt input |
| T1 | 6 | 9 | test T1 input |
| RESET | 7 | 10 | active HIGH reset, normally tied to $V_{SS}$, however, see PCF84CXXX description in *"Data Handbook IC14"*. |
| XTAL1/2 | 10, 9 | 13, 12 | crystal or ceramic resonator |
| $\overline{OUT}$ | 24 | 3 | pulse train output pin, capable of sinking 27 mA |
| $V_{DD}$ | 8 | 11 | power supply |
| $V_{SS}$ | 23 | 2 | ground |

## 7 POWER-ON RESET STATUS AND PORT OPTIONS

- All the Port lines are standard I/O (option 1)
- RESET (power-on reset) level of 1.3 V.

After power-on reset, Port 0 is reset to LOW and Port 1, Port 2 and $\overline{OUT}$ are reset to HIGH.

## 8 INTERFACE BETWEEN THE 84CXXX CORE AND DERIVATIVE LOGIC

There are three registers and one counter that must be loaded from the core:

- ON-time register; contents + 2 equals the number of oscillator cycles that the $\overline{OUT}$ pin is active during a single pulse.
- OFF-time register; contents + 2 equals the number of oscillator cycles that the $\overline{OUT}$ pin is deactivated between two consecutive pulses.
- PULSE COUNTER; should be loaded with the number of pulses in a pulse burst. Loading with 0 is not allowed.
- CONTROL register; contains the bits that control various possibilities for the output pulses, plus the reset of the interrupt flip-flop and the Watchdog timer.

### 8.1 Derivative register numbers

1. ON-time register
2. OFF-time register
3. PULSE COUNTER LOW
4. CONTROL register
5. PULSE COUNTER HIGH

#### 8.1.1 Instructions for data transfer between the 84CXXX core and derivative

| | |
|---|---|
| MOV Dx,A | Move contents of the accumulator to the derivative register. |
| MOV A,Dx | Move contents of derivative register to the accumulator. |
| ANL Dx,A | AND derivative register contents with contents of accumulator. Result is stored in derivative register. |
| ORL Dx,A | OR derivative register contents with contents of accumulator. Result is stored in derivative register. |

## 9 HARDWARE MODULATOR

The main derivative part in the PCA84C122 is the Hardware Modulator. Its function is to generate a pulse train whereby the ON-time of a single pulse and the OFF-time between pulses can be programmed in a number of clock cycles (period = $1/f_{osc}$); see Figs 7 and 8. The number of pulses of the train is also programmable. The time between pulse bursts is determined by software, or using the 8-bit timer.

Figure 8 shows the internal architecture of the Hardware Modulator.

The registers ON-time and OFF-time are loaded by software. The number of pulses required is loaded into the PULSE counter also by means of software. Both ON-time and OFF-time registers, and counters except the CONTROL register are 8-bits wide so values of 0 to 255 clock cycles $1/f_{osc}$ may be loaded. The Pulse Counter Register is a 10-bit long register, and actually consists of two registers. The Pulse Counter High is a 2-bit register and the Pulse Counter Low is an 8-bit register.

### 9.1 Operation of the Hardware Modulator

First the ON-time and OFF-time registers should be loaded. The actual ON-time and OFF-time are equal to: (register contents + 2) $\times$ $1/f_{osc}$. The number of pulses that should be generated must be stored in the PULSE COUNTER. As soon as the PULSE COUNTER is loaded the Hardware Modulator starts. First the signal $\overline{OUT}$ is activated and ON-time is stored in the PULSE TIMER. Then this timer is counted down with oscillator pulses $1/f_{osc}$. When it reaches zero the signal $\overline{OUT}$ is deactivated and the PULSE COUNTER is decreased by 1. If this is not yet zero, OFF-time is stored in the PULSE TIMER and the timer is counted down again. When it reaches zero now, signal $\overline{OUT}$ is activated again, ON-time is loaded and the counter starts counting down etc.

When the PULSE COUNTER has reached zero an interrupt will be given to the 84CXXX core, (Special Interrupt, SI) to indicate that the Hardware Modulator is ready.

The delay between two pulse bursts has to be timed by software. When a new burst must be generated with the same ON-time and OFF-time, only the PULSE COUNTER should be loaded with the required number of pulses and the Hardware Modulator will start automatically. The interrupt signal will be reset when the PULSE COUNTER is loaded. It may also be reset directly by writing a '1' to bit-3 of the CONTROL REGISTER.

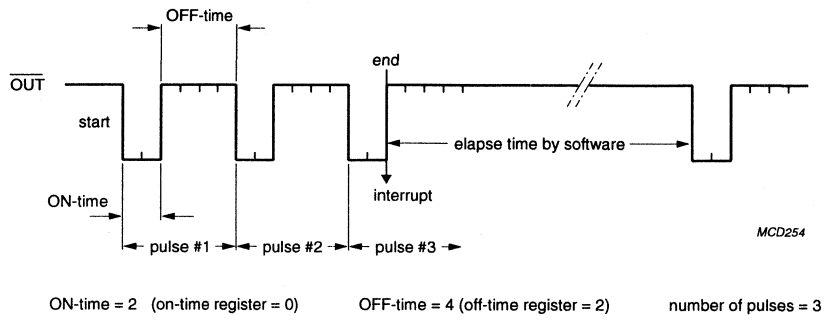



Fig.7 Example pulse train output of $\overline{OUT}$ pin.

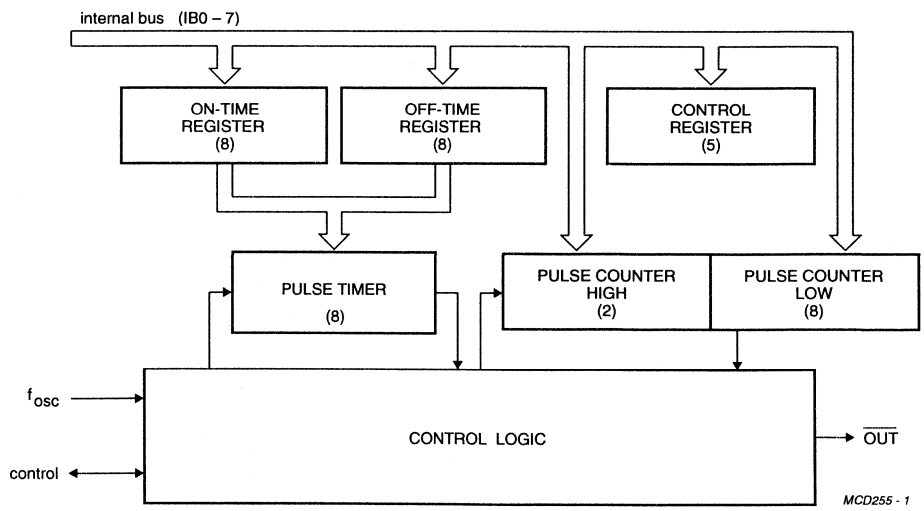



Fig.8 Hardware Modulator.

## 10 CONTROL REGISTER

Via the CONTROL REGISTER various modes of the Hardware Modulator can be selected, and also the Interrupt and Watchdog Timer can be reset.

**Table 3** Control Register; address (03H).

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| – | – | – | RWDT | Rint | PWM | LgP | HF |

**Table 4** Description of Control Register bits.

| BIT | MODE | DESCRIPTION |
|---|---|---|
| 7 to 5 | – | Reserved |
| 4 | RWDT | **Reload Watchdog Timer:** This is not an actual flip-flop in the control register. If DA4 is '1' when writing to the control register the Watchdog Timer is reloaded (reset to 0); see note 2. |
| 3 | Rint | **Reset Interrupt:** This is not an actual flip-flop in the control register. If DA3 is '1' when writing to the control register the interrupt flip-flop is reset (see note 2). |
| 2 | PWM | **Pulse Width Modulation:** Is the standard term for a quasi analog signal. It is a square wave signal of which the duty cycle may be varied. When integrated a real analog signal may be obtained. When the PWM bit is '1' the 'PULSE COUNTER' register is ignored so that a continuous pulse train is generated (see note 1 and Fig.11). |
| 1 | LgP | **Long Pulse:** When this bit is '1' the 'OFF-time' register is ignored. The pulse burst consists of a single pulse with a length of 'ON-time' × 'number of pulses'. If HF = '1' this pulse is modulated with a frequency $f_{osc}/4$ (see note 1 and Fig.10). |
| 0 | HF | **High Frequency:** When this bit is '1' the 'ON-time' part of the generated pulse is modulated with a frequency $f_{osc}/4$ (see note 1 and Fig.9). |

**Notes**

1. This bit is Read or Write.
2. This bit is Write only.

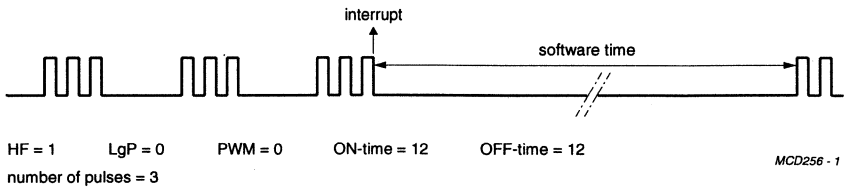



Fig.9 Modulated by $f_{osc}/4$.

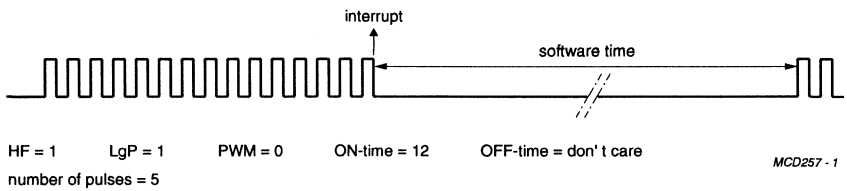



Fig.10 Long Pulse (modulated).

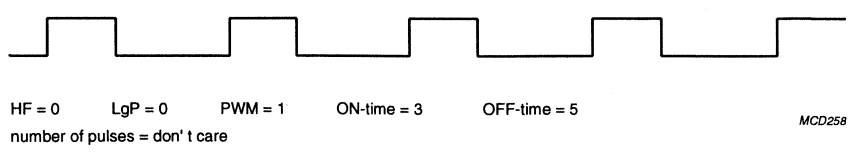



Fig.11 Analog output (PWM).

## 11 INTERRUPTS

There are three different interrupt sources with different vectors within the PCA84C122, these are:

- External Keypad wake-up and T0/$\overline{INT}$ pin - vector address 03H
- Hardware modulator (SI) - vector address: 05H
- Internal Timer/counter (TI) - vector address: 07H.

### 11.1 Internal timer/counter

The internal timer/counter is the same as in the other members of the PCF84CXXX family. Instructions 'EN TCNTI' and 'DIS TCNTI' are to enable/disable the interrupt.

### 11.2 External keypad wake-up and T0/$\overline{INT}$ pin

#### 11.2.1 KEYPAD WAKE-UP

The keypad wake-up function is to generate an interrupt signal to the CPU. This signal is used to terminate STOP mode and to cause program execution to continue. If the external interrupt is enabled the microcontroller executes the instruction immediately following the STOP instruction before executing the interrupt routine. If the external interrupt is disabled program execution continues with the instruction following the STOP instruction.

The wake-up function is activated when any of the sense lines (P1 pins) are pulled LOW, an interrupt signal is generated.

#### 11.2.2 T0/$\overline{INT}$ PIN

No internal pull-up or pull-down is present (floating input), so when not in use T0/$\overline{INT}$ should be connected to $V_{DD}$.

Because this pin is ANDed with P1 inputs, the result of JTO and JNTO instructions and the interrupt depends on the AND of all P1 and T0/$\overline{INT}$ inputs.

When a pull-up resistor is connected, T0/$\overline{INT}$ can serve as an extra sense line, however, multiple keys will not be detected.

## 12 OUTPUT DRIVER

The output of the Hardware Modulator is amplified and able to sink a current of 27 mA when the $\overline{OUT}$ signal is active. This means only one external (PNP) transistor is required to drive the LED.

Output sink current LOW
(at $V_{DD}$ = 2.0 V < $V_{\overline{OUT}}$ < $V_{DD}$ −1.0 V) minimum 27 mA.

## 13 WATCHDOG TIMER (WDT)

The watchdog timer consists of a 17 stage counter with a period = '30 × 1/$f_{osc}$' as its clock source. A '1' written to bit-4 of the control register (derivative register '03') clears the WDT to zero.

After a power-on reset, the WDT is cleared to zero. Then every 30-clock cycles it is increased, if it is not at zero before it overflows, a hardware RESET signal is generated to reset the whole chip. This will prevent the chip from being locked up or malfunction. The software has to clear the WDT at least 1/$f_{osc}$ × 30 × $2^{16}$ seconds (i.e. at $f_{osc}$ = 1 MHz, it is 1.92 s) before it generates a reset.

In the Idle mode, the WDT is still active as the oscillator is still working. In the Stop mode, the WDT is fully stopped and the value of the counter is kept.

## 14 LIMITING VALUES

In accordance with the Absolute Maximum Rating System (IEC 134).

| SYMBOL | PARAMETER | MIN. | MAX. | UNIT |
|---|---|---|---|---|
| $V_{DD}$ | supply voltage | −0.5 | +7.0 | V |
| $V_I$ | all input voltages | −0.5 | $V_{DD}$ + 0.5 | V |
| $I_{OH}$ | maximum source current (all Port lines) | – | −5 | mA |
| $I_{OL}$ | maximum sink current (all Port lines) | – | 5 | mA |
| $I_{\overline{OUT}}$ | maximum output current | 27 | – | mA |
| $I_{rev}$ | maximum reverse current | – | −500 | mA |
| $P_{tot}$ | total power dissipation | – | 500 | mA |
| $T_{stg}$ | storage temperature | −55 | +125 | °C |
| $T_{amb}$ | operating ambient temperature | −20 | +50 | °C. |

## 15 DC CHARACTERISTICS

$V_{DD}$ = 4.5 V to 5.5 V; $V_{SS}$ = 0 V; $T_{amb}$ = −20 to +50 °C; all voltages with respect to $V_{SS}$; unless otherwise specified.

| SYMBOL | PARAMETER | CONDITIONS | MIN. | TYP. | MAX. | UNIT |
|---|---|---|---|---|---|---|
| **Supply** | | | | | | |
| $V_{DD}$ | supply voltage | | 2.0 | 3.0 | 5.5 | V |
| $I_{DD}$ | supply current operating mode | $V_{DD}$ = 3 V; $f_{xtal}$ = 3 MHz | – | 0.4 | 0.9 | mA |
| | | $V_{DD}$ = 5 V; $f_{xtal}$ = 3 MHz | – | 0.9 | 1.8 | mA |
| $I_{DD(ID)}$ | supply current Idle mode | $V_{DD}$ = 3 V; $f_{xtal}$ = 3 MHz | – | 0.20 | 0.40 | mA |
| | | $V_{DD}$ = 5 V; $f_{xtal}$ = 3 MHz | – | 0.25 | 0.50 | mA |
| $I_{DD(ST)}$ | supply current Stop mode | $V_{DD}$ = 5 V; $f_{xtal}$ = 3 MHz | – | 1.20 | 10 | µA |
| **Inputs (RESET, T0/$\overline{INT}$, T1, P00 – P07, P10 – P17, P20 – P23)** | | | | | | |
| $V_{IL}$ | LOW level input voltage | | 0 | – | 0.3$V_{DD}$ | V |
| $V_{IH}$ | HIGH level input voltage | | 0.7$V_{DD}$ | – | $V_{DD}$ | V |
| $I_{LI}$ | input leakage current | $V_{SS} < V_I < V_{DD}$ | – | 0.20 | 10 | µA |
| **Outputs (Ports P00 – P07, P10 – P17, P20 – P23)** | | | | | | |
| $I_{OL}$ | sink current LOW | $V_{DD}$ = 5 V; $V_O$ = 0.4 V | 1.6 | 12.0 | – | mA |
| | | $V_{DD}$ = = 5 V; $V_O$ = 0.4 V; P20 – P23 | 10 | – | – | mA |
| $I_{OH}$ | pull-up output source current HIGH | $V_{DD}$ = 5 V; $V_O$ = 0.7$V_{DD}$ | −40 | −100 | – | µA |
| | | $V_{DD}$ = 5 V; $V_O$ = $V_{SS}$ | −140 | −140 | −400 | µA |
| PULSE OUTPUT ($\overline{OUT}$) | | | | | | |
| $I_{OL}$ | sink current LOW | $V_{DD}$ = 2 V; $V_O$ = 1.0 V | 27 | – | – | mA |
| $I_{OH}$ | source current HIGH | $V_{DD}$ = 2 V; $V_O$ = 0.7$V_{DD}$ | −1.6 | – | – | mA |

## 16 AC CHARACTERISTICS

$V_{DD}$ = 2.0 V to 5.5 V; $V_{SS}$ = 0 V; $T_{amb}$ = −20 to +50 °C; all voltages with respect to $V_{SS}$; unless otherwise specified.

| SYMBOL | PARAMETER | CONDITIONS | MIN. | TYP. | MAX. | UNIT |
|---|---|---|---|---|---|---|
| **Oscillator** | | | | | | |
| $f_{xtal}$ | operating crystal frequency | $V_{DD}$ = 2.5 V | 1.0 | – | 6.0 | MHz |
| | | $V_{DD}$ = 2.0 V | 1.0 | – | 5.0 | MHz |
| $g_{mL}$ | option low transconductance $g_m$ | | 0.3 | 0.7 | 1.4 | mS |
| $g_{mM}$ | option medium transconductance $g_m$ | | 0.9 | 1.6 | 3.2 | mS |
| $g_{mH}$ | option high transconductance $g_m$ | | 3.0 | 4.5 | 9.0 | mS |
| $R_{FB}$ | feedback resistor | | 0.3 | 1.0 | 3.0 | MΩ |

## 17 APPLICATION INFORMATION

The main application for the 84C122 is as a remote control transmitter, see Fig.12.

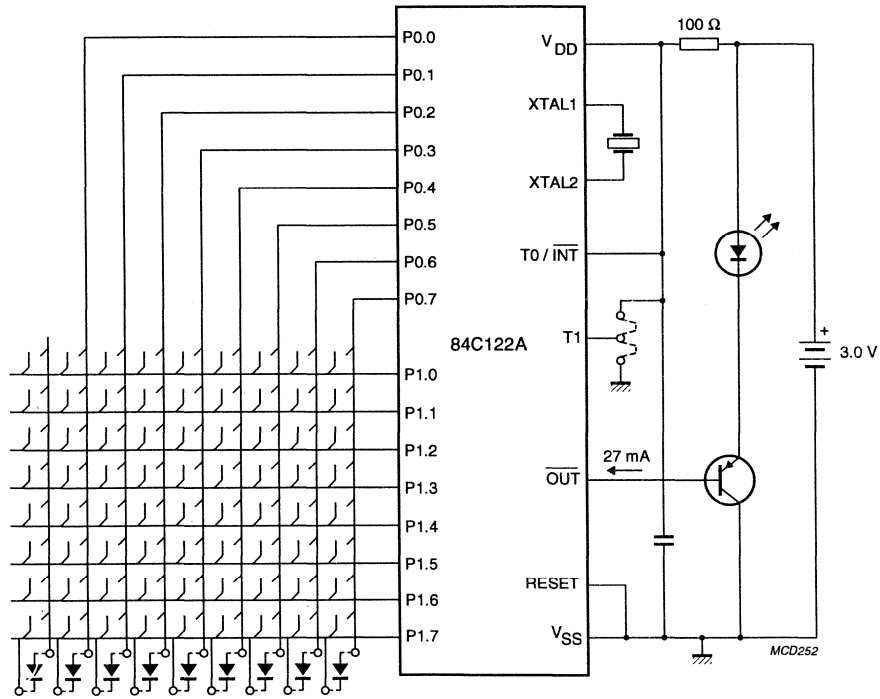



Fig.12 PCA84C122A Remote control transmitter application.

The transmitter is provided with a keypad in which each key represents a specific command. The keys are in an orderly matrix, with each key connected between an I/O line of Port 0 and an I/O line of Port 1. The lines of Port 0 are designated 'scan' (output) lines, and the lines of Port 1 are 'sense' (input) lines. By making each scan line logic 0 in turn, and each time looking at the sense lines, the key that is depressed is detected. With the use of a software look-up table the corresponding command code is determined. This code together with the system address is sent according to a coding scheme or protocol (e.g. RC-5). The pulses that are generated are available at the $\overline{OUT}$ pin. This pin has sufficient current capability to drive the output transistor, which in turn provides the current for the infrared LED.

When T0/$\overline{INT}$ is not used, it should be connected to $V_{DD}$, because no internal pull-up is present.

For system or option selection T1 may be used. A jumper may be connected to $V_{DD}$ or to ground.

When more options must be selected, this can be done in different ways.

Two examples are given:

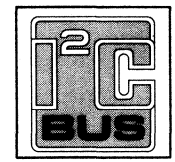
1. When the number of keys is sufficiently low so that a scan line can be spared, this scan line may be used to connect jumper wires to the sense lines. Normally the scan line should be logic 1. When this line is set to logic 0 the option setting can be read from the sense lines.
2. By connecting diodes between one sense line and a number of scan lines (see Fig.12). If necessary these diodes can be placed parallel to keys. When the sense line is set to logic 0 (acts as scan line), the option setting can be read from the scan lines (which act as sense lines and should be set to logic 1 beforehand).

For the oscillator a crystal or ceramic resonator may be used.

A resistor in series should limit the reverse current through the IC in the case of the supply current being reversed (i.e. wrong insertion of batteries).

# 8-bit microcontrollers with OSD and VST

# 84C44X; 84C64X; 84C84X




**CONTENTS**

## 1. FEATURES

### 1.1 84CXXX kernel

- 8-bit CPU, ROM, RAM, I/O in a single 42-lead shrink DIL package
- Over 80 instructions all of 1 or 2 cycles
- 29 quasi bidirectional standard I/O port lines
- Configuration of I/O lines individually selected by mask
- External interrupt T0/INTN
- 2 direct testable inputs T0, T1
- 8-bit programmable timer/event counter
- 3 single level vectored interrupts (external, timer/counter, I²C)
- Power-on-reset and low voltage detector
- Single power supply
- 2 power reduction modes: Idle and Stop
- Operating temperature range: –20 to +70 °C.
- Silicon gate CMOS fabrication process

### 1.2 Derivative features PCA84C640

- 6K bytes ROM
- 128 bytes RAM
- Multi-master I²C bus interface
- One 14-bit PWM output for VST
- AFC input (with 3-bit DAC and comparator)
- Five 6-bit PWM outputs for analog controls
- Eight port lines with 10 mA LED drive capability
- On screen display 2 rows of 16 characters
- OSD character set of 64 types
- Four programmable display dot sizes
- Seven colours for each character
- Half dot character rounding
- Programmable VSYNCN input polarity
- Programmable HSYNCN input polarity
- LC oscillator for on screen display function with the 84C441, 84C444, 84C641, 84C644, 84C841 and 84C844.
- RC oscillator for on screen display function with the 84C440, 84C443, 84C640, 84C643, 84C840 and 84C843.

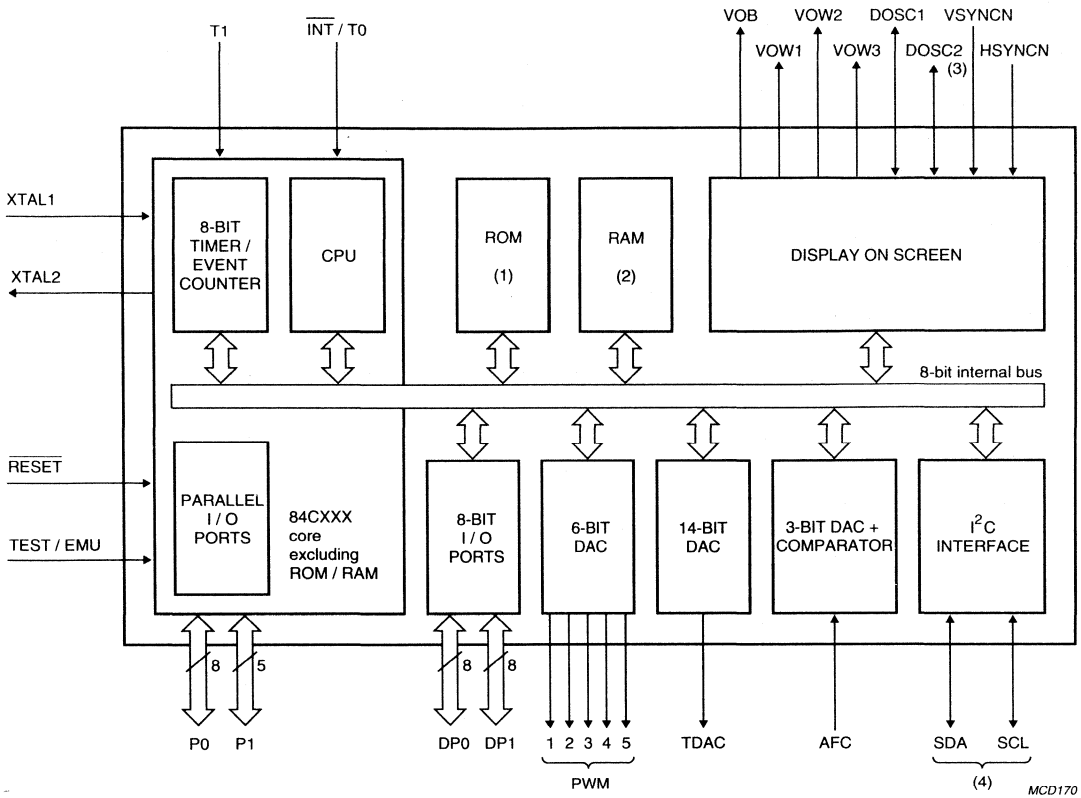
## 2. GENERAL DESCRIPTION

The 84C440, 84C441, 84C443, 84C444, 84C640, 84C641, 84C643, 84C644, 84C840, 84C841, 84C843 and 84C844 are 8-bit microcontrollers with On Screen Display (OSD) and Voltage Synthesized Tuning (VST) functions. All are members of the 84CXXX microcontroller family. Although the 84C640 is specifically referred to throughout this data sheet, the information applies to all the devices. The small differences between the 84C640 and the other devices are specified in the text and also highlighted in Section 12.

The 84C640 comprises the 84CXXX processor core, 6K bytes mask-programmable program ROM, 128 bytes RAM, a multi-master I²C bus interface, 2 direct testable lines, 18 general purpose bi-directional I/O lines plus 11 function-combined I/O lines, one 14-bit and five 6-bit PWM analog control outputs; AFC input for Voltage Synthesized Tuning and an On Screen Display facility for two rows of 16-characters.

**Important**

This data sheet details the specific properties of the 84C440, 84C441, 84C443, 84C444, 84C640, 84C641, 84C643, 84C644, 84C840, 84C841, 84C843, and 84C844. The shared characteristics of the 84CXXX family of microcontrollers are described in the 84CXXX family specification, which should be read in conjunction with this data sheet.




(1) 4 Kbytes for the 84C440, 84C441, 84C443 and 84C444. 6 Kbytes for the 84C640, 84C641, 84C643 and 84C644. 8 Kbytes for the 84C840, 84C841, 84C843 and 84C844.

(2) 128 bytes for the 84C440, 84C441, 84C443, 84C444, 84C640, 84C641, 84C643 and 84C644. 192 bytes for the 84C840, 84C841, 84C843 and 84C844.

(3) For use with an LC oscillator; only available with the 84C441, 84C444, 84C641, 84C644, 84C841 and 84C844.

(4) I²C-bus interface not available with the 84C443, 84C444, 84C643, 84C644, 84C843 and 84C844.

Fig.1 Block diagram.

## 3. ORDERING INFORMATION

| EXTENDED TYPE NUMBER | PACKAGE | | | | TEMPERATURE RANGE (°C) |
|---|---|---|---|---|---|
| | PINS | PIN POSITION | MATERIAL | CODE | |
| PCA84C440 | 42 | SDIL | plastic | SOT270 | −20 to + 70 |
| PCA84C441 | 42 | SDIL | plastic | SOT270 | −20 to + 70 |
| PCA84C443 | 42 | SDIL | plastic | SOT270 | −20 to + 70 |
| PCA84C444 | 42 | SDIL | plastic | SOT270 | −20 to + 70 |
| PCA84C640 | 42 | SDIL | plastic | SOT270 | −20 to + 70 |
| PCA84C641 | 42 | SDIL | plastic | SOT270 | −20 to + 70 |
| PCA84C643 | 42 | SDIL | plastic | SOT270 | −20 to + 70 |
| PCA84C644 | 42 | SDIL | plastic | SOT270 | −20 to + 70 |
| PCA84C840 | 42 | SDIL | plastic | SOT270 | −20 to + 70 |
| PCA84C841 | 42 | SDIL | plastic | SOT270 | −20 to + 70 |
| PCA84C843 | 42 | SDIL | plastic | SOT270 | −20 to + 70 |
| PCA84C844 | 42 | SDIL | plastic | SOT270 | −20 to + 70 |

**Note**

A 48-lead QFP package will be available shortly.

## 4. PINNING INFORMATION

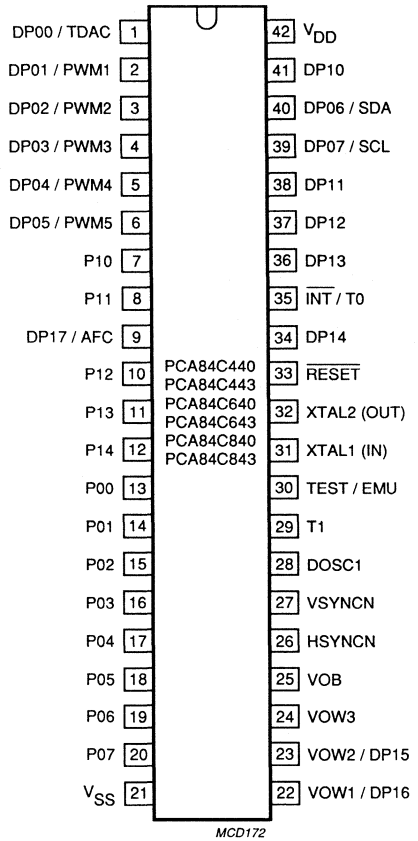



Fig.2 Pinning diagram; 84C440; 84C443; 84C640, 84C643, 84C840 and 84C843

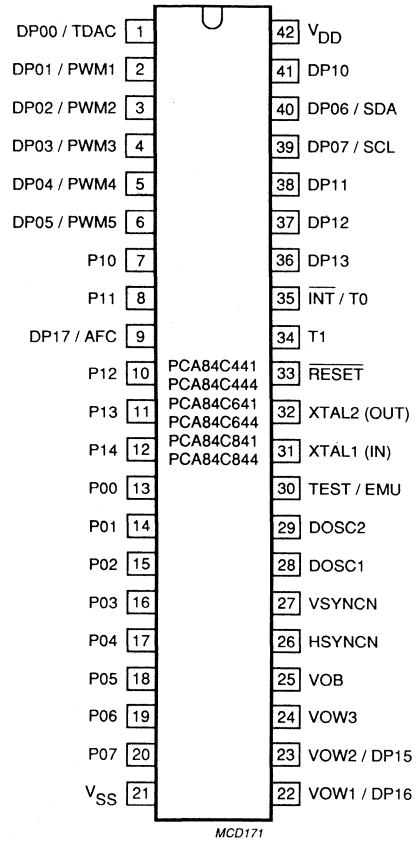



Fig.3 Pinning diagram; 84C441; 84C444; 84C641, 84C644, 84C841 and 84C844

**Table 1** Pin description - 84C440; 84C443; 84C640, 84C643, 84C840 and 84C843

| SYMBOL | PIN | DESCRIPTION |
|---|---|---|
| DP0.0/TDAC | 1 | Derivative Port 0 : quasi-bidirectional I/O line or 14-bit DAC PWM |
| DP0.1 to DP0.5/ PWM1 to PWM5 | 2 to 6 | Derivative Port 1 : quasi-bidirectional I/O lines or 6-bit DAC PWM |
| P1.0 to P1.4 | 7, 8, 10 to 12 | Port 1: quasi-bidirectional I/O lines |
| P0.0 to P0.7 | 13 to 20 | Port 0: quasi-bidirectional I/O port |
| DP1.7/AFC | 9 | Derivative Port 1 : quasi-bidirectional I/O line or comparator input with 3-bit DAC |
| DP0.6/SDA | 40 | Derivative open drain I/O port or I²C-bus data line |
| DP0.7/SCL | 39 | Derivative open drain I/O port or I²C- bus clock line |
| $\overline{INT}$/T0 | 35 | External interrupt or direct testable line |
| DP1.0 to DP1.4 | 41, 38, 37, 36, 34 | Derivative Port 1: quasi-bidirectional I/O lines |
| DP1.5 to DP1.6/ VOW2 to VOW1 | 23, 22 | Derivative Port 1 : quasi-bidirectional I/O lines or character video output |
| $\overline{RESET}$ | 33 | Initialize input, active low |
| XTAL2, XTAL1 | 32, 31 | Oscillator output or input terminal for system clock |
| TEST/EMU | 30 | Control input for testing and emulation mode. Ground for normal operation. |
| T1 | 29 | Direct testable pin and event counter input |
| DOSC1 | 28 | Connection to RC oscillator of OSD clock |
| VSYNCN | 27 | Vertical synchronous signal input |
| HSYNCN | 26 | Horizontal synchronous signal input |
| VOB | 25 | Blanking output |
| VOW3 | 24 | Character video output of OSD |
| $V_{SS}$ | 21 | Ground terminal |
| $V_{DD}$ | 42 | Power supply terminal |

**Table 2** Pin description - 84C441; 84C444; 84C641, 84C644, 84C841 and 84C844

| SYMBOL | PIN | DESCRIPTION |
|---|---|---|
| DP0.0/TDAC | 1 | Derivative Port 0 : quasi-bidirectional I/O line or 14-bit DAC PWM |
| DP0.1 to DP0.5/ PWM1 to PWM5 | 2 to 6 | Derivative Port 1 : quasi-bidirectional I/O lines or 6-bit DAC PWM |
| P1.0 to P1.4 | 7, 8, 10 to 12 | Port 1: quasi-bidirectional I/O lines |
| P0.0 to P0.7 | 13 to 20 | Port 0: quasi-bidirectional I/O port |
| DP1.7/AFC | 9 | Derivative Port 1 : quasi-bidirectional I/O line or comparator input with 3-bit DAC |
| DP0.6/SDA | 40 | Derivative open drain I/O port or I²C-bus data line |
| DP0.7/SCL | 39 | Derivative open drain I/O port or I²C-bus clock line |
| $\overline{INT}$/T0 | 35 | External interrupt or direct testable line |
| DP1.0 to DP1.3 | 41, 38, 37, 36 | Derivative Port 1: quasi-bidirectional I/O lines |
| DP1.5 to DP1.6/ VOW2 to VOW1 | 23, 22 | Derivative Port 1 : quasi-bidirectional I/O lines or character video output |
| $\overline{RESET}$ | 33 | Initialize input, active low |
| XTAL2, XTAL1 | 32, 31 | Oscillator output/input terminals for system clock |
| TEST/EMU | 30 | Control input for testing and emulation mode. Ground for normal operation. |
| T1 | 34 | Direct testable pin and event counter input |
| DOSC1/DOSC2 | 28, 29 | Connections to LC oscillator of OSD clock |
| VSYNCN | 27 | Vertical synchronous signal input |
| HSYNCN | 26 | Horizontal synchronous signal input |
| VOB | 25 | Blanking output |
| VOW3 | 24 | Character video output of OSD |
| $V_{SS}$ | 21 | Ground terminal |
| $V_{DD}$ | 42 | Power supply terminal |

## 5. RESET

The $\overline{\text{RESET}}$ pin is used as an active low input to initialize the microcontroller to a defined state. The Reset configuration is shown in Fig.5.

The Power-on-reset circuit monitors the voltage level of $V_{DD}$. If $V_{DD}$ remains below the internal reference voltage level $V_{ref}$ (typically 3.6 V), the oscillator is inhibited. When $V_{DD}$ rises above $V_{ref}$, the oscillator is released and the internal reset is active for a period of $t_d$ (typically 50 µs).

Three modes of Power-on-reset are possible:

1. An external $\overline{\text{RESET}}$ signal is applied during power-on.
2. If $V_{DD}$ rises above the minimum operation voltage before time period $t_d$ is exceeded, no external components are necessary, (See Fig 6).
3. External components are required if $V_{DD}$ has a slow rise time, such that after the time period ($t_{Vref} + t_d$) has elapsed the supply voltage is still below the minimum operation voltage ($V_{min}$). See Figs 7 and 4. In order to ensure a correct reset operation the time constant RC must be $\geq 8t_{VDD}$.

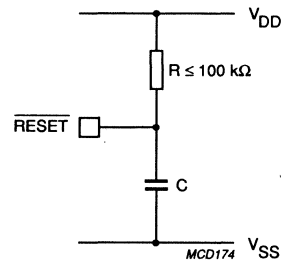



Fig.4 External components for $\overline{\text{RESET}}$ pin.

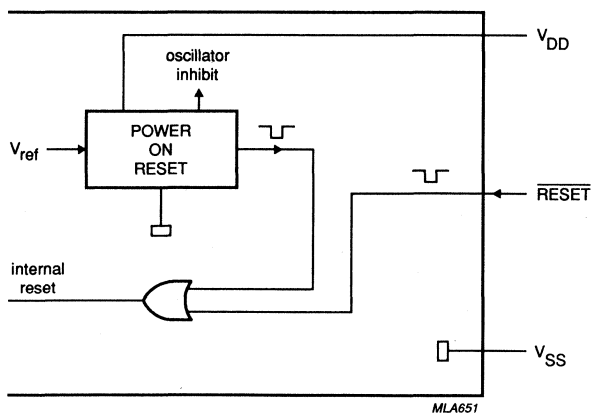



Fig.5 Reset configuration.

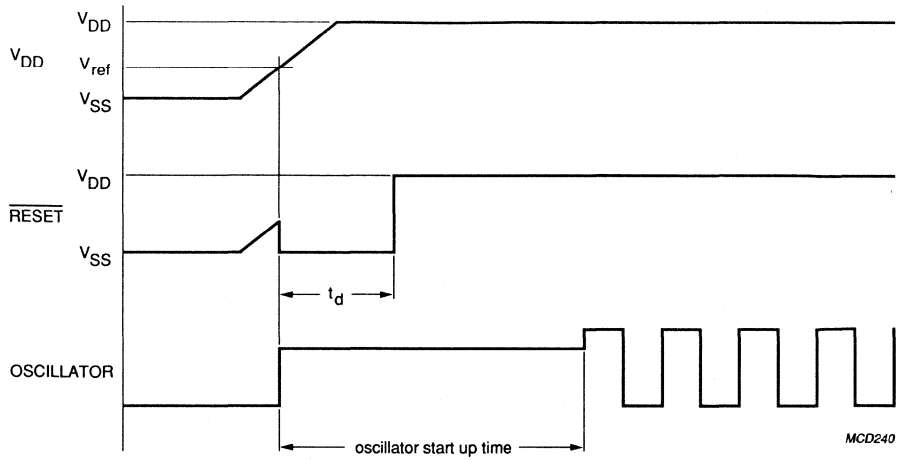



Fig.6 Reset with fast rising $V_{DD}$.

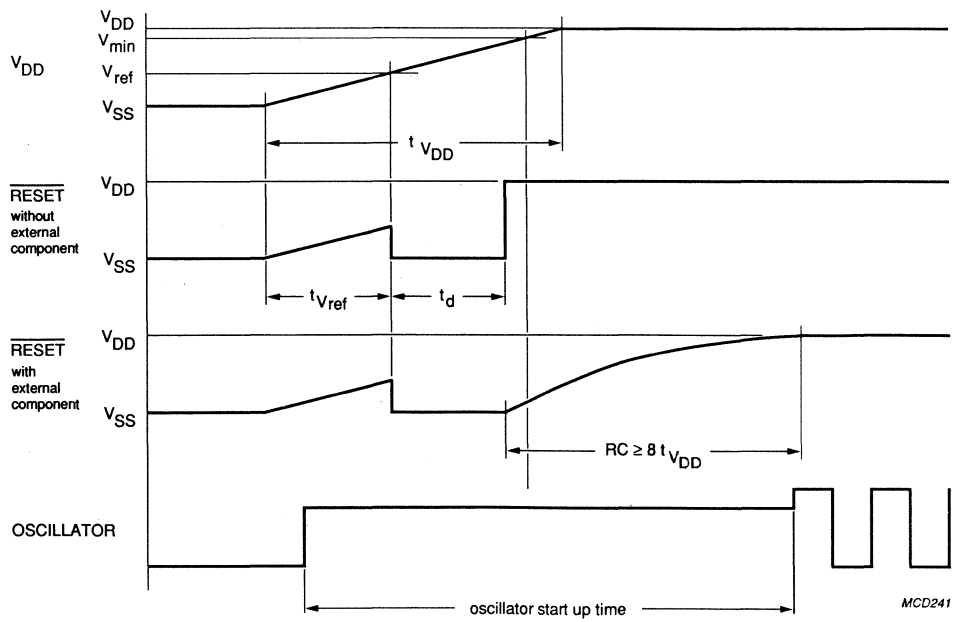



Fig.7 Reset with slow rising $V_{DD}$.

## 6. ANALOG CONTROL; 6-BIT PWM DACs

Five PWM outputs are available for analog control purposes e.g. volume, balance, brightness, saturation etc. The block diagram of a typical 6-bit PWM DAC is shown in Fig.8. Each PWM output can generate pulses of programmable length that have a repetition frequency of $f_{PWM}/64$, (where $f_{PWM} = f_{XTAL}/3$). The polarity of all five PWM outputs is selected by the state of the polarity control bit (P6LVL). Setting P6LVL to a logic 1 inverts all five PWM outputs. If the state of P6LVL is a logic 0, then the PWM outputs are not inverted. The PWM outputs PWM1 to PWM5, share the same pins as the Derivative Port lines DP0.1 to DP0.5. The PWM output function, for individual pins, is selected by setting the relevant PWM enable bit (PWMxE) to '1'. The Derivative Port function is selected by setting the relevant PWMxE bit to '0'.

A DC voltage proportional to the PWM control setting may be obtained by connecting an integrating network to each of the PWM outputs (see Fig.9). The analog value is calculated by multiplying the ratio of the high time and the repetition time of the pulse, by $V_O$.

The high time of any PWM output is: $t_{HIGH} = t_0 \times PWMDL$.

where: $t_0 = 3/f_{XTAL}$ and PWMDL is the decimal value of the contents of the PWM data latch.

The repetition time of any PWM output is: $t_R = t_0 \times 64$

Therefore, the analog output voltage is:
$V_A = (PWMDL/64) \times V_0$

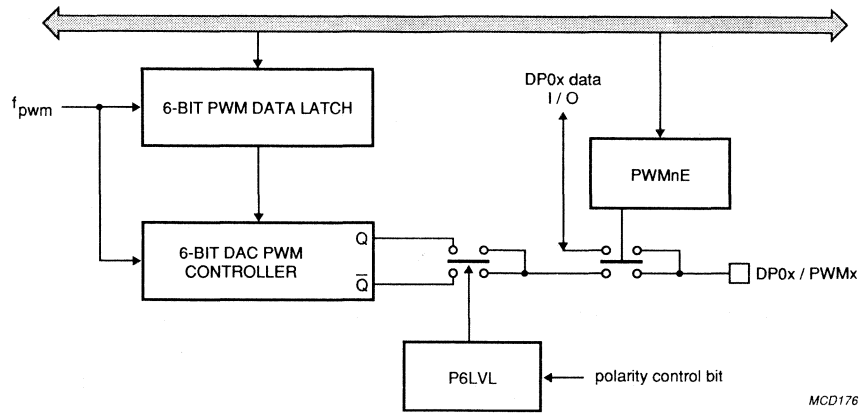



Fig.8 Block diagram; 6-bit PWM DAC.

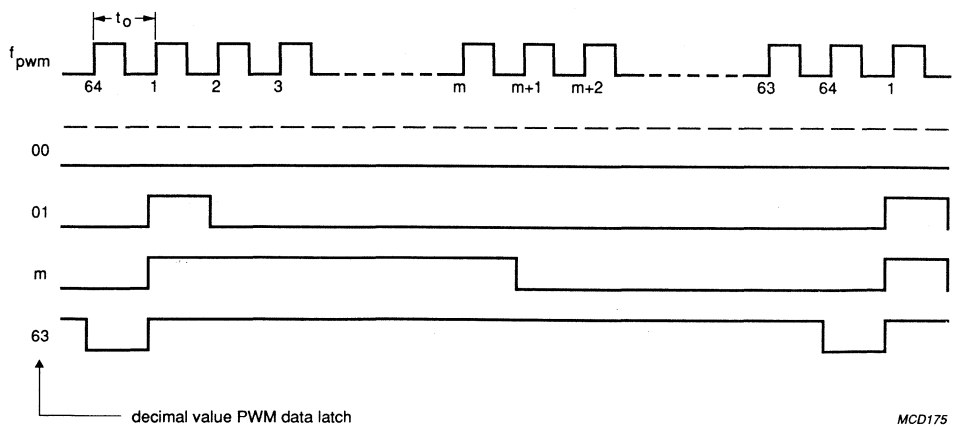



Fig.9 PWM output patterns (P6LVL = 0).

## 7. VST CONTROL; 14-BIT PWM DAC

The PCA84C640 has one 14-bit PWM DAC output with a resolution of 16384 levels for Voltage Synthesized Tuning. The PWM DAC is shown in Fig. 10 and consists of the following:

- two 7-bit DAC interface latches
- one 14-bit DAC data latch
- 14-bit counter
- pulse control

The 14-bit counter is continuously running and is clocked by $f_0$. The period ($t_0$) of the clock is $3/f_{XTAL}$.

The repetition time for one complete cycle of the 14-bit counter is: $t_r = t_0 \times 16384$.

The repetition time for one cycle of the lower 7-bits of the counter is: $t_{sub} = t_0 \times 128$.

Therefore, the number of $t_{sub}$ periods in a complete cycle $t_r$ is: $N = t_0 \times 16384/t_0 \times 128 = 128$

In order to ensure correct operation, data latch VSTH is loaded first and then data latch VSTL. The contents of VSTH are used for coarse adjustment; the contents of VSTL for fine adjustment. At the beginning of the first $t_{sub}$ period following the loading of VSTL, both data latches are downloaded into data latch VSTREG. After the contents of VSTH and VSTL are latched into VSTREG one $t_{sub}$ period is needed to generate the appropriate pulse pattern. To ensure correct DAC conversion two $t_{sub}$ periods should be allowed before beginning the next sequence.

### 7.1 Coarse adjustment

The coarse adjustment output (OUT1) is reset LOW (inactive) at the start of each $t_{sub}$ period. It will remain LOW until the time $t_0$ x (VSTH + 1) has elapsed and then will go HIGH and remain so until the next $t_{sub}$ period starts.

### 7.2 Fine adjustment

Fine adjustment is achieved by generating additional pulses at the start of particular sub-periods. These pulses have a width of $t_0$. The subperiod in which a pulse is added is determined by the contents of VSTL data latch. Table 3 gives the subperiod numbers at the start of which an additional pulse is generated, depending on the bit in VSTL being '0'. When more than one bit is '0' a combination of additional pulses are generated. For example if VSTL = 1111010 additional pulses will be given in subperiods 16, 48, 64, 80 and 112; this is illustrated in Fig.12.

If VSTH = 0011101, VSTL = 1111010 and P14LVL = 0, then the TDAC output is as shown in Fig. 13.

**Table 3** Additional pulse distribution

| LOWER 7 BITS (VSTL) | ADDITIONAL PULSE IN SUBPERIODS $t_{SUBN}$ |
|---|---|
| 111 1110 | 64 |
| 111 1101 | 32, 96 |
| 111 1011 | 16, 48, 80, 112 |
| 111 0111 | 8, 24, 40, 56, 72, 88, 104, 120 |
| 110 1111 | 4, 12, 20, 28, 36, 44, 52, 60 .... 116, 124 |
| 101 1111 | 2, 6, 10, 14, 18, 22, 26, 30, .... 122, 126 |
| 011 1111 | 1, 3, 5, 7, 9, 11, 13, 15, 17, .... 125, 127 |

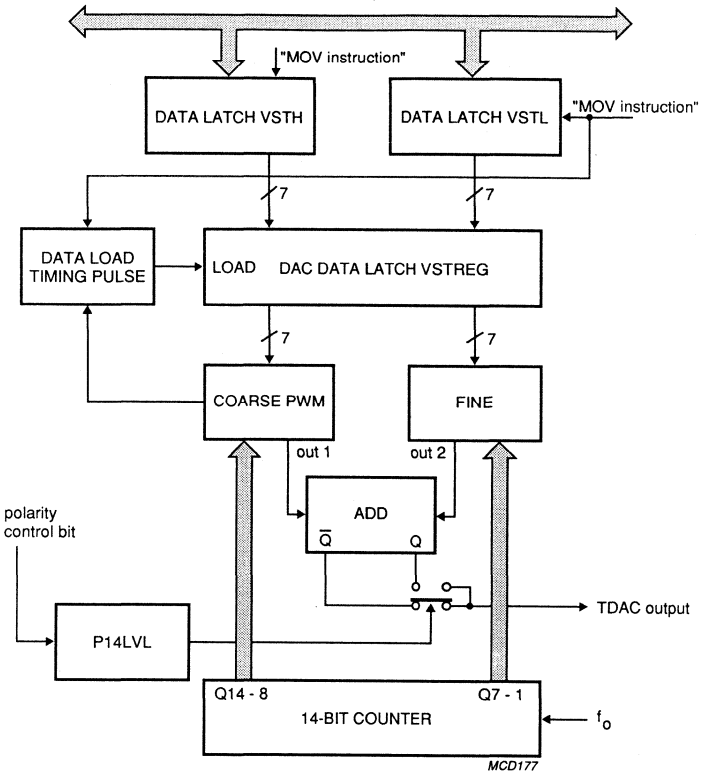



Fig.10 Block diagram: 14-bit PWM DAC.

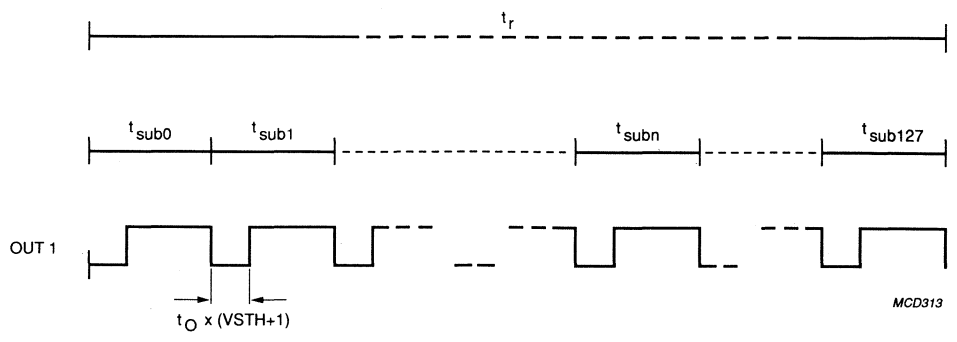



Fig.11 Coarse adjustment output (Out1).

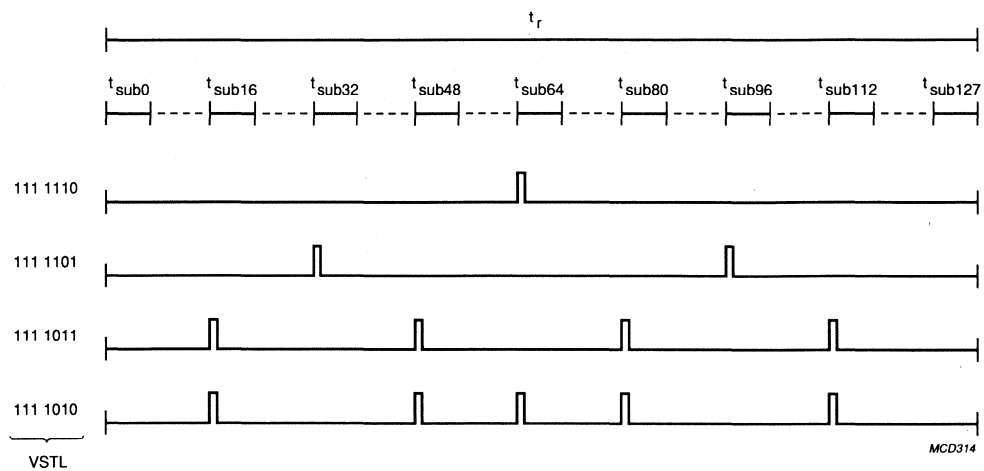



Fig.12 Fine adjustment output (Out2).

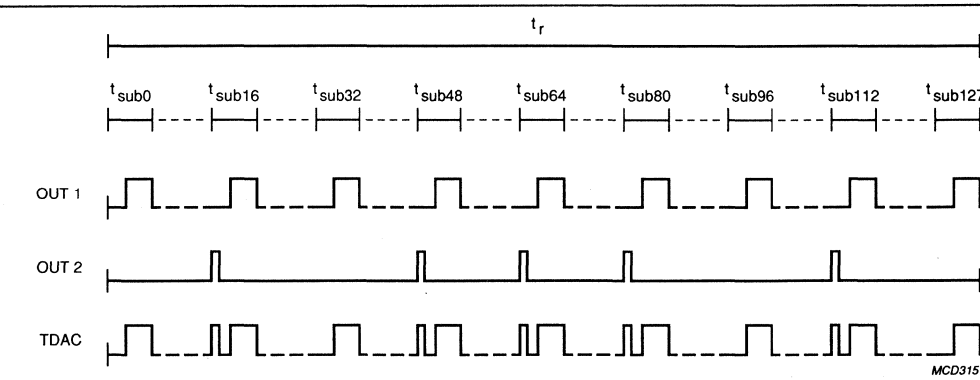



Fig.13 TDAC output.

## 8. AFC INPUT

The AFC input is used to measure the level of the Automatic Frequency Control signal. This is achieved by comparing the AFC input signal with the output of a 3-bit DAC as shown in Fig.14. DAC analog switches select one of 8 resistor taps connected between $V_{DD}$ and $V_{SS}$. Consequently, eight different voltages may be selected (see Table 4). The compare signal AFCC, can be tested to determine whether the AFC input is higher or lower than the DAC level. The AFC input shares the same pin as the Derivative Port line DP1.7. The AFC function is selected by setting the enable bit AFCE to a logic 1. The Derivative Port function is selected by setting the AFCE bit to a logic 0.

**Table 4** Selection of $V_{ref}$

| AFC2 | AFC1 | AFC0 | $V_{ref}$ | $V_{ref}$ ($V_{DD}$ = 5.0 V) |
|---|---|---|---|---|
| 0 | 0 | 0 | $V_{DD}$ x 0.125 | 0.625 V |
| 0 | 0 | 1 | $V_{DD}$ x 0.250 | 1.250 V |
| 0 | 1 | 0 | $V_{DD}$ x 0.375 | 1.875 V |
| 0 | 1 | 1 | $V_{DD}$ x 0.500 | 2.500 V |
| 1 | 0 | 0 | $V_{DD}$ x 0.625 | 3.125 V |
| 1 | 0 | 1 | $V_{DD}$ x 0.750 | 3.750 V |
| 1 | 1 | 0 | $V_{DD}$ x 0.875 | 4.375 V |
| 1 | 1 | 1 | $V_{DD}$ | 5.000 V |

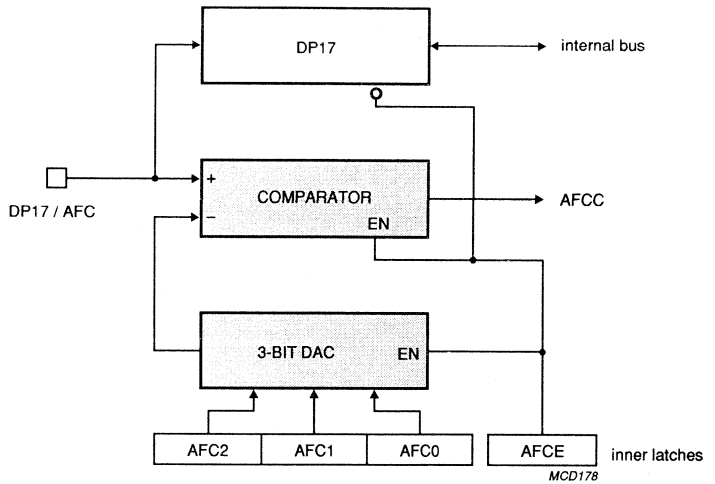



Fig.14 AFC circuit.

## 9. I/O

Each parallel I/O port line may be individually configured using one of three possible I/O mask options. The three I/O mask options are specified below:-

Option 1: Standard port with switched pull-up current source (see Fig. 15).

Option 2: Open drain (see Fig. 16).

Option 3: Push-pull (output only, see Fig. 17).

Table 5 specifies the possible port option list. When these devices are used for emulation purposes, in order to match the piggy back device provided it is recommended that the port options listed in Table 6 are used.

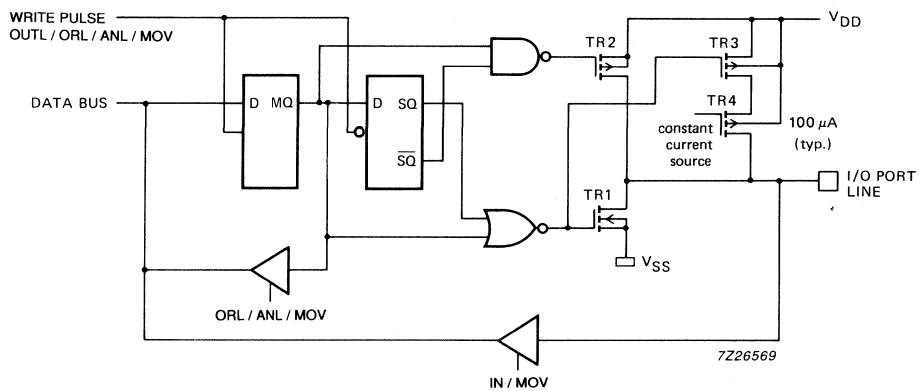



Fig.15 Standard output with switched pull-up current source (Option 1).

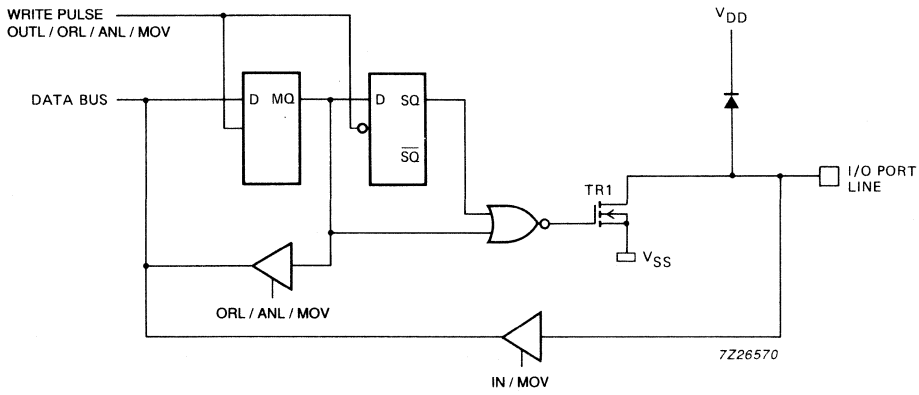



Fig.16 Open drain type I/O (Option 2).

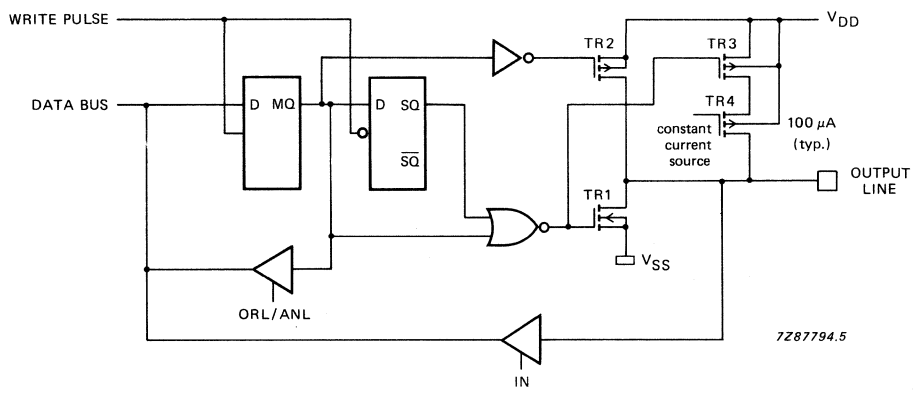



Fig.17 Push-pull type output (Option 3).

**Table 5** User mask programmable port option list

| PORT | PIN |
|---|---|
| P0.0 | 13 |
| P0.1 | 14 |
| P0.2 | 15 |
| P0.3 | 16 |
| P0.4 | 17 |
| P0.5 | 18 |
| P0.6 | 19 |
| P0.7 | 20 |
| P1.0 | 7 |
| P1.1 | 8 |
| P1.2 | 10 |
| P1.3 | 11 |
| P1.4 | 12 |
| DP0.0 | 1 |
| DP0.1 | 2 |
| DP0.2 | 3 |
| DP0.3 | 4 |
| DP0.4 | 5 |
| DP0.5 | 6 |
| DP0.6 | 40 |
| DP0.7 | 39 |
| DP1.0 | 41 |
| DP1.1 | 38 |
| DP1.2 | 37 |
| DP1.3 | 36 |
| DP1.4 | 34 |
| DP1.5 | 23 |
| DP1.6 | 22 |
| DP1.7 | 9 |
| VOB | 25. |
| VOW3 | 24. |

**Table 6** Port options for the 84C640 in emulation mode

| PORT | PIN | OPTION |
|---|---|---|
| P0.0 | 13 | 1 |
| P0.1 | 14 | 1 |
| P0.2 | 15 | 1 |
| P0.3 | 16 | 1 |
| P0.4 | 17 | 1 |
| P0.5 | 18 | 1 |
| P0.6 | 19 | 1 |
| P0.7 | 20 | 1 |
| P1.0 | 7 | 1 |
| P1.1 | 8 | 1 |
| P1.2 | 10 | 1 |
| P1.3 | 11 | 1 |
| P1.4 | 12 | 1 |
| DP0.0 | 1 | |
| DP0.1 | 2 | |
| DP0.2 | 3 | |
| DP0.3 | 4 | |
| DP0.4 | 5 | |
| DP0.5 | 6 | |
| DP0.6 | 40 | 2 S |
| DP0.7 | 39 | 2 S |
| DP1.0 | 41 | |
| DP1.1 | 38 | |
| DP1.2 | 37 | |
| DP1.3 | 36 | |
| DP1.4 | 34 | |
| DP1.5 | 23 | |
| DP1.6 | 22 | |
| DP1.7 | 9 | |
| VOB | 25 | |
| VOW3 | 24 | |

**Notes**

1. DP1.4 available only with the 84C440, 84C443, 84C640, 84C643, 84C840 and 84C843.
2. VOB and VOW3 pins; Option 1 not available.
3. Each pin can be configured to a High (S) or Low (R) state after power-on-reset. The required state of each pin is therefore specified by R or S.

## 10. ON SCREEN DISPLAY

### 10.1 Features

- Display format: 2 rows x 16 characters
- Software controlled vertical and horizontal display position
- 64 different characters in ROM - mask programmable
- Black box background
- Four programmable display character sizes
- Four programmable character dot matrix sizes: 6 x 9, 8 x 9, 6 x 13 or 8 x 13
- Half-dot rounding - for whole screen
- 4 from 7 colours possible on screen
- LC oscillator for on screen display function with the 84C441, 84C444, 84C641, 84C644, 84C841 and 84C844.
- RC oscillator for on screen display function with the 84C440, 84C443, 84C640, 84C643, 84C840 and 84C843.

HORIZONTAL DISPLAY POSITION CONTROL

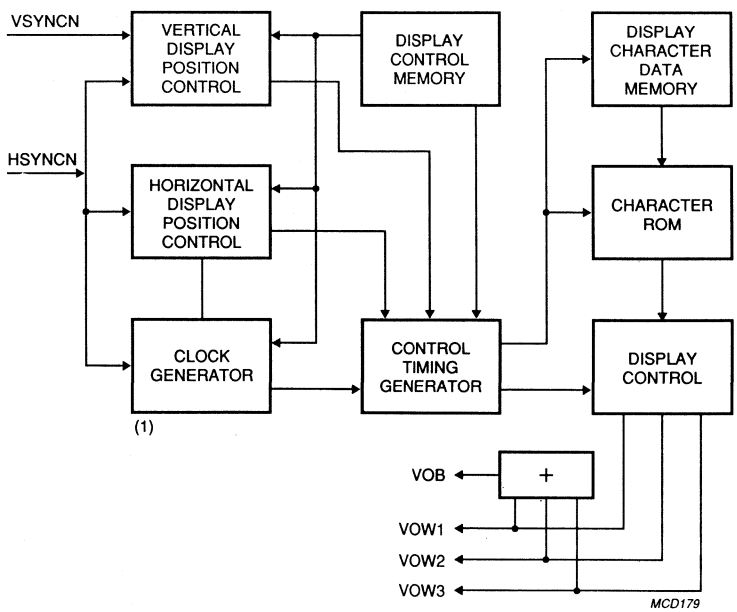
The horizontal position counter is incremented every OSD cycle after the programmed level of HSYNCN occurs at the HSYNCN pin. The counter is reset when the opposite polarity of the HSYNCN pulse is reached.

VERTICAL DISPLAY POSITION CONTROL

The vertical position counter is incremented every HSYNCN cycle and is reset by the VSYNCN signal.

CLOCK GENERATOR

The clock generator consists of an LC oscillator; the external LC network being connected across pins 28 and 29, see Fig.20. (The 84C440, 84C443, 84C640, 84C643, 84C840 and 84C843 use an RC oscillator connected between pin 28 and $V_{SS}$, see Fig.19). The OSD oscillator must be externally adjusted to the desired frequency (decreasing the OSD frequency gives broader characters). The oscillator is triggered on the trailing edge of HSYNC when the OSD logic is enabled and stops on the following leading edge of HSYNC.




(1) See Figs 19 and 20 for connection of external components.

Fig.18 OSD block diagram.

### 10.2 Display data registers

The display data registers consists of a group of 32 derivative registers located at addresses 20H to 3FH inclusive. The 16 registers located in memory at addresses 20H to 2FH contain display data for the first line. The remaining 16 registers located in memory at addresses 30H to 3FH, contain display data for the second line. At power-up the contents of the display data registers are undefined. The format of each Display data register is shown in Fig.21.

CC1 and CC0 - COLOUR CODE

The state of these two bits enable individual characters to be displayed in one of four colours.

MD0 to MD5 - CHARACTER CODE

The character set is stored in ROM and consists of 64 different characters. The selection of each character is dependent on the state of the 6 bits, MD0 to MD5.

### 10.3 Display control registers

The display control registers consists of a group of 6 derivative registers located at addresses 40H to 45H inclusive. (See Table 7 Display control registers). Each register may be read from or written to. After a reset operation the contents of the display control registers are zero.

**Derivative register OSDCA**

This register resides at address 40H.

HLVL - HORIZONTAL SYNCHRONOUS SIGNAL LEVEL

This bit selects the active level of the HSYNCN input signal. When HLVL is a logic 1, HSYNCN is active HIGH. When HLVL is a logic 0, HSYNCN is active LOW. See Fig.23.

VLVL - VERTICAL SYNCHRONOUS SIGNAL LEVEL

This bit selects the active level of the VSYNCN input signal. When VLVL is a logic 1, VSYNCN is active HIGH. When VLVL is a logic 0, VSYNCN is active LOW. See Fig.23.

STBY - STANDBY

This bit is used to enable or disable the OSD facility. If the STBY bit is a logic 1, the OSD oscillator is disabled. If the STBY bit is a logic 0, the OSD oscillator is enabled and the OSD facility is available. Before the oscillation frequency can be adjusted HSYNCN must be HIGH (if HLVL = 1). Oscillation stops by setting the HSYNCN pin LOW when HLVL = 1.

ROUND - CHARACTER ROUNDING CONTROL

The rounding function generates half dots where the corners of two dots meet. See Figs 24 and 25. The rounding function also works with multiple cell characters. When the Round bit is a logic 1, the rounding function is selected. When the Round bit is a logic 0, the rounding function is disabled.

RBLK - RASTER BLANKING CONTROL

When the RBLK bit is a logic 1, the VOB output is driven high to display the OSD characters on a blank screen. When the RBLK bit is a logic 0, the VOB output returns to its normal output state on the trailing edge of VSYNCN. See Fig.26.

CC34, CC24 and CC14 - CHARACTER COLOUR BITS

These bits are used for colour selection purposes. See Tables 9 to 14.

**Table 7** Display control registers

| ADDRESS | REGISTER | BIT 7 | BIT 6 | BIT 5 | BIT 4 | BIT 3 | BIT 2 | BIT 1 | BIT 0 |
|---|---|---|---|---|---|---|---|---|---|
| 40H | OSDCA | CC34 | CC24 | CC14 | RBLK | ROUND | STBY | VLVL | HLVL |
| 41H | LINE 0A | SZ01 | SZ00 | VP05 | VP04 | VP03 | VP02 | VP01 | VP00 |
| 42H | LINE 0B | BLK0 | VB0 | HP05 | HP04 | HP03 | HP02 | HP01 | HP00 |
| 43H | OSDCB | CDTW | CDTH | CC33 | CC23 | CC32 | CC12 | CC21 | CC11 |
| 44H | LINE 1A | SZ11 | SZ10 | VP15 | VP14 | VP13 | VP12 | VP11 | VP10 |
| 45H | LINE 1B | BLK1 | VB1 | HP15 | HP14 | HP13 | HP12 | HP11 | HP10 |

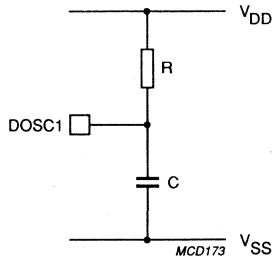



Fig.19 RC oscillator (84C440; 84C443; 84C640; 84C643; 84C840 and 84C843).

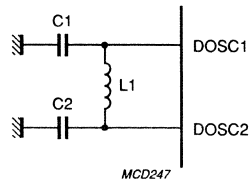



Fig.20 LC oscillator (84C441; 84C444; 84C641; 84C644; 84C841 and 84C844).

| CC1 (MSB) | CC0 | MD5 | MD4 | MD3 | MD2 | MD1 | MD0 (LSB) |
|---|---|---|---|---|---|---|---|

Fig.21 Display data register.

| CC34 (MSB) | CC24 | CC14 | RBLK | ROUND | STBY | VLVL | HLVL (LSB) |
|---|---|---|---|---|---|---|---|

Fig.22 Derivative register OSDCA.

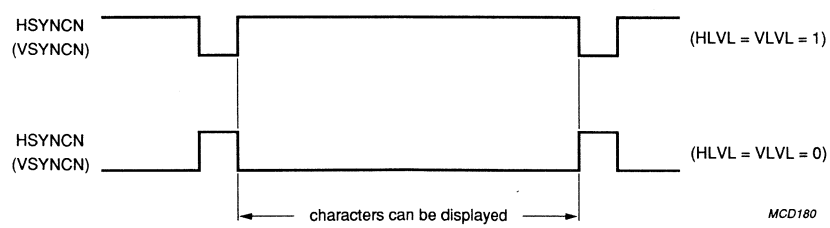



Fig.23 VSYNCN and HSYNCN active level.

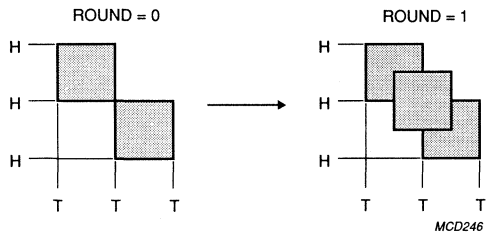



Fig.24 Rounding function.

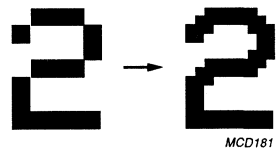



Fig.25 Rounding effect.

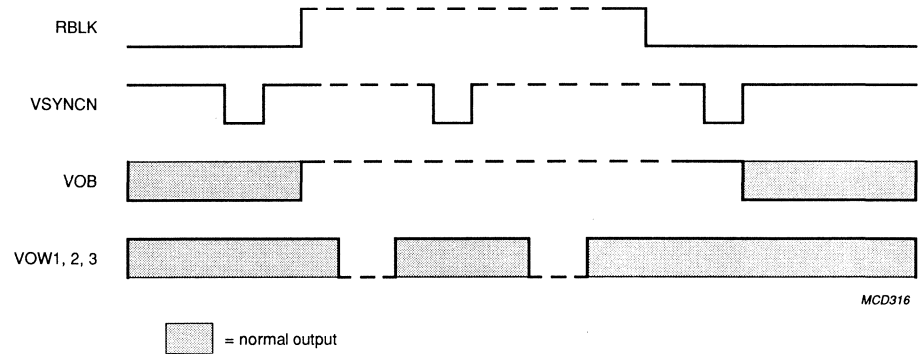



Fig.26 Raster blanking timing RLBK.

### Derivative register LINE 0A

This register resides at address 41H. Its contents determine the character size and vertical position of the first row of display.

SZ00 and SZ01 - CHARACTER SIZE

The state of these two bits enable one of four possible character sizes to be selected for Row 0. Character sizes include background.

VP00 to VP05 - VERTICAL POSITION CONTROL

The vertical position of the first display row is selected by the state of the 6 bits, VP00 to VP05. The line number of the vertical start position for Row 0 is 4 x (decimal value of bits VP00 to VP05).

Row 0 and Row 1 must not overlap each other and therefore VP1 must be greater than or equal to (VP0 + H0), see Fig. 28. H0 is the character height of the characters in Row 0 and is a function of the number of dots per character and the state of the Size control bits SZ00 and SZ01. The four possible character heights are shown in Table 8.

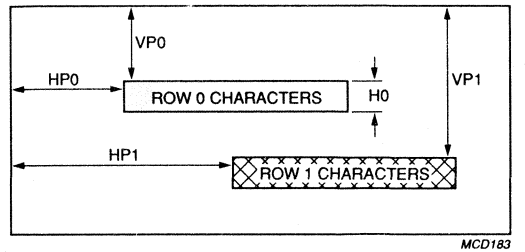



Fig.28 Display position.

| SZ01 (MSB) | SZ00 | VP05 | VP04 | VP03 | VP02 | VP01 | VP00 (LSB) |
|---|---|---|---|---|---|---|---|

Fig.27 Derivative register LINE 0A.

**Table 8** OSD character sizes

| DATA | | CHARACTER SIZE | | | | DOT MATRIX POINT | |
|---|---|---|---|---|---|---|---|
| SZx1 | SZx0 | VERTICAL | | HORIZONTAL | | VERTICAL | HORIZONTAL |
| | | 9D | 13D | 6D | 8D | | |
| 0 | 0 | 18H | 26H | 12T | 16T | 2H | 2T |
| 0 | 1 | 36H | 52H | 24T | 32T | 4H | 4T |
| 1 | 0 | 54H | 78H | 36T | 48T | 6H | 6T |
| 1 | 1 | 72H | 104H | 48T | 64T | 8H | 8T |

**Note**

H denotes one horizontal line, T denotes one OSDC clock period and D denotes dots per character width/height.

**Derivative register LINE 0B**

The Derivative register LINE 0B resides at address 42H. Its contents determine the horizontal position of the first row of the display and whether the background and blanking functions are selected.

BLK0 - BLANKING

This bit enables or disables the character display. When BLK0 is a logic 1, the outputs VOW1, VOW2, VOW3 and VOB are disabled; consequently no characters are displayed. When BLK0 is a logic 1, the outputs VOW1, VOW2, VOW3 and VOB are enabled and characters are displayed.

VB0 - BACKGROUND

The state of the VB0 bit determines whether the background display is selected or not. If the VB0 bit is set to a logic 1, the characters in this line are displayed with background. If the VB0 bit is logic 0 the background is disabled and only the character is displayed.

HP01 to HP05 - HORIZONTAL POSITION CONTROL

These 6 bits determine the start position of the first display row. The horizontal position control is only active during OSDC clock cycles. (See Fig. 28).

The horizontal start position (HP) of Row 0 may be calculated using:

HP = 4 x (decimal value of bits HP00 to HP05) + 5 x (OSD C periods)

The decimal value of bits HP00 to HP05 must be greater than 10. Therefore, HP ≥ 45 (OSDC clock pulses).

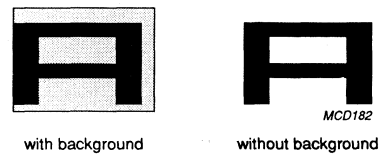



Fig.30 Background versus no background.

| BLK0 (MSB) | VB0 | HP05 | HP04 | HP03 | HP02 | HP01 | HP00 (LSB) |
|---|---|---|---|---|---|---|---|

Fig.29 Derivative register LINE 0B.

**Derivative register OSDCB**

The OSDCB register resides at address 43H. The contents of this register permit the size of the dot matrix grid to be selected and also enable four colours from a possible seven to be chosen for the display.

CDTW - CHARACTER DOT WIDTH CONTROL

The state of this bit determines the dot width of the character. When the CDTW bit is set to a logic 1, the character width is 6 dots. When the CDTW bit is set to a logic 0, the character width is 8 dots.

CDTH - CHARACTER DOT HEIGHT CONTROL

The state of this bit determines the dot height of the character. When the CDTH bit is set to a logic 1, the character height is 13 dots. When the CDTH bit is set to a logic 0, the character height is 9 dots.

CCXX - COLOUR CONTROL BITS

In every VSYNCN cycle one screen can select any 4 colours from 7 and in addition a blank or black screen. Colour selection is achieved using bits CC34, CC24 and CC14 in the OSDCA register; bits CC33, CC23, CC32, CC12, CC21, and CC11 in the OSDCB register and bits CC1 and CC0 from the display data registers. (See Table 9). Bits CC1X control VOW1 (red), bits CC2X control VOW2 (green) and bits CC3X control VOW3 (blue). Combinations of CCV2 and CCV3 control VOW1, VOW2 and VOW3.

In this way every combination of four colours can be made (black and white can not be displayed at the same time). Table 11 shows the possible combinations. The user may choose one colour out of each block. For example see Table 10.

**Derivative register LINE 1A**

This register resides at address 44H. Its contents determine the character size and vertical position of the second row of display.

SZ10 and SZ11 - CHARACTER SIZE

The state of these two bits enable one of four possible character sizes to be selected for Row 1. Character sizes include background. See Table 8 for character size selection.

VP10 to VP15 - VERTICAL POSITION CONTROL

The vertical position of the second display row is selected by the state of the 6 bits, VP10 to VP15. The line number of the vertical start position for Row 1 is 4 x (decimal value of bits VP15 to VP10).

Row 0 and Row 1 must not overlap each other and therefore VP1 must be greater than or equal to (VP0 + H0), see Fig. 28. H0 is the character height of the characters in Row 0 and is a function of the number of dots per character and the state of the size control bits SZ00 and SZ01. The four possible character heights are shown in Table 8.

| CDTW (MSB) | CDTH | CC33 | CC23 | CC32 | CC12 | CC21 | CC11 (LSB) |
|---|---|---|---|---|---|---|---|

Fig.31 Derivative register OSDCB

**Table 9** Character colour control

| CC1 | CC0 | VOW1 | VOW2 | VOW3 |
|---|---|---|---|---|
| 0 | 0 | CC11 | CC21 | $\overline{CC11 + CC21}$ |
| 0 | 1 | CC12 | $\overline{CC12 + CC32}$ | CC32 |
| 1 | 0 | $\overline{CC23 + CC33}$ | CC23 | CC33 |
| 1 | 1 | CC14 | CC24 | CC34 |

**Table 10** Example of selecting colours by programming CCxx

| CHARACTER DATA | CHARACTER OUTPUT PINS | | |
|---|---|---|---|
| VOW1 (Red) | VOW2 (Green) | VOW3 (Blue) | COLOUR |
| 0 | 0 | 0 | black |
| 0 | 0 | 1 | blue (2) |
| 0 | 1 | 0 | green (3) |
| 0 | 1 | 1 | cyan |
| 1 | 0 | 0 | red |
| 1 | 0 | 1 | magenta (4) |
| 1 | 1 | 0 | yellow (1) |
| 1 | 1 | 1 | white |

**Notes**

If VOW1 = Red, VOW2 = Green, VOW3 = Blue, then using the bit combination CC11:1, CC21:1, CC12:0, CC32:1, CC23:1, CC33:0, CC14:1, CC24:0, CC34:1 the following colours may be selected.

1. Yellow; (CC1, CC0 = 00)
2. Blue; (CC1, CC0 = 01)
3. Green; (CC1, CC0 = 10)
4. Magenta; (CC1, CC0 = 11)

| SZ11 (MSB) | SZ10 | VP15 | VP14 | VP13 | VP12 | VP11 | VP10 (LSB) |
|---|---|---|---|---|---|---|---|

Fig.32 Derivative register LINE 1A.

### Derivative register LINE 1B

The Derivative register LINE 1B resides at address 45H. Its contents determine the horizontal position of the second line of the display (Line 1) and whether the background and blanking functions are selected.

BLK1 - BLANKING

This bit enables or disables the character display. When BLK1 is a logic 1, the outputs VOW1, VOW2, VOW3 and VOB are disabled; consequently no characters are displayed. When BLK1 is a logic 1, the outputs VOW1, VOW2, VOW3 and VOB are enabled and characters are displayed.

VB1 - BACKGROUND

The state of the VB1 bit determines whether the background display is selected or not. If the VB1 bit is set to a logic 1, the characters in this line are displayed with background. If the VB1 bit is logic 0, the background is disabled and only the character is displayed. The visual effect of background versus no background is shown in Fig.30.

HP11 to HP15 - HORIZONTAL POSITION CONTROL

These 6 bits determine the start position of the second display line (Line 1). The horizontal position control is only active during OSDC clock cycles. (See Fig. 28). The horizontal start position (HP) of Line 1 may be calculated using:

HP = 4 x (decimal value of bits HP10 to HP15)) + 5 x (OSD C periods)

The decimal value of bits HP10 to HP15 must be greater than 10. Therefore, HP ≥ 45 (OSDC clock pulses)

### Character ROM

Character ROM contains the dot character fonts. 13 x 8 dots are reserved for each character, regardless of the dot matrix size actually selected. The dot matrix grid is shown in Fig.34.

The document 'How to prepare the character ROM code for the OSD part of the PCA84C640' is available on request.

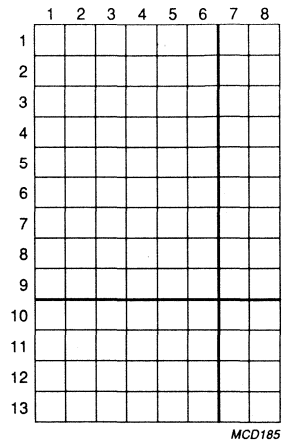



Fig.34 Character ROM.

| BLK1 (MSB) | VB1 | HP15 | HP14 | HP13 | HP12 | HP11 | HP10 (LSB) |
|---|---|---|---|---|---|---|---|

Fig.33 Derivative register LINE 1B.

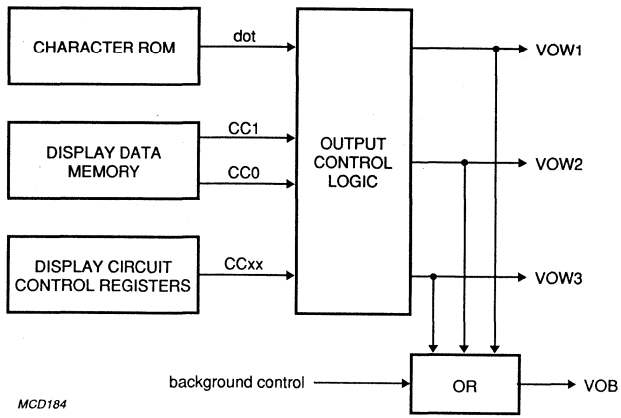



Fig.35 Colour control.

**Table 11** Possible colour combinations

| CHARACTER DATA | | VOW1 | VOW2 | VOW3 | COLOUR |
|---|---|---|---|---|---|
| CC1 | CC0 | CC11 | CC21 | $\overline{CC11+CC21}$ | |
| 0 | 0 | 0 | 0 | 1 | blue |
| 0 | 0 | 0 | 1 | 0 | green |
| 0 | 0 | 1 | 0 | 0 | red |
| 0 | 0 | 1 | 1 | 0 | yellow |

**Table 12** Possible colour combinations

| CHARACTER DATA | | VOW1 | VOW2 | VOW3 | COLOUR |
|---|---|---|---|---|---|
| CC1 | CC0 | CC12 | $\overline{CC12+CC32}$ | CC32 | |
| 0 | 1 | 0 | 0 | 1 | blue |
| 0 | 1 | 0 | 1 | 0 | green |
| 0 | 1 | 1 | 0 | 0 | red |
| 0 | 1 | 1 | 0 | 1 | magenta |

**Table 13** Possible colour combinations

| CHARACTER DATA | | VOW1 | VOW2 | VOW3 | COLOUR |
|---|---|---|---|---|---|
| CC1 | CC0 | $\overline{CC23+CC33}$ | CC23 | CC33 | |
| 1 | 0 | 0 | 0 | 1 | blue |
| 1 | 0 | 0 | 1 | 0 | green |
| 1 | 0 | 0 | 1 | 1 | cyan |
| 1 | 0 | 1 | 0 | 0 | red |

**Table 14** Possible colour combinations

| CHARACTER DATA | | VOW1 | VOW2 | VOW3 | COLOUR |
|---|---|---|---|---|---|
| CC1 | CC0 | CC14 | CC24 | CC34 | |
| 1 | 1 | 0 | 0 | 0 | black |
| 1 | 1 | 0 | 0 | 1 | blue |
| 1 | 1 | 0 | 1 | 0 | green |
| 1 | 1 | 0 | 1 | 1 | cyan |
| 1 | 1 | 1 | 0 | 0 | red |
| 1 | 1 | 1 | 0 | 1 | magenta |
| 1 | 1 | 1 | 1 | 0 | yellow |
| 1 | 1 | 1 | 1 | 1 | white |

## 11. EMULATION MODE

The emulation mode configuration is shown in Fig.36.

In the emulation mode configuration the PCA84C640's CPU is disabled and only its derivative logic is active. The device is controlled by the PCF84C00 bond-out chip. The PCA84C640's two derivative ports act as additional ports for the PCF84C00. The interaction between the two devices is as follows:

1. During the first machine cycle the PCF84C00 fetches an instruction from EPROM and then decodes that instruction.
2. During the second machine cycle the PCF84C00 executes the decoded instruction. If the instruction is related to the derivative ports then DXALE, DXRDN and/or DXWRN become active and the PCA84C640 operates as a peripheral of the PCF84C00.
3. Depending on the type of instruction executed during the second machine cycle the following data transfer happens:
   (a) During TS1 data from the EPROM is available on P0.0 to P0.7 which is then available on IB0.0 of the PCF84C00.
   (b) During TS4 data from the PCA84C640 can be transferred to the PCF84C00.
   (c) During TS6 data from the PCF84C00 can be transferred to the PCA84C640.

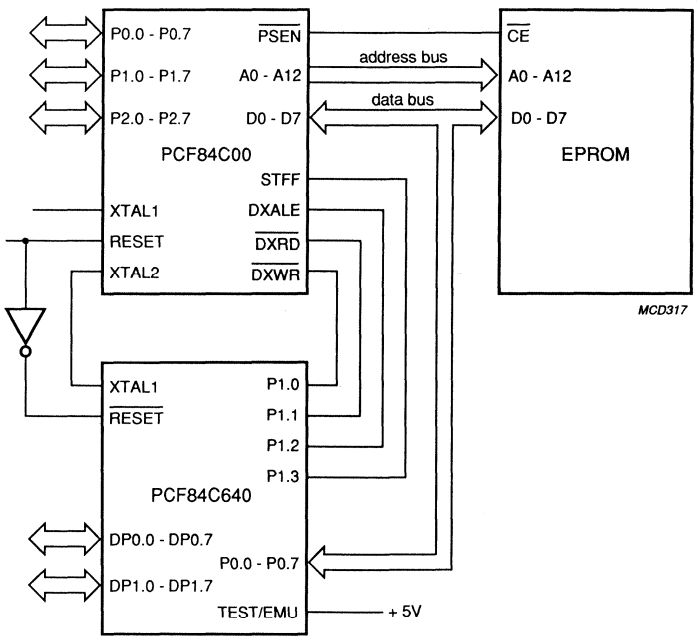



Fig.36 Emulation mode configuration.

## 12. DIFFERENCES BETWEEN THE 84C44X, 84C64X AND 84C84X

**Table 15** Differences between the 84C440; 84C441; 84C443 and 84C444

| FEATURE | 84C440 | 84C441 | 84C443 | 84C444 |
|---|---|---|---|---|
| ROM | 4 Kbytes | 4 Kbytes | 4 Kbytes | 4 Kbytes |
| RAM | 128 bytes | 128 bytes | 128 bytes | 128 bytes |
| OSD oscillator | RC | LC | RC | LC |
| general purpose I/O lines | 18 | 17 | 18 | 17 |
| pin assignment | pin 34 = DP1.4<br>pin 29 = T1 | pin 34 = T1<br>pin 29 = DOSC2 | pin 34 = DP1.4<br>pin 29 = T1 | pin 34 = T1<br>pin 29 = DOSC2 |
| Register DP1 - pin; bit DP1.4 | available | not available | available | not available |
| Register DP1 - latch; bit DP1.4 | available | not available | available | not available |
| port line DP14 | available | not available | available | not available |
| I²C interface | available | available | not available | not available |

**Table 16** Differences between the 84C640; 84C641; 84C643 and 84C644

| FEATURE | 84C640 | 84C641 | 84C643 | 84C644 |
|---|---|---|---|---|
| ROM | 6 Kbytes | 6 Kbytes | 6 Kbytes | 6 Kbytes |
| RAM | 128 bytes | 128 bytes | 128 bytes | 128 bytes |
| OSD oscillator | RC | LC | RC | LC |
| general purpose I/O lines | 18 | 17 | 18 | 17 |
| pin assignment | pin 34 = DP1.4<br>pin 29 = T1 | pin 34 = T1<br>pin 29 = DOSC2 | pin 34 = DP1.4<br>pin 29 = T1 | pin 34 = T1<br>pin 29 = DOSC2 |
| Register DP1 - pin; bit DP1.4 | available | not available | available | not available |
| Register DP1 - latch; bit DP1.4 | available | not available | available | not available |
| port line DP14 | available | not available | available | not available |
| I²C interface | available | available | not available | not available |

**Table 17** Differences between the 84C840; 84C841; 84C843 and 84C844

| FEATURE | 84C840 | 84C841 | 84C843 | 84C844 |
|---|---|---|---|---|
| ROM | 8 Kbytes | 8 Kbytes | 8 Kbytes | 8 Kbytes |
| RAM | 192 bytes | 192 bytes | 192 bytes | 192 bytes |
| OSD oscillator | RC | LC | RC | LC |
| general purpose I/O lines | 18 | 17 | 18 | 17 |
| pin assignment | pin 34 = DP1.4<br>pin 29 = T1 | pin 34 = T1<br>pin 29 = DOSC2 | pin 34 = DP1.4<br>pin 29 = T1 | pin 34 = T1<br>pin 29 = DOSC2 |
| Register DP1 - pin; bit DP1.4 | available | not available | available | not available |
| Register DP1 - latch; bit DP1.4 | available | not available | available | not available |
| port line DP14 | available | not available | available | not available |
| I²C interface | available | available | not available | not available |

## 13. REGISTER MAP

| ADDR | BIT 7 | BIT 6 | BIT 5 | BIT 4 | BIT 3 | BIT 2 | BIT 1 | BIT 0 | REGISTER |
|---|---|---|---|---|---|---|---|---|---|
| 00H | DP0.7 (x) R | DP0.6 (x) R | DP0.5 (x) R | DP0.4 (x) R | DP0.3 (x) R | DP0.2 (x) R | DP0.1 (x) R | DP0.0 (x) R | DP0R (pin) |
| 01H | DP1.7 (x) R | DP1.6 (x) R | DP1.5 (x) R | DP1.4[1] (x) R | DP1.3 (x) R | DP1.2 (x) R | DP1.1 (x) R | DP1.0 (x) R | DP1R (pin) |
| 02H | DP0.7 (1) R/W | DP0.6 (1) R/W | DP0.5 (1) R/W | DP0.4 (1) R/W | DP0.3 (1) R/W | DP0.2 (1) R/W | DP0.1 (1) R/W | DP0.0 (1) R/W | DP0R (latch) |
| 03H | DP1.7 (1) R/W | DP1.6 (0) R/W | DP1.5 (0) R/W | DP1.4[1] (1) R/W | DP1.3 (1) R/W | DP1.2 (1) R/W | DP1.1 (1) R/W | DP1.0 (1) R/W | DP1R (latch) |
| 10H | - | - | PWM15 (0) R/W | PWM14 (0) R/W | PWM13 (0) R/W | PWM12 (0) R/W | PWM11 (0) R/W | PWM10 (0) R/W | PWM1 |
| 11H | - | - | PWM22 (0) R/W | PWM24 (0) R/W | PWM23 (0) R/W | PWM22 (0) R/W | PWM21 (0) R/W | PWM20 (0) R/W | PWM2 |
| 12H | - | - | PWM35 (0) R/W | PWM34 (0) R/W | PWM33 (0) R/W | PWM32 (0) R/W | PWM31 (0) R/W | PWM30 (0) R/W | PWM3 |
| 13H | - | - | PWM45 (0) R/W | PWM44 (0) R/W | PWM43 (0) R/W | PWM42 (0) R/W | PWM41 (0) R/W | PWM40 (0) R/W | PWM4 |
| 14H | - | - | PWM55 (0) R/W | PWM54 (0) R/W | PWM53 (0) R/W | PWM52 (0) R/W | PWM51 (0) R/W | PWM50 (0) R/W | PWM5 |
| 15H | - | VST06 (0) R/W | VST05 (0) R/W | VST04 (0) R/W | VST03 (0) R/W | VST02 (00 R/W | VST01 (0) R/W | VST00 (0) R/W | VSTL |
| 16H | - | VST13 (0) R/W | VST12 (0) R/W | VST11 (0) R/W | VST10 (0) R/W | VST09 (0) R/W | VST08 (0) R/W | VST07 (0) R/W | VSTH |
| 17H | - | - | - | - | - | AFC2 (0) R/W | AFC1 (0) R/W | AFC0 (0) R/W | AFCO |
| 18H | - | - | - | - | - | - | - | AFCC (x) R | AFCC |
| 19H | SCLE (0) R/W | SDAE (0) R/W | PWM5E (0) R/W | PWM4E (0) R/W | PWM3E (0) R/W | PWM2E (0) R/W | PWM1E (0) R/W | TDACE (0) R/W | DP0E/PWME |
| 1AH | - | - | - | AFCE (0) R/W | P14LVL (0) R/W | P6LVL (0) R/W | VOW2E (0) R/W | VOW1E (0) R/W | DP1E/ PWMLVL |
| 20H-3FH | CC1 (x) W | CC0 (x) W | MD5 (x) W | MD4 (x) W | MD3 (x) W | MD2 (x) W | MD1 (x) W | MD0 (x) W | DATA DISPLAY MEMORY |

| ADDR | BIT 7 | BIT 6 | BIT 5 | BIT 4 | BIT 3 | BIT 2 | BIT 1 | BIT 0 | REGISTER |
|---|---|---|---|---|---|---|---|---|---|
| 40H | CC34 (0) R/W | CC24 (0) R/W | CC14 (0) R/W | RBLK (0) R/W | ROUND (0) R/W | STBY (1) R/W | VLVL (0) R/W | HLVL 0 R/W | OSDCA |
| 41H | SZ01 (0) R/W | SZ00 (0) R/W | VP05 (0) R/W | VP04 (0) R/W | VP03 (0) R/W | VP02 (0) R/W | VP01 (0) R/W | VP00 (0) R/W | LINE 0A |
| 42H | BLK0 (0) R/W | VB0 (0) R/W | HP05 (0) R/W | HP04 (0) R/W | HP03 (0) R/W | HP02 (1) R/W | HP01 (0) R/W | HP00 (0) R/W | LINE 0B |
| 43H | CDTW (0) R/W | CDTH (0) R/W | CC33 (0) R/W | CC23 (0) R/W | CC32 (0) R/W | CC12 (1) R/W | CC21 (0) R/W | CCV11 (0) R/W | OSDCB |
| 44H | SZ11 (0) R/W | SZ10 (0) R/W | VP15 (0) R/W | VP14 (0) R/W | VP13 (0) R/W | VP12 (1) R/W | VP11 (0) R/W | VP10 (0) R/W | LINE 1A |
| 45H | BKL1 (0) R/W | VB1 (0) R/W | HP15 (0) R/W | HP14 (0) R/W | HP13 (0) R/W | HP12 (1) R/W | HP11 (0) R/W | HP10 (0) R/W | LINE 1B |

**Notes**

1. These bits are not available in the 84C441, 84C444, 84C641, 84C644, 84C841 and the 84C844.
2. The number within parentheses denotes the initial state; 'x' denotes don't care.

## 14. LIMITING VALUES

In accordance with the Absolute Maximum System (IEC 134)

| SYMBOL | PARAMETER | MIN. | MAX. | UNIT |
|---|---|---|---|---|
| $V_{DD}$ | supply voltage range | –0.3 | +7.0 | V |
| $V_I$ | all input voltages | –0.3 | $V_{DD}$ + 0.3 | V |
| $-I_{OH}$ | maximum source current - all port lines | - | 10 | mA |
| $-I_{OL}$ | maximum sink current - all port lines | - | 30 | mA |
| $P_{tot}$ | total power dissipation | - | 900 | mW |
| $T_{stg}$ | storage temperature range | –55 | + 125 | °C |
| $T_{amb}$ | operating ambient temperature range - all devices | –20 | + 70 | °C |

## 15. DC CHARACTERISTICS

$V_{DD}$ = 5 V ±10%; $V_{SS}$ = 0 V; $T_{amb}$ = –20 to + 70 °C; all voltages with respect to $V_{SS}$ unless otherwise specified

| SYMBOL | PARAMETER | CONDITIONS | MIN. | TYP. | MAX. | UNIT |
|---|---|---|---|---|---|---|
| **Supply** | | | | | | |
| $V_{DD}$ | operating voltage | | 4.5 | 5.0 | 5.5 | V |
| $I_{DD}$ | operating current | $f_{OSDCRC} = f_{OSDCLC} = f_{XTAL}$; see note 1 | | | | |
| | | $V_{DD}$ = 5 V; $f_{XTAL}$ = 10 MHz | - | 5 | 10 | mA |
| | | $V_{DD}$ = 5 V; $f_{XTAL}$ = 6 MHz | - | 3.5 | 8 | mA |
| $I_{DD}$ | operating current | $f_{OSDCRC} = f_{OSDCLC}$ = STOP; see note 1 | | | | |
| | | $V_{DD}$ = 5 V; $f_{XTAL}$ = 10 MHz | - | 3 | 7 | mA |
| | | $V_{DD}$ = 5 V; $f_{XTAL}$ = 6 MHz | - | 1.5 | 3.5 | mA |
| $I_{DD}$ | Idle mode current | $V_{DD}$ = 5 V; $f_{XTAL}$ = 10 MHz | - | 1.3 | 3 | mA |
| | | $V_{DD}$ = 5 V; $f_{XTAL}$ = 6 MHz; see note 1 | - | 0.8 | 1.5 | mA |
| $I_{DD}$ | Stop mode current | $V_{DD}$ = 5.5 V; see notes 1 and 2 | - | 5 | 10 | µA |
| **Inputs** | | | | | | |
| $V_{IL}$ | input LOW voltage; Ports P0; P1; DP0; DP1; HSYNCN; VSYNCN | | 0 | - | $0.3V_{DD}$ | V |
| $V_{IH}$ | input HIGH voltage; Ports P0; P1; DP0; DP1; HSYNCN; VSYNCN | | $0.7V_{DD}$ | - | $V_{DD}$ | V |
| $I_{IH}$ | input current RESET | $V_{in}$ = 0.5 V | 20 | - | - | µA |
| $V_{AI}$ | input voltage; DP1.7; AFC1 | | $V_{SS}$ | - | $V_{DD}$ | V |
| $V_{AE}$ | conversion error range (AFC) | | - | - | ± 1/2 | LSB |
| $\pm I_{LI}$ | input leakage current; Ports P0; P1; DP0; DP1 | $V_{SS} < V_I < V_{DD}$ | - | - | 10 | µA |
| $\pm I_{LI}$ | input leakage current INTN/T0; T1 | $V_{SS} < V_I < V_{DD}$ | 0.01 | 0.20 | 10 | µA |

| SYMBOL | PARAMETER | CONDITIONS | MIN. | TYP. | MAX. | UNIT |
|---|---|---|---|---|---|---|
| **Outputs** | | | | | | |
| $I_{OL}$ | output LOW sink current; Port P0 | $V_{DD}$ = 5 V ±10%; $V_O$ = 1.2 V | 10 | - | - | mA |
| $I_{OL}$ | output LOW sink current; Ports P1; DP0; DP1; VOB; VOW3 | $V_{DD}$ = 5 V ±10%; $V_O$ = 0.4 V | 1.2 | 3 | - | mA |
| $I_{OH}$ | pull-up output HIGH source current; Ports P0; P1; DP0; DP1 | $V_{DD}$ = 5 V ± 10%; $V_O$ = 0.7$V_{DD}$ | 40 | - | - | µA |
| $I_{OH}$ | pull-up output HIGH source current; Ports P0; P1; DP0; DP1 | $V_{DD}$ = 5 V ± 10%; $V_O$ = $V_{SS}$ | - | - | 400 | µA |
| $I_{OH}$ | push-pull output HIGH source current; Ports P0; P1; DP0; DP1; VOB; VOW3 | $V_{DD}$ = 5 V ± 10%; $V_O$ = $V_{DD}$–0.4V | 1.2 | 3 | - | mA |

**Notes**

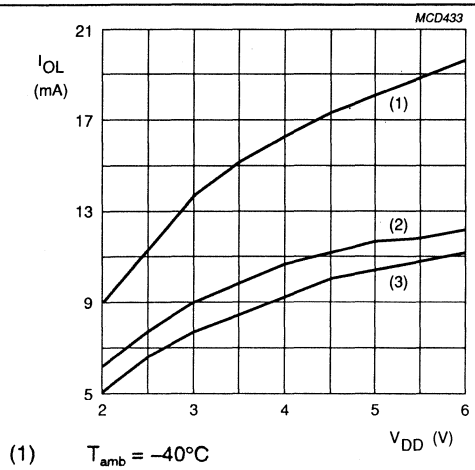
1. $V_{IL}$ = $V_{SS}$; $V_{IH}$ = $V_{DD}$; all outputs and sense input lines unloaded. All open drain ports connected to $V_{SS}$.
2. Crystal is connected between XTAL1 and XTAL2; T1 = $V_{SS}$; INTN/T0 = $V_{DD}$.

## 16. AC CHARACTERISTICS

### 16.1 Oscillator requirements

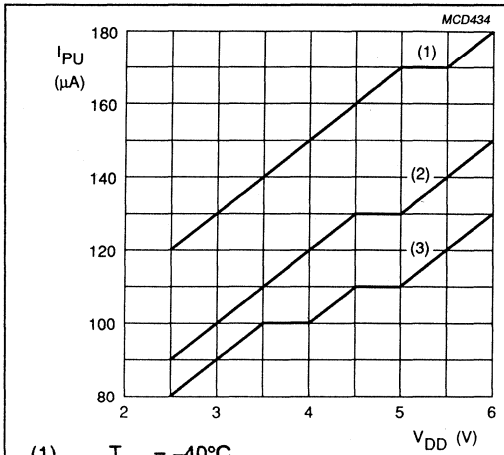
| SYMBOL | PARAMETER | CONDITIONS | MIN. | MAX. | UNIT |
|---|---|---|---|---|---|
| $f_{XTAL}$ | crystal frequency | $V_{DD}$ = 5 V ±10% | 1 | 10 | MHz |
| $f_{DOSC}$ | DOS oscillator frequency | $V_{DD}$ = 5 V ±10% | - | 10 | MHz |

### 16.2 Characteristic curves



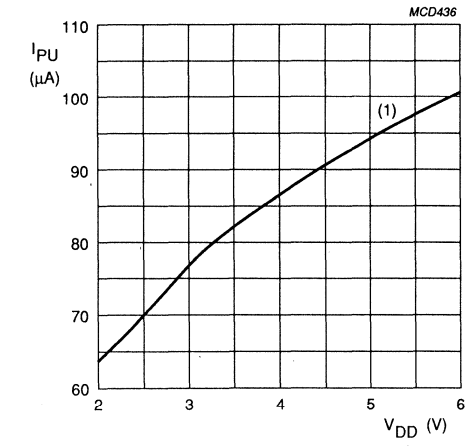

(1) $T_{amb}$ = –40°C
(2) $T_{amb}$ = 25°C
(3) $T_{amb}$ = 85°C

Fig.37 Typical output sink current as a function of supply voltage; $V_O$ = 0.4 V.



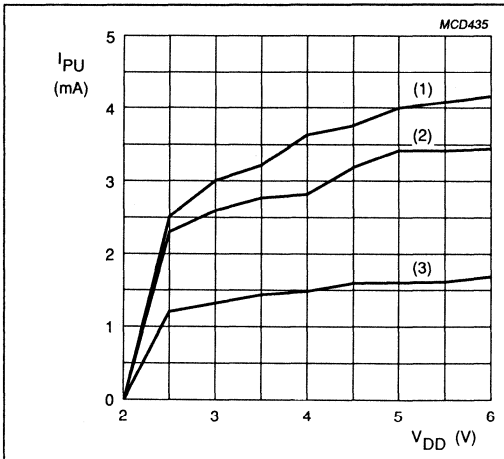

(1) $T_{amb}$ = –40°C
(2) $T_{amb}$ = 25°C
(3) $T_{amb}$ = 85°C

Fig.38 Typical pull-up current as a function of supply voltage; $V_O$ = 0 V.



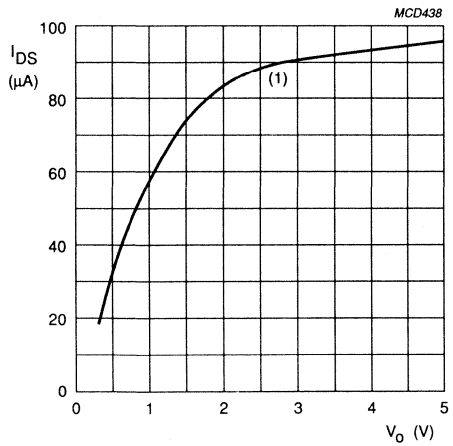

(1) $T_{amb}$ = 25°C

Fig.39 Typical pull-up current as a function of supply voltage; $V_O = 0.7V_{DD}$.



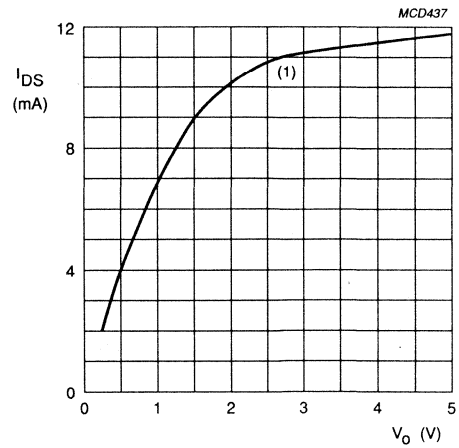

(1) $T_{amb}$ = –40°C
(2) $T_{amb}$ = 25°C
(3) $T_{amb}$ = 85°C

Fig.40 Typical booster current as a function of supply voltage; $V_O = V_{DD} - 0.4$ V.




(1) $T_{amb}$ = 25°C

Fig.41 Typical open drain sink capability as a function of $V_O$; Port 0




(1) $T_{amb}$ = 25°C

Fig.42 Typical open drain sink capability as a function of $V_O$; DP0.0 to DP0.7, DP1.5 to DP1.7 and P1.0 to P1.4

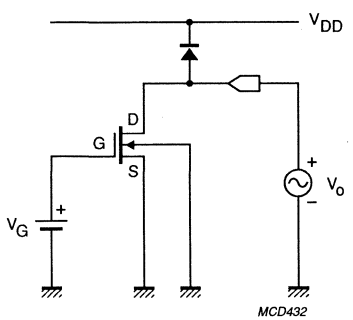
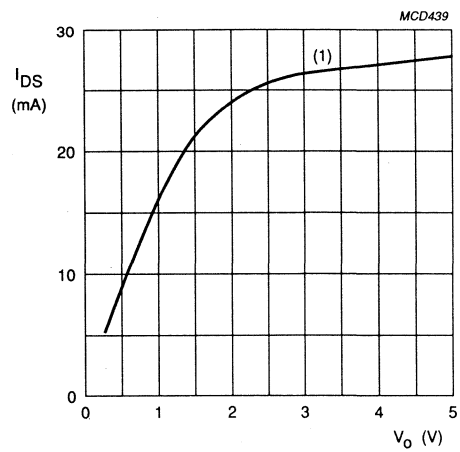



Fig.43 Open drain test circuit; DP0.0 to DP0.7, DP1.5 to DP1.7 and P1.0 to P1.4




(1) $T_{amb}$ = 25°C

Fig.44 Typical open drain sink capability as a function of $V_O$; VOW3 and VOB0

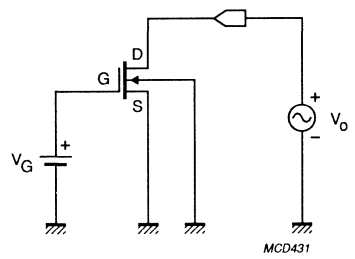



Fig.45 Open drain test circuit; VOW3 and VOB

**PURCHASE OF PHILIPS I²C COMPONENTS**

Purchase of Philips I²C components conveys a license under the Philips' I²C patent to use the components in the I²C system provided the system conforms to the I²C specification defined by Philips. This specification can be ordered using the code 9398 358 10011.

# SINGLE-CHIP 8-BIT MICROCONTROLLER WITH LCD DRIVERS, DERIVATIVE PORT, TIMER/CAPTURE AND TIMER/COUNTER

## DESCRIPTION

The PCF84C633A is a microcontroller with 20 on-chip liquid crystal display (LCD) outputs. These can be configured for one to four backplanes and 19 to 16 segment lines, yielding a maximum of 64 display elements. In addition to the shared features of the PCF84CXX family of microcontrollers, the PCF84C633A includes a 16-bit timer with capture and compare registers, a 16-bit up/down counter/timer and two filtered control inputs. Together with additional derivative port lines, these powerful extensions make the device attractive for demanding real-time applications.

## IMPORTANT

This data sheet details the specific properties of the PCF84C633A. The shared characteristics of the 84CXXX family of microcontrollers are described in 84CXXX family specification, which should be read in conjunction with this publication.

## FEATURES

- 8-bit CPU, ROM, RAM, I/O in a single 56-lead package
- 6 K bytes ROM
- 256 bytes RAM
- Over 80 instructions (based on MAB 8048) all of 1 or 2 cycles
- 28 quasi-bidirectional I/O port lines
- 8-bit programmable timer/event counter
- 16-bit derivative timer with capture and compare registers (T2)
- 16-bit up/down counter/timer (T3)
- 2 filtered input lines coupled to T2 and T3
- 3 single-level vectored interrupts; external, 8-bit programmable timer/event counter, derivative (triggered by 4 events in T2 and T3)
- 2 test inputs of which one also serves as the external interrupt input
- 20 LCD output configurable for one to four backplanes and 19 to 16 segment lines
- Drive for up to 64 display elements
- Display memory bank switching in static and duplex drive modes
- 19 of the LCD outputs may serve as additional low-drive logic outputs with optional level-shift
- Stop and idle modes
- Logic supply $V_{DD}$; 2.5 V to 5.5 V
- Independent LCD supply $V_{DDLC}$; 2.5 V to 5.5 V
- Clock frequency: 1 MHz to 16 MHz
- Operating temperature range: −40 °C to 85 °C
- Manufactured in silicon gate CMOS process

## PACKAGE OUTLINE

PCF84C633AT: 56-lead mini-pack; plastic (VSO56; SOT190)

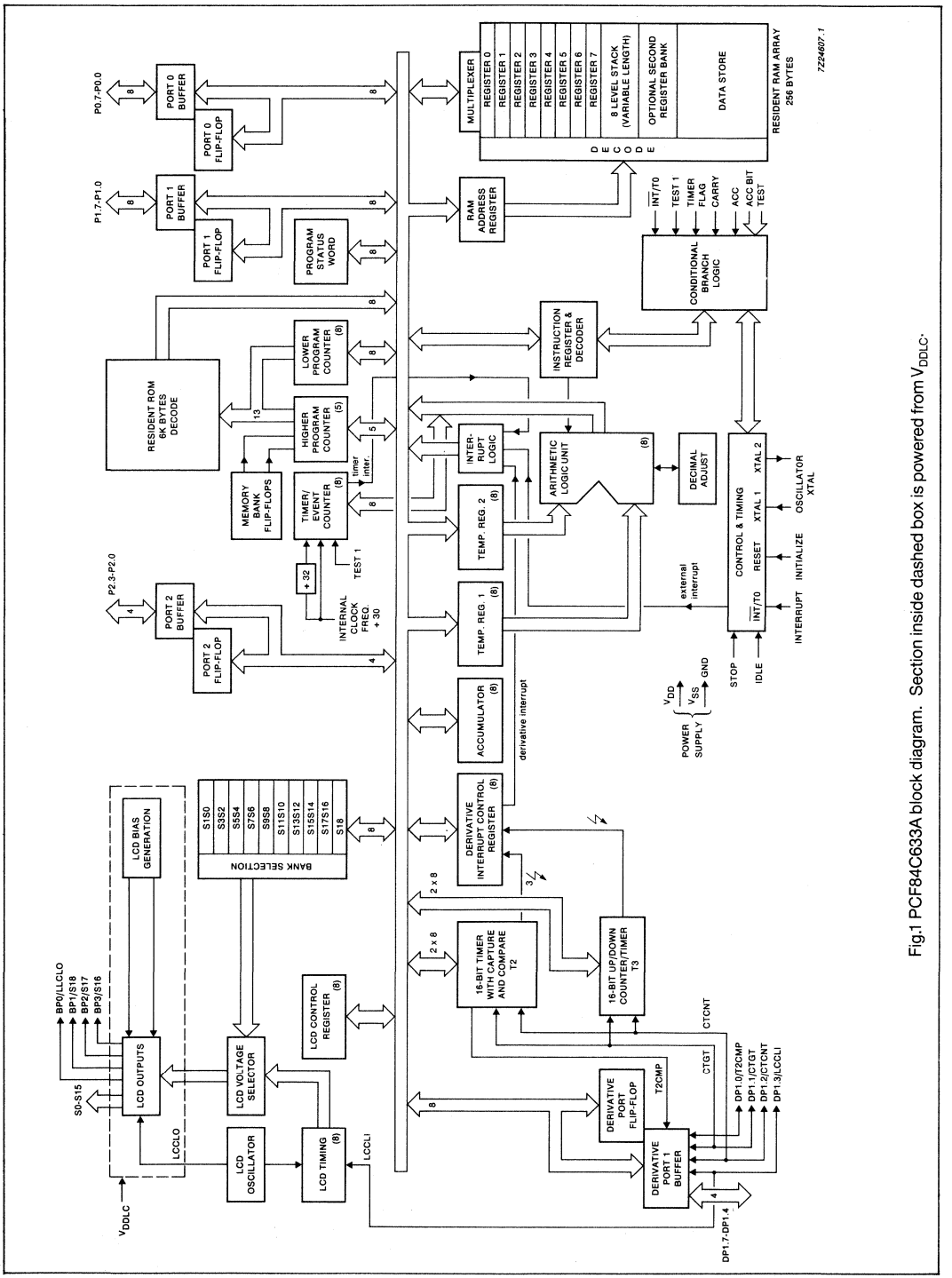



Fig.1 PCF84C633A block diagram. Section inside dashed box is powered from $V_{DDLC}$.

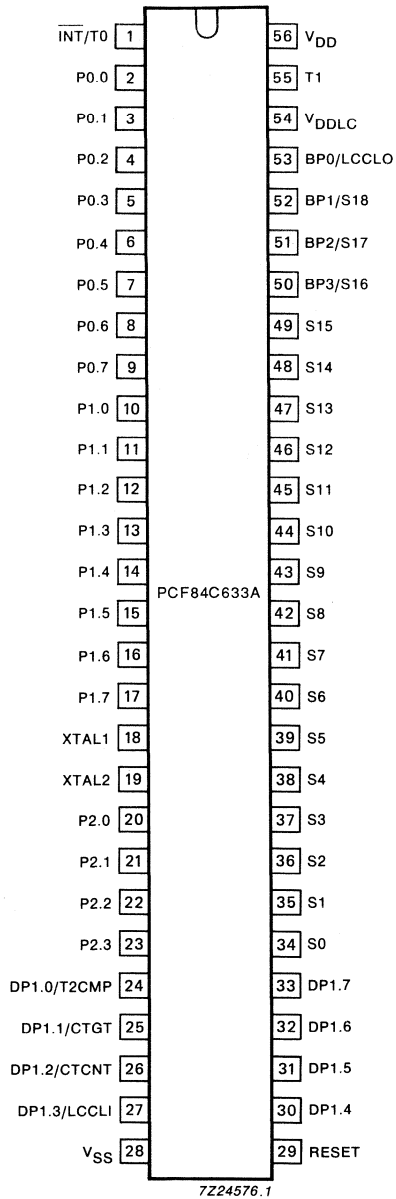



Fig.2 Pinning diagram.

**Pinning**

| PIN | SYMBOL | TYPE | FUNCTION |
|---|---|---|---|
| 1 | INT/T0 | I | Interrupt/Test 0 |
| 2-9 | P0.0-P0.7 | I/O | Port 0; 8-bit quasi-bidirectional I/O port |
| 10-17 | P1.0-P1.7 | I/O | Port 1; 8-bit quasi-bidirectional I/O port |
| 18 | XTAL1 | I | Crystal oscillator/external clock input |
| 19 | XTAL2 | O | Crystal oscillator output |
| 20-23 | P2.0-P2.3 | I/O | Port 2; 4-bit quasi-bidirectional I/O port |
| 24 | DP1.0/T2CMP | I/O | Derivative Port 1; quasi-bidirectional I/O line/compare output of T2 |
| 25 | CP1.1/CTGT | I/O | Derivative Port 1; quasi-bidirectional I/O line/capture input (T2) and/or gate input (T3) |
| 26 | DP1.2/CTCNT | I/O | Derivative Port 1; quasi-bidirectional I/O line/capture input (T2) and/or count input (T3) |
| 27 | DP1.3/LCCLI | I/O | Derivative Port 1; quasi-bidirectional I/O line/LCD clock input |
| 28 | $V_{SS}$ | P | Ground |
| 29 | RESET | I | Reset input |
| 30-33 | DP1.4-DP1.7 | I/O | Derivative Port 1; quasi-bidirectional I/O port |
| 34-49 | S0-S15 | O | LCD segment outputs |
| 50-52 | BP3/S16-BP1/S18 | O | LCD backplane/segment outputs |
| 53 | BP0/LCCLO | O | LCD backplane/LCD oscillator output |
| 54 | $V_{DDLC}$ | P | LCD supply voltage |
| 55 | T1 | I | Test 1/count input of 8-bit timer/event counter |
| 56 | $V_{DD}$ | P | Logic supply voltage |

## PARALLEL PORTS

All standard quasi-bidirectional I/O ports are available:

- Port 0 parallel port of 8 lines (P0.0 to P0.7)
- Port 1 parallel port of 8 lines (P1.0 to P1.7)
- Port 2 parallel port of 4 lines (P2.0 to P2.3)

Since no serial I/O interface is provided, P2.3 is a general purpose line without restrictions.
In addition to the standard ports, a derivative I/O port is available:

- DP1 derivative port of 8 lines (DP1.0/T2CMP, DP1.1/CTGT, DP1.2/CTCNT, DP1.3/LCCLI, DP1.4 to DP1.7)

Table 1 summarizes the derivative addresses of DP1.

**Table 1** Derivative port address pair.

| Dx ADDRESS (HEX) | TYPE | DESCRIPTION |
|---|---|---|
| 02 | R | DP1 derivative port lines |
| 03 | R/W | DP1 derivative port flip-flop |

Four lines of derivative Port 1 are shared with signals used for timer T2 , up/down counter/timer T3 and the LCD driver section; DP1.0/T2CMP, DP1.1/CTGT, DP1.2/CTCNT and DP1.3/LCCLI. Before an alternative function can be used, the output driver FET of the corresponding port-line must be turned off by writing a '1' to the port latch. The port pin is then pulled high by the internal pull-up (standard I/O and push-pull I/O) or is floating (open drain I/O), but may be driven by an external signal. In their non-port functions, DP1.1/CTGT, DP1.2/CTCNT and DP1.3/LCCLI serve as inputs. Therefore, they may only be configured as standard outputs (option 1) or open drain outputs (option 2). All other standard or derivative port lines are eligible for all three mask options.

The alternative functions are controlled by the corresponding control bits. When an alternative function is used the corresponding I/O port is not disabled, so the state of the pin and latch may still be read using the MOV A,DX instruction. Note, if a MOV DX,A instruction sets the port latch to 'O', it can no longer be driven by an external signal anymore.

## DP1.1/CTGT AND DP1.2/CTCNT INPUT FILTERS

The T2 capture transitions and the T3 gating and count signals are digitally filtered, to remove spikes shorter than one machine cycle. The signals which are presented on DP1.1/CTGT and DP1.2/CTCNT are sampled on the positive edge of the internal machine cycle signal ($f_{XTAL}/30$). A change on the input lines must be stable for two consecutive samples to be accepted. Therefore the filter delay ranges from one to two machine cycles (see Fig.3). The T2 capture and a T3 counting pulses are delayed by two to three machine cycles with respect to the corresponding input signal transition (see Fig.3). The internal signals GT, CT and CNT occur only if the corresponding derivative function is enabled.
The filter delays shown in Fig.3 are only valid if the corresponding function is enabled.

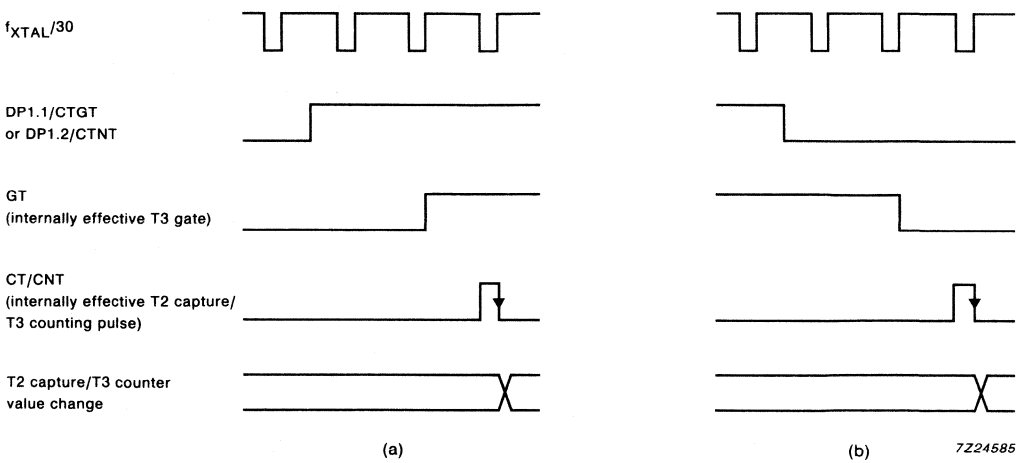



Fig.3 Filter delays at positive edge (a) or negative edge (b) on DP1.1/CTGT and DP1.2/CTCNT.

### 16-BIT TIMER T2

A derivative 16-bit timer (T2) with capture and compare registers is provided. Communication between the CPU and T2 is handled through derivative registers, making use of the derivative input/output instructions. 16-bit values are accessed by two consecutive byte accesses. Fig.4 gives the block diagram of the T2 section.
Typical applications of T2 are for measuring signal deviations (in conjunction with the capture register) or for generating pulse deviations (in conjunction with the compare register).

### T2 Derivative Registers

Table 2 summarizes the derivative addresses, the register mnemonics and the access types for the T2 section.

**Table 2** Derivative addresses for the T2 section.

| Dx ADDRESS (HEX) | TYPE | MNEMONIC | DESCRIPTION |
|---|---|---|---|
| 04 | R | T2H | T2 timer register high byte |
| 05 | R | T2L | T2 timer register low byte |
| 06 | R | T2LB | T2 low byte register |
| 07 | R/W | T2CMH | T2 compare register high byte |
| 08 | R/W | T2CML | T2 compare register low byte |
| 09 | R | T2CTH | T2 capture register high byte |
| 0A | R | T2CTL | T2 capture register low byte |
| 0B | R/W | T2CON | T2 control register |

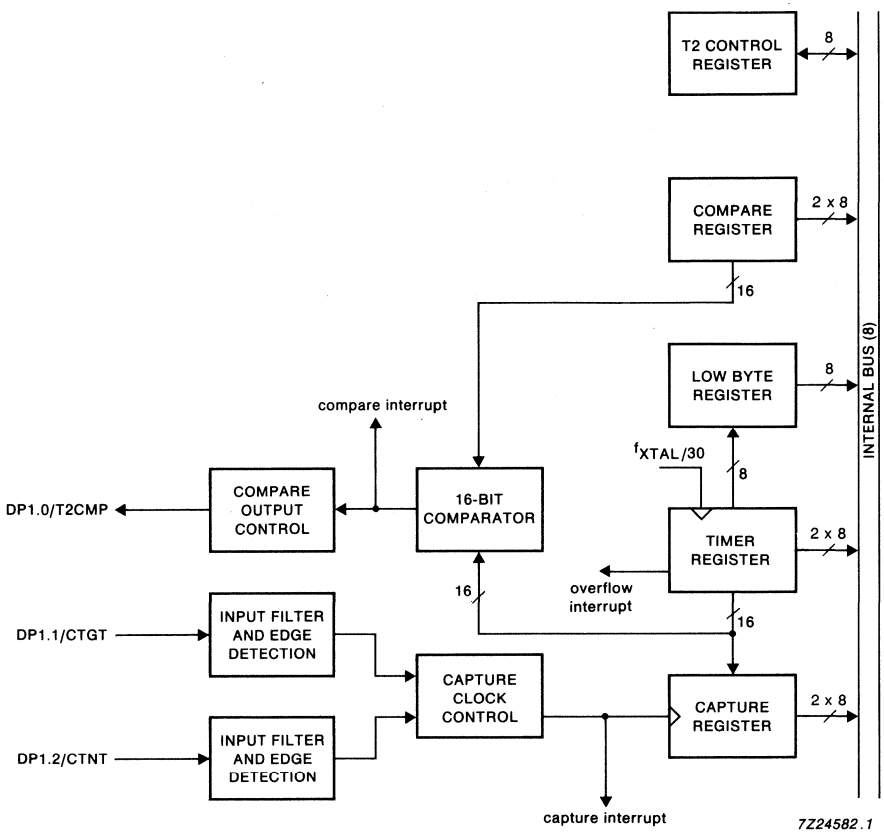



Fig.4 Block diagram of the derivative 16-bit timer T2.

### Timer Register and Low Byte Register

The central unit of T2 is the 16-bit timer register. Its behaviour depends on the state of bits ET2 and RUN in the T2 control register (see Table 3). If ET2 = 0, the timer register is disabled and cleared. Otherwise the timer register is enabled and keeps its value (RUN = 0) or increments (RUN = 1) every machine cycle ($f_{XTAL}/30$).

When the timer register overflows from FFFFH to 0000H, the T2 overflow flag T20V is set in the derivative interrupt control register (see section on derivative interrupts). This event can be used to generate one of four different derivative interrupt request. The source of the interrupt is found by reading DIRCON (see section Derivative Interrupts).

The 16-bit timer register is subdivided into two 8-bit registers. T2H contains the high byte of the timer value, T2L contains the low byte. Both are read-only registers and can be read anytime without disturbing the timer function. Whenever T2H is read, T2L is simultaneously copied into the low byte register T2LB. This allows the full 16-bit timer value to be read consistently. The value in T2LB remains fixed until T2H is read again.

### Capture Function

The 16-bit capture register latches the value of the 16-bit timer register when a predefined transition on DP1.1/CTGT or DP1.2/CTCNT occurs. Bits CT0 and CT1 in the T2 control register (see Table 3) are used to either disable the capture function or to select the positive, the negative or both edges of the input signal to latch the timer value into the capture register.

Bit CTS in the T2 control register selects DP1.1/CTGT (CTS = 1) or DP1.2/CTCNT (CTS = O) as the source for the capture clock. Signals on DP1.1/CTGT and DP1.2/CTCNT are filtered (see Fig.4). For a transition to be detected, the signal on DP1.1/CTGT and DP1.2/CTCNT must be stable for at least two machine cycles ($60/f_{XTAL}$) before and after the selected edge. If several consecutive capture conditions occur, the capture register always contains the most recent value.

When the capture trigger occurs, the capture interrupt flag CTI is set in the derivative interrupt control register (see section on derivative interrupts). This event can be used to generate a derivative interrupt request. The source of the interrupt is found by reading DIRCON (see section Derivative Interrupts).

The 16-bit register is subdivided into two 8-bit registers. T2CTH contains the high byte of the capture value, T2CTL contains the low byte. Both are read-only. When T2CTH is read, the capture function is inhibited until T2CTL is also read. This allows the full 16-bit capture value to be consistently read.

### Compare Function

If (ET2 = 1).(RUN = 1) in the T2 control register, a match between the timer register and the 16-bit compare register will set the compare interrupt flag CMI in the derivative interrupt control register (see section on Derivative Interrupts). This event can be used to generate a derivative interrupt request. The source of the interrupt is found by reading DIRCON (see section Derivative Interrupts).

The 16-bit compare register is subdivided into two 8-bit registers. T2CMH contains the high byte of the compared value, T2CML contains the low byte. Both can be read or written. If the compare function is not used, T2CMH and T2CML may be used as storage locations. When T2CMH is written, the compare function is inhibited until T2CML is also written. This allows the full 16-bit compare value to be consistently defined.

If bit ECO = 1 in the T2 control register, a compare event can generate an output on DP1.0/T2CMP. Bit CM0 and CM1 in the T2 control register select between no change, output low, output high or output toggle. This may be useful to generate a pulse width or to signal a lapse of time.

### T2 Control Register (T2CON)

The 8-bit T2 control register defines the behaviour of the T2 section. It can be read or written. Fig.5 gives the structure of the T2 control register, and Table 3 summarizes the significance of the individual control bits.

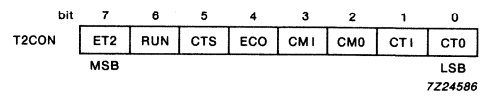



Fig.5 T2 control register (T2CON).

**Table 3** Overview of T2 control register bits.

| BIT | NAME | DESCRIPTION | | |
|---|---|---|---|---|
| ET2 | Enable T2 | ET2 = 0: timer register disabled and cleared<br>ET2 = 1: timer register enabled (see RUN) | | |
| RUN | RUN | RUN = 0: timer register value held<br>RUN = 1: timer register increments if ET2 = 1 | | |
| CTS | Capture Source | CTS = 0: DP1.2/CTCNT capture signal source<br>CTS = 1: DP1.1/CTGT capture signal source | | |
| ECO | Enable Compare Out | ECO = 0: DP1.0/T2CMP derivative port line<br>ECO = 1: DP1.0/T2CMP compare output | | |
| CM1<br>CM0 | Compare 1<br>Compare 0 | **CM1** | **CM0** | |
| | | 0 | 0 | no change on DP1.0/T2CMP after compare match |
| | | 0 | 1 | low level on DP1.0/T2CMP after compare match if ECO = 1 |
| | | 1 | 0 | high level on DP1.0/T2CMP after compare match if ECO = 1 |
| | | 1 | 1 | toggles DP1.0/T2CMP after compare match if ECO = 1 |
| CT1<br>CT0 | Capture 1<br>Capture 0 | **CT1** | **CT0** | |
| | | 0 | 0 | capture function disabled |
| | | 0 | 1 | capture on positive edge of capture signal source |
| | | 1 | 0 | capture on negative edge of capture signal source |
| | | 1 | 1 | capture on any edge of capture signal source |

### 16-BIT UP/DOWN COUNTER/TIMER T3

A derivative 16-bit up/down counter/timer (T3) is provided. Communication between the CPU and T3 is handled through derivative registers, making use of the derivative input/output instructions. The 16-bit counter/timer value is accessed by two consecutive byte accesses. Fig.6 gives the block diagram of the T3 section.

Typical applications of T3 are for measuring signal durations (in timer mode in conjunction with the gating signal on DP1.1/CTGT), for interval generation (in timer mode) and for counting signal edges (in counter mode).

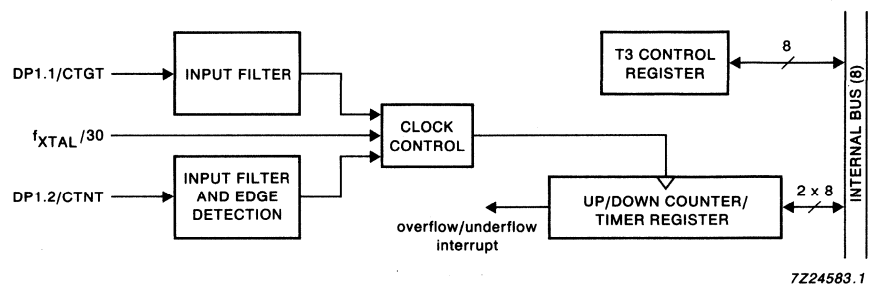



Fig.6 Block diagram of the derivative 16-bit up/down counter/timer T3.

### T3 Derivative Registers

Table 4 summarizes the derivative addresses, the register mnemonics and the access types for the T3 section.

**Table 4** Derivative addresses for the T3 section.

| Dx ADDRESS (HEX) | TYPE | MNEMONIC | DESCRIPTION |
|---|---|---|---|
| 0D | R/W | T3H | T3 counter/timer register high byte |
| 0E | R/W | T3L | T3 counter/timer register low byte |
| 0F | R/W | T3CON | T3 control register |

### Up/Down Counter/Timer Register

The central unit of T3 is the 16-bit up/down counter/timer register. Its behaviour is defined by the state of mode bits MD0, MD1 and MD2 in the T3 control register (see Tables 5 and 6 and also Fig.7).

In timer mode (MD2 = 0), the register increments (MD1 = 1) or decrements (MD0 = 1) every machine cycle ($f_{XTAL}/30$).

In counter mode(MD2 = 1), the register increments (MD1 = 1) with the positive edge on DP1.2/CTCNT. The register decrements (MD = 1) with the negative edge on DP1.2/CTCNT. For a transition to be detected, the clock on DP1.2/CTCNT must be stable for at least two machine cycles ($60/f_{XTAL}$) before and after the signal change.

In both timer and counter modes, counting can be inhibited if the gating signal on DP1.1/CTGT equals zero. This feature is turned on by bit ET3GT (enable T3 gate) in the T3 control register. Being digitally filtered, the state of DP1.1/CTGT must be stable for at least two machine cycles ($60/f_{XTAL}$) before it is recognized as a valid gating signal. It is possible to simultaneously enable counting up and counting down in count mode (MD1 = MD0 = 1). When this is done, the gating signal should be enabled by setting bit ET3GT (in DIRCON) to '1'. Then signal edges will only be counted while the gating signal (DP1.1/CTGT) is '1'.

In both timer and counter modes, an overflow (from FFFFH to 0000H) or an underflow (from 0000H to FFFFH) assert the overflow flag T30V in the derivative interrupt control register (see section on derivative interrupts). This event can be used to generate a derivative interrupt request. The source of the interrupt is found by reading DIRCON (see section Derivative Interrupts).

The 16-bit up/down counter/timer register is subdivided into two 8-bit registers. T3H contains the high byte of the counter/timer value, T3L contains the low byte. Both can be read or written. It is the responsibility of the user software to ensure that accesses to T3H and T3L do not violate data consistency. One way to guarantee this is by disabling the T3 section for the duration of the access. If the up/down counter/timer is not used, T3H and T3L may serve as storage locations.

**Table 5** Truth table T3 Up/Down Timer.

| T3 CLOCK INPUT | | T3 CONTROL REGISTER | | | | T3 GATE | TIMER MODE |
|---|---|---|---|---|---|---|---|
| $f_{XTAL}/30$ | DP1.2/ CTCNT | MD2 | MD1 | MD0 | ET3GT | DP1.1/ CTGT | |
| ⎍ | X | 0 | 1 | 0 | 0 | X | Timer Up |
| ⎍ | X | 0 | 1 | 0 | 1 | 1 | Timer Up |
| ⎍ | X | 0 | 1 | 0 | 1 | 0 | Inhibit Timer Up by DP1.1/CTGT |
| ⎍ | X | 0 | 0 | 1 | 0 | X | Timer Down |
| ⎍ | X | 0 | 0 | 1 | 1 | 1 | Timer Down |
| ⎍ | X | 0 | 0 | 1 | 1 | 0 | Inhibit Timer Down by DP1.1/CTGT |
| X | X | X | 0 | 0 | X | X | Inhibit Timer |

'X' denotes don't care states

**Table 6** Truth table Up/Down Counter.

| T3 CLOCK INPUT | | T3 CONTROL REGISTER | | | | T3 GATE | COUNTER MODE |
|---|---|---|---|---|---|---|---|
| $f_{XTAL}/30$ | DP1.2/ CTCNT | MD2 | MD1 | MD0 | ET3GT | DP1.1/ CTGT | |
| X | ↓ | 1 | 1 | 0 | 0 | X | Count Up |
| X | ↓ | 1 | 1 | 0 | 1 | 1 | Count Up |
| X | ↓ | 1 | 1 | 0 | 1 | 0 | Inhibit Count Up by DP1.1/CTGT |
| X | ↑ | 1 | 0 | 1 | 0 | X | Count Down |
| X | ↑ | 1 | 0 | 1 | 1 | 1 | Count Down |
| X | ↑ | 1 | 0 | 1 | 1 | 0 | Inhibit Count Down by DP1.1/CTGT |
| X | ↓ | 1 | 1 | 1 | 1 | 1 | Count Up |
| X | ↑ | 1 | 1 | 1 | 1 | 1 | Count Down |
| X | X | 1 | 1 | 1 | 1 | 0 | Inhibit Count Down by DP1.1/CTGT |
| X | X | X | 0 | 0 | X | X | Inhibit Counter |

'X' denotes don't care states

### T3 Control Register (T3CON)

The 8-bit T3 control register defines the behaviour of the T3 section. It can be read or written. Fig.7 gives the structure of T3CON, and Table 7 summarizes the significance of the individual control bits.

Bits 5 to 7 are fixed at zero.

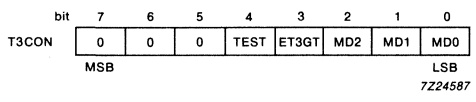



Fig.7 T3 control register (T3CON).

**Table 7** Overview of T3 control register bits.

| BIT | NAME | DESCRIPTION | | | |
|---|---|---|---|---|---|
| TEST | Test | TEST = 0: test mode disabled (state for user software)<br>TEST = 1: test mode enabled (not allowed in user software) | | | |
| ET3GT | Enable T3 Gate | ET3GT = 0: state of DP1.1/CTGT irrelevant to T3<br>ET3GT = 1: counter/timer T3 inhibited/enabled if DP1.1/CTGT = 0/1 | | | |
| MD2<br>MD1<br>MD0 | Mode 2<br>Mode 1<br>Mode 0 | **MD2** | **MD1** | **MD0** | |
| | | X | 0 | 0 | counter/timer T3 inhibited |
| | | 0 | 0 | 1 | decrementing timer mode |
| | | 0 | 1 | 0 | incrementing timer mode |
| | | 0 | 1 | 1 | not allowed |
| | | 1 | 0 | 1 | decrementing counter mode |
| | | 1 | 1 | 0 | incrementing counter mode |
| | | 1 | 1 | 1 | decrementing and incrementing counter mode (use with ET3GT = 1) |

## DERIVATIVE INTERRUPTS

Since the PCF84C633A includes no serial I/O interface, the SIO/derivative interrupt (see 84CXX family data sheet) reduces to a derivative interrupt. Four derivative interrupt events are defined. These events are controlled by the derivative interrupt control register (DIRCON). Table 8 summarizes the significance of the individual control bits.

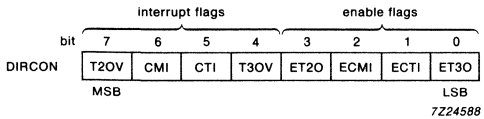



Fig.8 Derivative interrupt control register (DIRCON).

A derivative interrupt request is honoured if:-

- no interrupt routine proceeds
- no external interrupt request is pending
- the SIO/derivative interrupt is enabled
- the corresponding enable bit in the derivative interrupt control register is set

The derivative interrupt routing must include instructions that will remove the cause of the derivative interrupt by implicitly clearing the corresponding interrupt flag. Table 9 gives derivative address, the register mnemonics and the access type for the derivative interrupt control register.

DIRCON can be read or written. Read access is necessary to determine the cause of a derivative interrupt request. If the derivative interrupt is not used, the flags may be directly tested by the program. Write access is necessary to remove the cause of the derivative interrupt request. The flags may also be directly written to generate a software interrupt.

**Table 8** Overview of derivative interrupt control register bits.

| BIT | NAME | DESCRIPTION |
|---|---|---|
| T20V | T2 Overflow | Set: if T2 timer register overflows from FFFFH to 0000H (or by program)<br>Reset: by program and by RESET |
| CMI | Compare Interrupt | Set: if T2 timer register equals T2 compare register while (ET2 = 1) and (RUN = 1) in the T2 control register (or by program)<br>Reset: by program and by RESET |
| CTI | Capture Interrupt | Set: if capture trigger occurs (or by program)<br>Reset: by program and by RESET |
| T30V | T3 Overflow | Set: if T3 up/down counter/timer register over flows from FFFFH to 0000H or underflows from 0000H to FFFFH (or by program)<br>Reset: by program and by RESET |
| ET20 | Enable T2 Overflow | ET20 = 0: T20V event cannot request interrupt<br>ET20 = 1: T20V event requests interrupt |
| ECMI | Enable Compare Interrupt | ECMI = 0: CMI event cannot request interrupt<br>ECMI = 1: CMI event requests interrupt |
| ECTI | Enable Capture Interrupt | ECTI = 0: CTI event cannot request interrupt<br>ECTI = 1: CTI event requests interrupt |
| ET30 | Enable T3 Overflow | ET30 = 0: T30V event cannot request interrupt<br>ET30 = 1: T30V event requests interrupt |

**Table 9** Derivative interrupt control register address.

| Dx ADDRESS (HEX) | TYPE | MNEMONIC | DESCRIPTION |
|---|---|---|---|
| 0C | R/W | DIRCON | Derivative interrupt control register |

### LCD DRIVER SECTION

A versatile derivative LCD driver section is provided which interfaces the PCF84C633A to a wide variety of LCDs. The 20 LCD outputs can be configured as one backplane/19 segments (static drive), as two backplanes/18 segments (1:2 multiplex), as three backplanes/17 segments (1:3 multiplex) and as four backplanes/16 segments (1:4 multiplex).

It is also possible to configure the LCD section as up to 19 low-drive logic outputs including a level-shift to $V_{DDLC}$. These may be useful as an extension of port output when a large number of signal lines must be controlled, possibly referenced to another supply.

Table 10 summarizes the most typical applications of the LCD driver section. As indicated in the block diagram (Fig.9), communication between the CPU and the LCD driver section is handled through derivative registers, making use of the derivative input/output instructions.

**Table 10** Typical applications of the LCD driver section.

| ACTIVE BACKPLANES | NO. OF SEGMENTS | 7-SEGMENT NUMERIC | 14-SEGMENT ALPHANUMERIC | DOT MATRIX |
|---|---|---|---|---|
| 4 | 16 | 8 digits + 8 indicator symbols | 4 characters + 8 indicator symbols | 64 dots (4 x 16) |
| 3 | 17 | 6 digits + 9 indicator symbols | 3 characters + 9 indicator symbols | 51 dots (3 x 17) |
| 2 | 18 | 4 digits + 8 indicator symbols | 2 characters + 8 indicator symbols | 36 dots (2 x 18) |
| 1 | 19 | 2 digits + 5 indicator symbols | 1 characters + 5 indicator symbols | 19 dots (1 x 19) |
| – | 19 low-drive logic outputs with level-shift (referenced to $V_{SS}$ and $V_{DDLC}$) | | | |

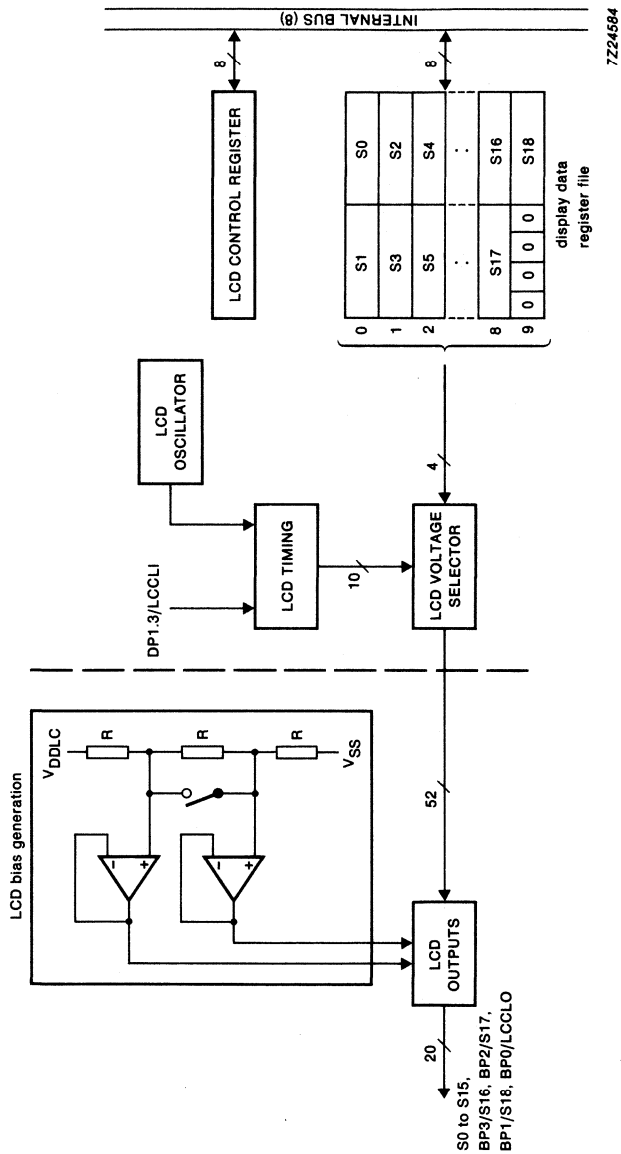



Fig.9 Block diagram of the LCD driver section.

### LCD Derivative Registers

Table 11 summarizes the derivative addresses, the register mnemonics and the access types for the LCD driver section.

**Table 11** Derivative addresses for the LCD driver section.

| Dx ADDRESS (HEX) | TYPE | MNEMONIC | DESCRIPTION |
|---|---|---|---|
| 20 | R/W | S1S0 | Display data nibbles for S0 and S1 |
| 21 | R/W | S3S2 | Display data nibbles for S2 and S3 |
| 22 | R/W | S5S4 | Display data nibbles for S4 and S5 |
| 23 | R/W | S7S6 | Display data nibbles for S6 and S7 |
| 24 | R/W | S9S8 | Display data nibbles for S8 and S9 |
| 25 | R/W | S11S10 | Display data nibbles for S10 and S11 |
| 26 | R/W | S13S12 | Display data nibbles for S12 and S13 |
| 27 | R/W | S15S14 | Display data nibbles for S14 and S15 |
| 28 | R/W | S17S16 | Display data nibbles for S16 and S17 |
| 29 | R/W | S18 | Display data nibble for S18 |
| 2A | R/W | LCDCON | LCD control register |

### Display Data Register File

The display data register file defines the LCD segment outputs. There is a one-to-one correspondence between segment outputs (S0 to BP1/S18) and nibbles in the display data register file (see Fig.10).

The four nibble bits A, B, C and D represent the four banks of data behind every LCD segment output. In 1:4 multiplex, banks A to D represent the states of the LCD elements multiplexed with backplanes BP0 to BP3/S16, respectively. Likewise, in 1:3 multiplex, banks A to C correspond to LCD elements multiplexed with BP0 to BP2/S17. In this case, bank D will not be used for LCD output.

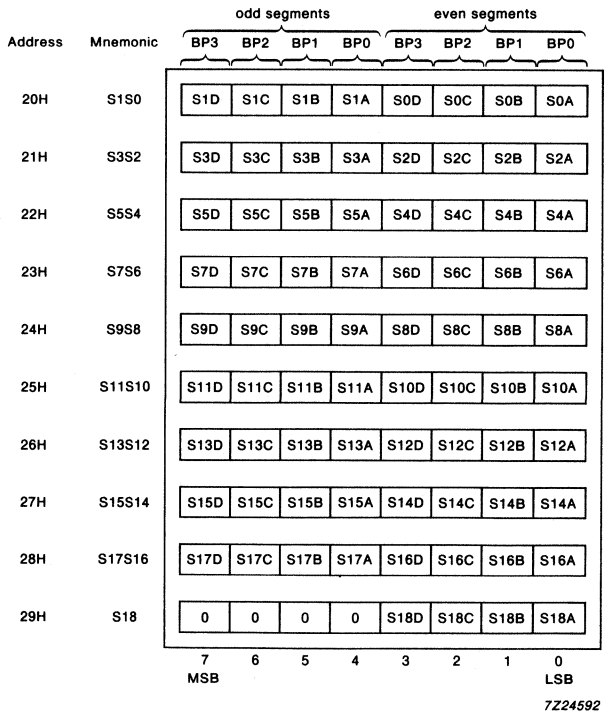



Fig.10 Display data register file.

In 1:2 multiplex, either bank pair A/B or bank pair C/D may be multiplexed with BP0 and BP1/S18. Likewise, in static drive and in low-drive logic mode, a choice of four banks exists. Bank selection allows preparation of additional information in alternative banks and to efficiently switch back and forth, making use of bits BK0 and BK1 in the LCD control register. This feature may be useful to efficiently blink groups of display elements.

All locations of the display data register file can be read or written. Note that bits 4 to 7 of the location at derivative address 29H are fixed at zero (see Fig.10). A logic one or zero in a bit of the display register file respectively translate into an on or off-waveform for the corresponding LCD segment output. In low-drive logic mode, a logic one or zero in a bit of the display data register file respectively translate into a HIGH or LOW level for the corresponding output.

Display data register file locations that are unused for output may serve as general purpose storage. In particular, this applies to nibbles corresponding to segment outputs shared with active backplane outputs, in bank D in 1:3 multiplex and to non-selected banks in static, duplex and low-drive logic modes.

### LCD Control Register (LCDCON)

The LCD control register completely defines the behaviour of the LCD driver section. It can be read or written. Fig.11 gives the structure of LCDCON, and Table 12 summarizes the significance of the individual control bits.

Bit 7 fixed at zero

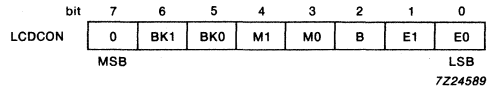



Fig.11 LCD control register (LCDCON).

**Table 12** Overview of LCD control register bits.

| BIT | NAME | DESCRIPTION | | | |
|---|---|---|---|---|---|
| BK1<br>BK0 | Bank 1<br>Bank 0 | **BK1** | **BK0** | **Bank** | |
| | | 0 | 0 | A | selected in static drive and in low-drive mode |
| | | 0 | 1 | B | selected in static drive and in low-drive mode |
| | | 1 | 0 | C | selected in static drive and in low-drive mode |
| | | 1 | 1 | D | selected in static drive and in low-drive mode |
| | | X | 0 | pair A/B | selected in 1:2 multiplex mode |
| | | X | 1 | pair C/D | selected in 1:2 multiplex mode |
| | | X | X | triplet A/B/C | selected in 1:3 mode |
| | | X | X | quartet A/B/C/D | selected in 1:4 mode |
| M1<br>M0 | Multiplex 1<br>Multiplex 0 | **M1** | **M0** | | |
| | | 0 | 0 | | 1:4 multiplex (4 BPs) |
| | | 0 | 1 | | static drive (1 BP) |
| | | 1 | 0 | | 1:2 multiplex (2 BPs) |
| | | 1 | 1 | | 1:3 multiplex (3 BPs) |
| | | X | X | | irrelevant in low-drive logic mode (E1 = 0, E0 = 1) |
| B | Bias | B = 0 | | 1/3 bias | in 1:2, 1:3 and 1:4 multiplex |
| | | B = 1 | | 1/2 bias | in 1:2, 1:3 and 1:4 multiple |
| | | B = X | | – | irrelevant in static drive and in low-drive logic mode (E1 = 0, E0 = 1) |

| E1 | Enable 1 | **E1** | **E0** | **Mode** | **S0 to BP1/S18** | **BP0/LCCLO** | **LCD oscillator** | **DP1.3/LCCLI** |
|---|---|---|---|---|---|---|---|---|
| E0 | Enable 0 | 0 | 0 | disable (LCD blank) | at $V_{SS}$ | at $V_{SS}$ | stopped | derivative line |
| | | 0 | 1 | low-drive logic | logic outputs | LCD clock output | running | derivative port line |
| | | 1 | 0 | LCD drive 1 | LCD outputs | BP0 output | stopped | LCD clock input |
| | | 1 | 1 | LCD drive 2 | LCD outputs | BP0 output | running | derivative |

'X' denotes don't care states.

### LCD Timing

The LCD timing organizes the internal data flow of the LCD driver section. This includes the transfer of data from the display data register file to the LCD outputs.
In LCD drive modes (see bits E0 and E1 in the LCD control register), the LCD frame corresponds to 24 periods of the LCD clock. At a nominal LCD frame frequency of 64 Hz, the LCD clock frequency must be about 1.5 kHz. The LCD clock must always be supplied otherwise, the LCD may be frozen in a DC state. As a clocking source, either an external clock on DP1.3/LCCLI (LCD drive mode 1, Table 12) or the internal LCD oscillator (LCD drive mode 2, Table 12) may be selected. If an external LCD clock is used, the port flip-flop corresponding to DP1.3/LCCLI should remain set to avoid conflict between input drive and port output.

### LCD Oscillator

A low-power oscillator delivering a nominal frequency of 1.5 kHz is provided as a clocking source for the LCD timing. This clock can be made available on BP0/LCCLO in low-drive logic mode (see bit E0, E1 in the LCD control register).
The LCD register is stopped in disable and LCD drive 1 modes (see Table 12).

### LCD Bias Generation

The full scale LCD voltage ($V_{op}$) is obtained from $V_{DDLC} - V_{SS}$. $V_{op}$ may be temperature compensated externally through the $V_{DDLC}$ supply. Fractional LCD biasing voltages are generated by an internal voltage divider of three series connected resistors between $V_{DDLC}$ and $V_{SS}$. The center resistor can be switched out of circuit to provide a 1/2 bias voltage level (see bit B in the LCD control register).

### LCD Outputs

The LCD driver section includes 20 outputs. They are level-shifted to $V_{DDLC} - V_{SS}$ levels.
In LCD drive modes, the number of segment varies backplane drivers depends on the type of multiplexer. Segment and backplane drivers should be connected directly to the LCD. Unused segment drivers should be left open.
In disable mode, all LCD outputs are forced to $V_{SS}$, independently of the display data register file. This can be used as an efficient means to blank a connected LCD. If the disable mode is alternated with an LCD drive mode, the whole display can be blinked. The disable mode may be held indefinitely since no DC appears across the LCD.
In low-drive logic mode, up to 19 binary output lines are available. If a level-shift is not required, $V_{DDLC}$ can be tied to $V_{DD}$.

### LCD Voltage Selector

The LCD voltage selector co-ordinates the multiplexing of the LCD according to the selected multiplex and bias configurations (see bits B, M0 and M1 in the LCD control register). The preferred multiplex and bias configurations, together with their characteristics as functions of $V_{op} = V_{DDLC} - V_{SS}$ and the resulting discrimination ratio (D), are given in Table 13.

**Table 13** Preferred LCD multiplexing and bias configurations: summary of characteristics.

| LCD MULTIPLEX CONFIGURATION | LCD BIAS CONFIGURATION | $V_{off(rms)}/V_{op}$ | $V_{on(rms)}/V_{op}$ | $D = V_{on(rms)}/V_{off(rms)}$ |
|---|---|---|---|---|
| static (1 BP) | static (2 levels) | 0 | 1 | $\infty$ |
| 1:2 MUX (2 BP) | 1/2 (3 levels) | $\sqrt{2}/4 = 0.354$ | $\sqrt{10}/4 = 0.791$ | $\sqrt{5} = 2.236$ |
| 1:2 MUX (2 BP) | 1/3 (4 levels) | $1/3 = 0.333$ | $\sqrt{5}/3 = 0.745$ | $\sqrt{5} = 2.236$ |
| 1:3 MUX (3 BP) | 1/3 (4 levels) | $1/3 = 0.333$ | $\sqrt{33}/9 = 0.638$ | $\sqrt{33}/3 = 1.915$ |
| 1:4 MUX (4 BP) | 1/3 (4 levels) | $1/3 = 0.333$ | $\sqrt{3}/3 = 0.577$ | $\sqrt{3} = 1.732$ |

A practical value for $V_{op}$ is determined by equating $V_{off(rms)}$ with a defined LCD threshold voltage ($V_{th}$), typically where the LCD exhibits approximately 10% contrast. In the static drive mode a suitable choice is $V_{op} \geq 3\,V_{th}$.
Multiplex configurations of 1:3 and 1:4 with 1/2 bias are possible but the discrimination and hence the contrast ratios are smaller ($\sqrt{3} = 1.732$ for 1:3 multiplex or $\sqrt{21}/3 = 1.528$ for 1:4 multiplex). The advantage of these modes is a reduction of the LCD full scale voltage $V_{op}$ as follows:

1:3 multiplex (1/2 bias) : $V_{op} = 6\,V_{off(rms)} = 2.449\,V_{off(rms)}$

1:4 multiplex (1/2 bias) : $V_{op} = 4\sqrt{3}\,V_{off(rms)} = 2.309\,V_{off(rms)}$

These compare with $V_{op} = 3\,V_{off(rms)}$ when 1/3 bias is used.

### LCD Waveforms

The static LCD drive is used when a single backplane is provided in the LCD. Backplane and segment drive waveforms for this mode are shown in Fig.12.

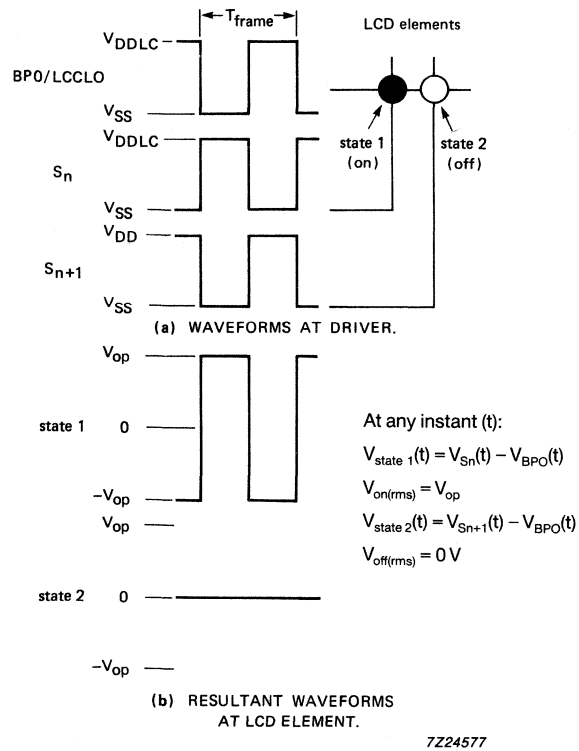



Fig.12 Static drive waveforms ($V_{op} = V_{DDLC} - V_{SS}$).

When two backplanes are provided in the LCD the 1:2 multiplex drive applies. The PCF84C633A allows use of 1/2 and 1/3 bias in this mode as shown in Figs.13 and 14.

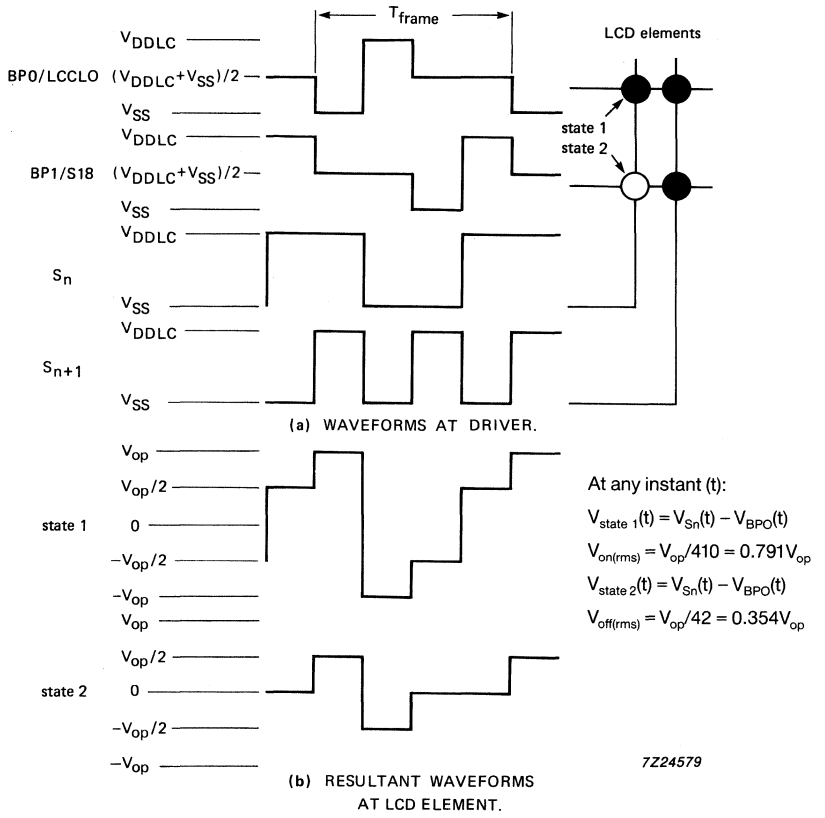



Fig.13 Waveforms for 1:2 multiplex drive with 1/2 bias ($V_{op} = V_{DDLC} - V_{SS}$).

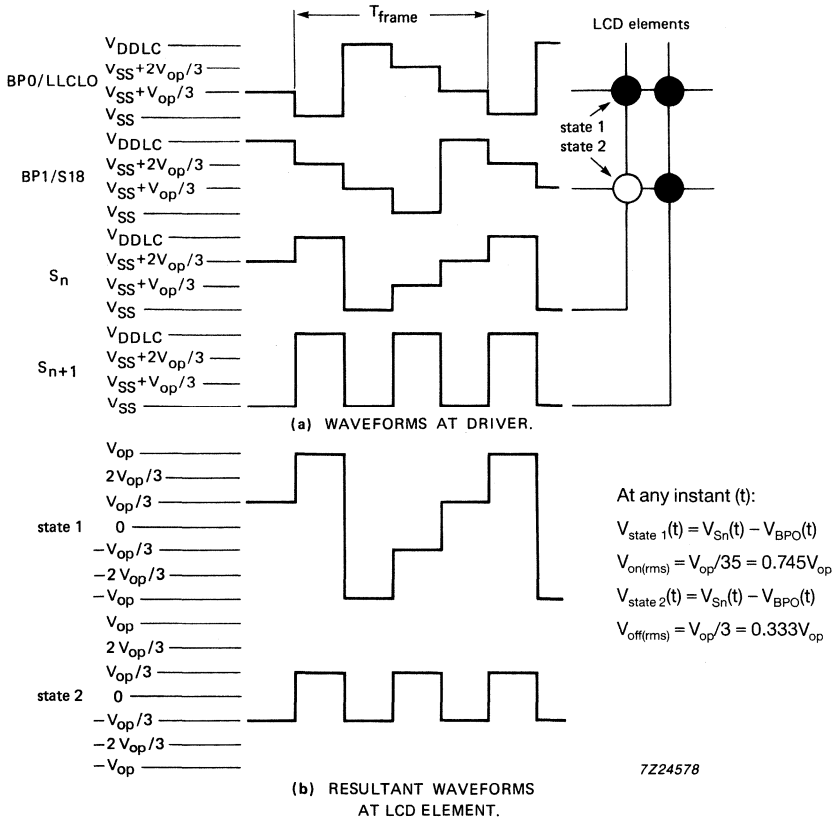



Fig.14 Waveforms for 1:2 multiplex drive with 1/3 bias.

The backplane and segment drive waveform for the 1:3 multiplex drive (three LCD backplanes) and for the 1:4 multiplex drive (four LCD backplanes) are shown in Figs.15 and 16 respectively.

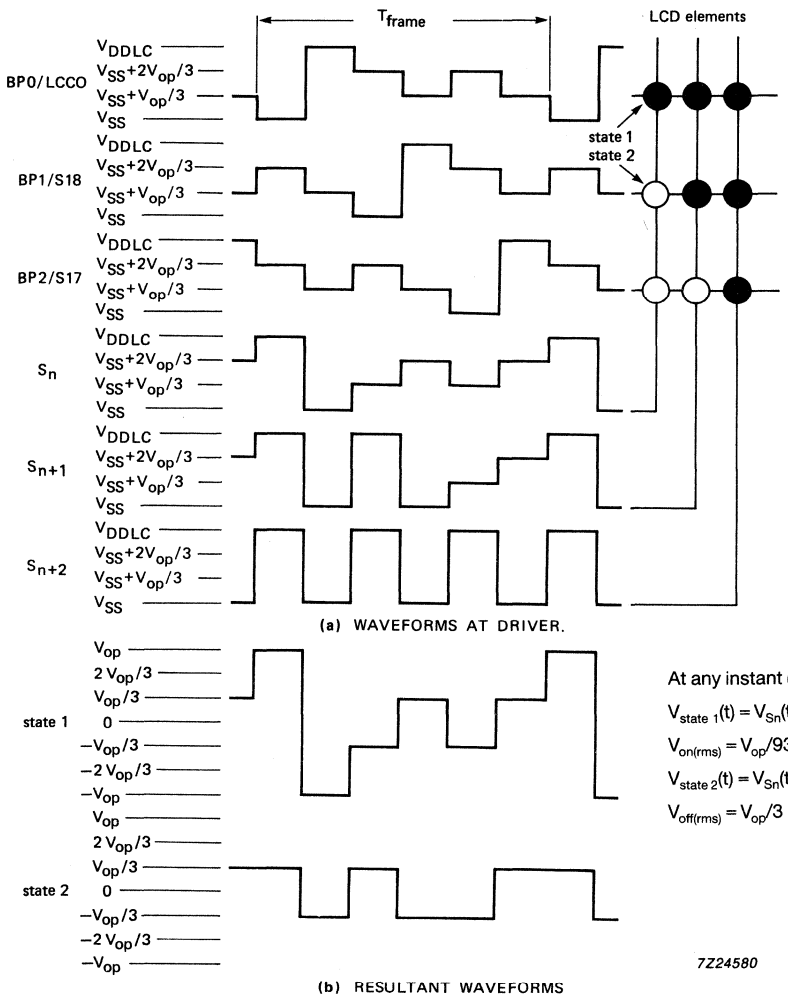



Fig.15 Waveforms for 1:3 multiplex drive ($V_{op} = V_{DDLC} - V_{SS}$).

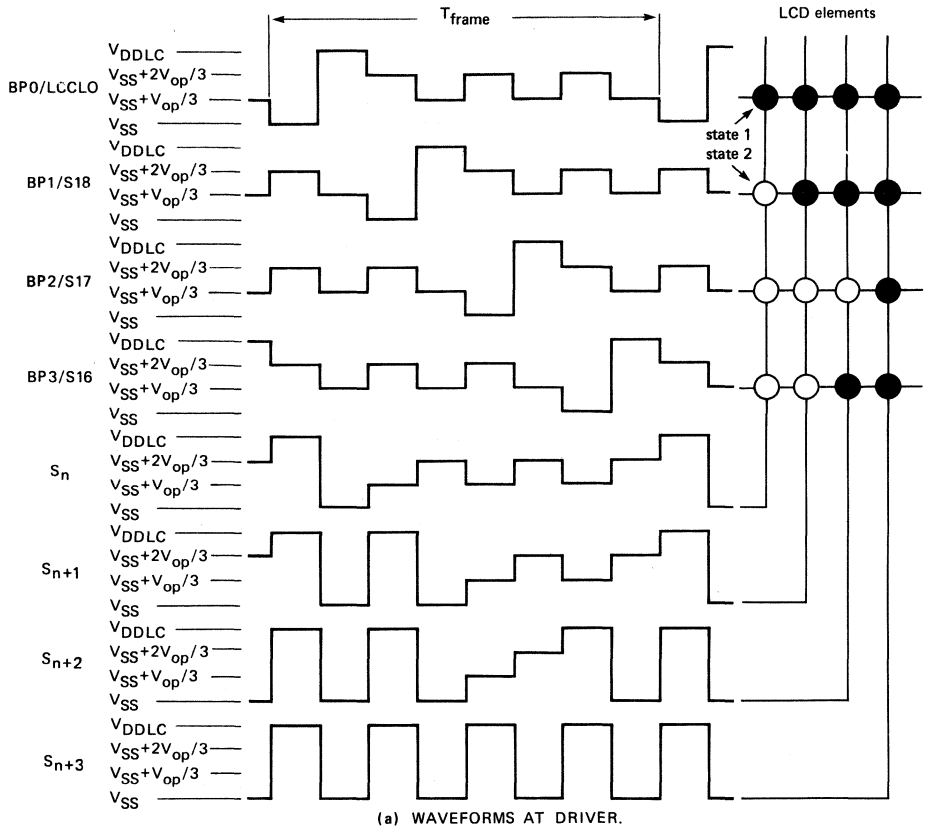
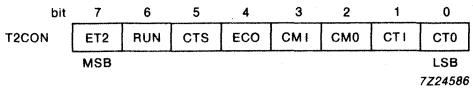
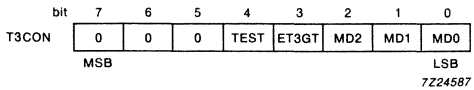


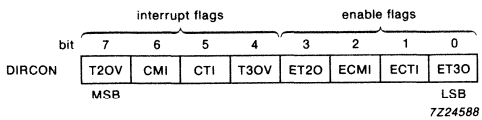

Fig.16 Waveforms for 1:4 multiplex drive ($V_{op} = V_{DDLC} - V_{SS}$).

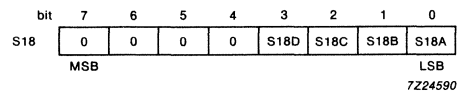
**SUMMARY OF DERIVATIVE ADDRESSES AND CONTROL REGISTERS**

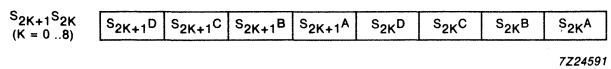
| Dx ADDRESS (HEX) | TYPE | MNEMONIC | DESCRIPTION |
|---|---|---|---|
| 00 to 01 | – | – | not used |
| 02 | R | DP1I | DP1 derivative port lines |
| 03 | R/W | DP1FF | DP1 derivative port flip-flop |
| 04 | R | T2H | T2 timer register high byte |
| 05 | R | T2L | T2 timer register low byte |
| 06 | R | T2LB | T2 low byte register |
| 07 | R/W | T2CMH | T2 compare register high byte |
| 08 | R/W | T2CML | T2 compare register low byte |
| 09 | R | T2CTH | T2 capture register high byte |
| 0A | R | T2CTL | T2 capture register low byte |
| 0B | R/W | T2CON | T2 control register |
| 0C | R/W | DIRCON | Derivative interrupt control register |
| 0D | R/W | T3H | T3 counter/timer register high byte |
| 0E | R/W | T3L | T3 counter/timer register low byte |
| 0F | R/W | T3CON | T3 control register |
| 10 to 1F | – | – | not used |
| 20 | R/W | S1S0 | Display data nibbles for S0 and S1 |
| 21 | R/W | S3S2 | Display data nibbles for S2 and S3 |
| 22 | R/W | S5S4 | Display data nibbles for S4 and S5 |
| 23 | R/W | S7S6 | Display data nibbles for S6 and S7 |
| 24 | R/W | S9S8 | Display data nibbles for S8 and S9 |
| 25 | R/W | S11S10 | Display data nibbles for S10 and S11 |
| 26 | R/W | S13S12 | Display data nibbles for S12 and S13 |
| 27 | R/W | S15S14 | Display data nibbles for S14 and S15 |
| 28 | R/W | S17S16 | Display data nibbles for S16 and S17 |
| 29 | R/W | S18 | Display data nibble for S18 |
| 2A | R/W | LCDCON | LCD control register |
| 2B to FF | – | – | not used |


bit 7 6 5 4 3 2 1 0
T2CON
ET2 RUN CTS ECO CMI CM0 CTI CT0
MSB
LSB
7Z24586


bit 7 6 5 4 3 2 1 0
T3CON
0 0 0 TEST ET3GT MD2 MD1 MD0
MSB
LSB
7Z24587


interrupt flags
enable flags
bit 7 6 5 4 3 2 1 0
DIRCON
T2OV CMI CTI T3OV ET2O ECMI ECTI ET3O
MSB
LSB
7Z24588


bit 7 6 5 4 3 2 1 0
S18
0 0 0 0 S18D S18C S18B S18A
MSB
LSB
7Z24590


S2K+1S2K
(K = 0 ..8)
S2K+1D S2K+1C S2K+1B S2K+1A S2KD S2KC S2KB S2KA
7Z24591

### TIMING

The PCF84C633 has been optimized for high speed. Allowed clock frequencies range from 1 MHz to a maximum which is a function of supply voltage. At $V_{DD} \geq 4.5$ V, a 16 MHz maximum clock frequency is guaranteed.

### OSCILLATOR

The transconductance (gm) of the inverter stage can be mask-programmed, optimizing the oscillator for a specific frequency and resonator. Three standard transconductance options, referred to as 'LOW', 'MEDIUM' and 'HIGH' gm, can be specified by the user. Table 14 is intended as a guide for typical quartz and PXE resonators.

With C1 = C2 = 8 pF on-chip, external capacitors are not required for quartz oscillators. However, for adequate frequency stability, PXE resonators need external capacitors on the order of the static resonator capacitance $C_O$, such as external C1 = C2 = 30 to 100 pF.

### RESET

The PCF84C633A does not contain a power-on reset circuit. This has the following consequences:

- RESET is an input pin only
- Passive reset always requires an external RC circuit
- When the supply voltage drops, the oscillator is not inhibited

In addition to the conditions given in the PCF84CXX family data sheet, the reset state is characterized as follows:

- all port flip-flops set to 1 (including P2.3)
- all T2 and T3 registers cleared
- derivative interrupt control register cleared
- LCD control register cleared

### IDLE MODE

In addition to the oscillator and the 8-bit programmable timer/event counter, the T2, T3 and LCD driver sections remain operative.

### STOP MODE

Since the crystal oscillator is switched off, T2, T3 and the digital filters of DP1.1/CTGT and DP1.2/CTCNT receive no clock. Therefore, the complete T2 and T3 sections are frozen. After exit from STOP mode by a LOW level on INT/T0, T2 and T3 proceed from the held state.

LCD waveforms continue during STOP mode because they are independent of the crystal oscillator. If it is permissible to blank the LCD during STOP mode, power dissipation can be further reduced by switching to the disable mode (see bits E0 and E1 in the LCD control register) before executing the STOP instruction.

In low-drive logic mode (Table 10), the levels on S0 to BP1/S18 are held while the LCD oscillator is available on BP0/LCCLO.

**Table 14** Recommended transconductance options for popular quartz and PXE resonators.

| OPTION | TYPICAL gm (mA/V) AT 5 V | $f_{osc}$ (MHz) FOR QUARTZ CRYSTAL | $f_{osc}$ (MHz) FOR PXE RESONANCE |
|---|---|---|---|
| LOW (gmL) | 0.4 | 1 to 6 | – |
| MEDIUM (gmM) | 1.2 | 4 to 12 | 1 to 5 |
| HIGH (gmH) | 4 | 10 to 16 | 3 to 16 |

## INSTRUCTION SET

Since no serial I/O interface is provided, the serial input/output instructions are not available except EN SI and DIS SI, which enable and disable respectively, the derivative interrupt.

ROM space being restricted to 6K bytes, the SEL MB3 instruction would define a non-existing program memory bank and should therefore be avoided.

## RATINGS

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

| PARAMETER | SYMBOL | MIN. | MAX. | UNIT |
|---|---|---|---|---|
| Supply voltages | $V_{DD}$,$V_{DDLC}$ | −0.8 | + 7 | V |
| All input voltages | $V_I$ | −0.5 | $V_{DD}$ + 0.5 | V |
| DC current into any input or output | $I_I$, $I_O$ | – | ±10 | mA |
| Total power dissipation | $P_{tot}$ | – | + 125 | mW |
| Storage temperature range | $T_{stg}$ | −65 | + 150 | °C |
| Operating ambient temperature range ($P_{tot} \leq 100$ mW) | $T_{amb}$ | −40 | + 70 | °C |
| Operating ambient temperature range ($P_{tot} \leq 30$ mW) | $T_{amb}$ | −40 | + 85 | °C |
| Operating junction temperature | $T_j$ | – | + 90 | °C |

## HANDLING

Inputs and outputs are protected against electrostatic discharge in normal handling. However, it is good practice to take normal precautions appropriate to handling MOS devices (see 'Handling MOS devices' Data Handbook IC14).

## DC CHARACTERISTICS

$V_{DD}$ = 2.5 to 5.5 V; $V_{DDLC}$ = 2.5 to 5.5 V; $V_{SS}$ = 0 V; $T_{amb}$ = −40 to + 85 °C; all voltages with respect to $V_{SS}$; unless otherwise specified. See Figs.17 to 31.

| PARAMETER | CONDITIONS | SYMBOL | MIN. | TYP. | MAX. | UNIT |
|---|---|---|---|---|---|---|
| Logic supply voltage operating | | $V_{DD}$ | 2.5 | – | 5.5 | V |
| LCD supply voltage | $V_{DDLC} \geq 3V$ for 1/3 bias | $V_{DDLC}$ | 2.5 | – | 5.5 | V |
| Logic supply current operating | (note 1) | | | | | |
| | $V_{DD}$ = 5 V, $f_{XTAL}$ = 16 MHz | $I_{DD}$ | – | 2.6 | 5.2 | mA |
| | $V_{DD}$ = 5 V, $f_{XTAL}$ = 10 MHz | $I_{DD}$ | – | 1.6 | 3.2 | mA |
| | $V_{DD}$ = 3 V; $f_{XTAL}$ = 3.58 MHz | $I_{DD}$ | – | 0.3 | 0.6 | mA |
| IDLE mode | (note 1) | | | | | |
| | $V_{DD}$ = 5 V; $f_{XTAL}$ = 16 MHz | $I_{DD}$ | – | 1.3 | 2.6 | mA |
| | $V_{DD}$ = 5 V; $f_{XTAL}$ = 10 MHz | $I_{DD}$ | – | 0.8 | 1.6 | mA |
| | $V_{DD}$ = 3 V; $f_{XTAL}$ = 3.58 MHz | $I_{DD}$ | – | 0.15 | 0.4 | mA |
| STOP mode | (notes 1 and 2) | | | | | |
| | LCD oscillator running, $V_{DD}$ = 2.5 V | $I_{DD}$ | – | 10 | 30 | µA |
| STOP mode | (notes 1, 2 and 3) | | | | | |
| | LCD oscillator stopped, $V_{DD}$ = 2.5 V | $I_{DD}$ | – | 5 | 10 | µA |
| LCD supply current | (note 4) | $I_{DDLC}$ | – | 10 | 20 | µA |
| **Inputs** | | | | | | |
| Input voltage LOW | | $V_{IL}$ | 0 | – | $0.3V_{DD}$ | V |
| Input voltage HIGH | | $V_{IH}$ | $0.7V_{DD}$ | – | $V_{DD}$ | V |
| Input leakage current | $V_{SS} \leq V_I \leq V_{DD}$ | $\pm I_L$ | – | – | 1 | µA |
| **Port Outputs** | | | | | | |
| Output sink current LOW | $V_{DD}$ = 5 V; $V_O$ = 0.4 V | $I_{OL}$ | 1.6 | 5.5 | – | mA |
| Pull-up output source current HIGH | $V_{DD}$ = 5 V; $V_O = 0.7V_{DD}$ | $-I_{OH}$ | 40 | – | – | µA |
| | $V_{DD}$ = 5 V; $V_O = V_{SS}$ | $-I_{OH}$ | – | – | 400 | µA |
| Push-pull output source current HIGH | $V_{DD}$ = 5 V; $V_O = V_{DD}$ − 0.4 V | $-I_{OH}$ | 1.6 | 4 | – | mA |

| PARAMETER | CONDITIONS | SYMBOL | MIN. | TYP. | MAX. | UNIT |
|---|---|---|---|---|---|---|
| **LCD Outputs** | | | | | | |
| Output sink current LOW | $V_{DDLC}$ = 5 V; $V_O$ = 0.4 V | $I_{OL}$ | 1 | 2.8 | – | mA |
| Output source current HIGH | $V_{DDLC}$ = 5 V; $V_O$ = $V_{DDLC}$ – 0.4 V | $-I_{OH}$ | 1 | 2 | – | mA |
| Output impedance (BP3/S16 to BP0/LCCLO) | $V_{DDLC}$ = 5 V and $I_{BP}$ = 100 µA Outputs measured one at a time | $R_{BP}$ | – | – | 5 | kΩ |
| Output impedance (S0 to S15) | $V_{DDLC}$ = 5 V and $I_S$ = 100 µA Outputs measured one at a time | $R_S$ | – | – | 7 | kΩ |
| DC voltage component (BP3/S16 to BP0/LCCLO) | $V_{DDLC}$ = 5 V; LCD modes | $\pm V_{BP}$ | – | 20 | – | mV |
| DC voltage component (S0 to S15) | $V_{DDLC}$ = 5 V; LCD modes | $\pm V_S$ | – | 20 | – | mV |
| Oscillator transconductance | | | | | | |
| option 'LOW' gm | $V_{DD}$ = 5 V | gmL | 0.3 | 0.4 | – | mA/V |
| option 'MEDIUM' gm | $V_{DD}$ = 5 V | gmM | 0.9 | 1.2 | – | mA/V |
| option 'HIGH' gm | $V_{DD}$ = 5 V | gmH | 3 | 4 | – | mA/V |
| Oscillator feedback resistor | | $R_f$ | 0.4 | 0.8 | 1.6 | MΩ |

**Notes to characteristics**

1. $V_{IL}$ = 0, $V_{IH}$ = $V_{DD}$; open drain outputs connected to $V_{SS}$; all other outputs open.
2. Crystal connected between XTAL1 and XTAL2; pin T1 at $V_{SS}$; pin INT/T0 at $V_{DD}$.
3. E0 = 0 in LCD control register; if E1 = 1, $f_{LCCLI}$ = 1.5 kHz.
4. $f_{LCCLI}$ = 1.5 kHz if measured in LCD drive 1 mode.

## AC CHARACTERISTICS

$V_{DD}$ = 2.5 to 5.5 V; $V_{SS}$ = 0 V; $T_{amb}$ = –40 to + 85 °C. All voltages with respect to $V_{SS}$; unless otherwise specified. See Fig.17.

| PARAMETER | CONDITIONS | SYMBOL | MIN. | TYP. | MAX. | UNIT |
|---|---|---|---|---|---|---|
| Rise time all outputs | $V_{DD}$ = 5 V, $T_{amb}$ = 25 °C, $C_L$ = 50 pF | $t_R$ | – | 30 | – | ns |
| Fall time all outputs | $V_{DD}$ = 5 V, $T_{amb}$ = 25 °C, $C_L$ = 50 pF | $t_F$ | – | 30 | – | ns |
| Clock frequency | | $f_{XTAL}$ | 1 | – | 16 | MHz |
| LCD oscillator frequency | $V_{DD}$ = 5 V | $f_{LCCLO}$ | 0.9 | 1.5 | 2.4 | kHz |

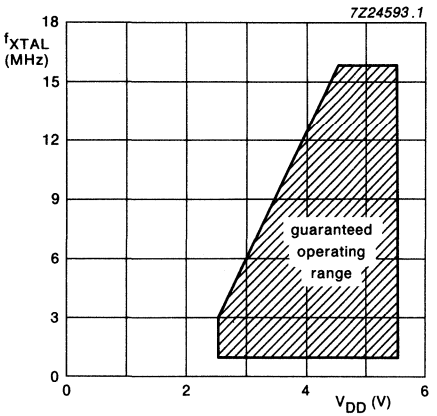



Fig.17 Maximum clock frequency ($f_{XTAL}$) as a function of logic supply voltage ($V_{DD}$).

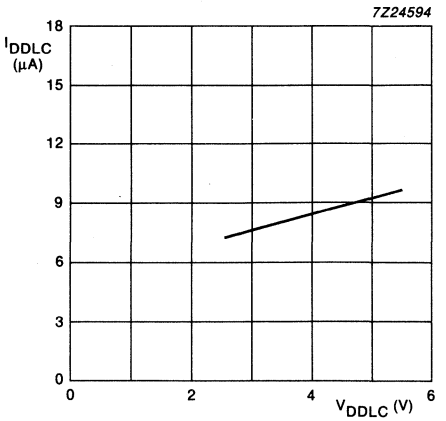
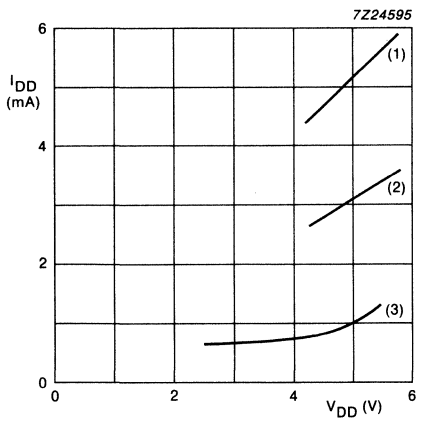



Fig.18 Typical supply current of the LCD section ($I_{DDLC}$) as a function of LCD supply voltage ($V_{DDLC}$).




(1) $f_{xtal}$ = 16 MHz
(2) $f_{xtal}$ = 10 MHz
(3) $f_{xtal}$ = 3.58 MHz

Fig.19 Maximum supply current ($I_{DD}$) in operation as a function of logic supply voltage ($V_{DD}$).

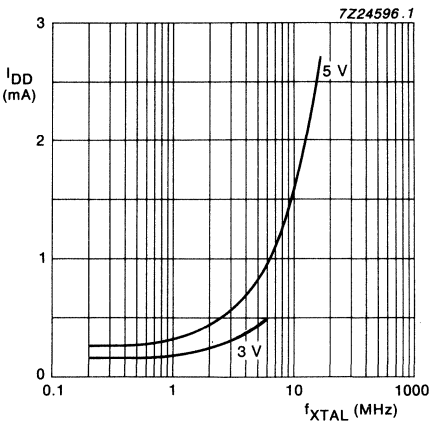
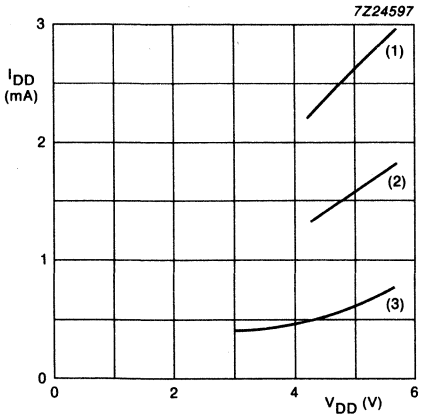



Fig.20 Typical supply current during operation as a function of frequency at $V_{DD}$ = 3 V and $V_{DD}$ = 5 V.




(1) $f_{xtal}$ = 16 MHz
(2) $f_{xtal}$ = 10 MHz
(3) $f_{xtal}$ = 3.58 MHz

Fig.21 Maximum supply current ($I_{DD}$) in IDLE mode as a function of logic supply voltage ($V_{DD}$).

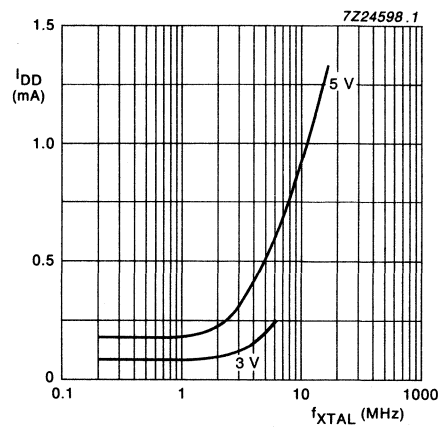



Fig.22 Typical supply current during IDLE mode as a function of frequency at $V_{DD}$ = 3 V and $V_{DD}$ = 5 V.

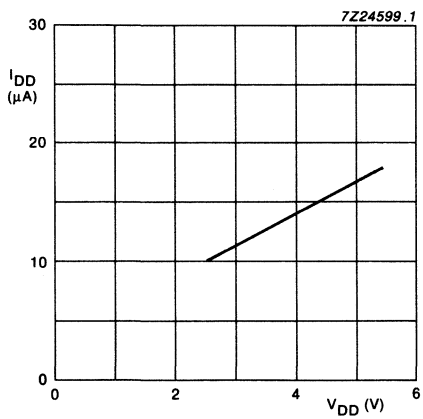



Fig.23 Typical supply current ($I_{DD}$) in STOP mode with LCD oscillator running as a function of logic supply voltage ($V_{DD}$).

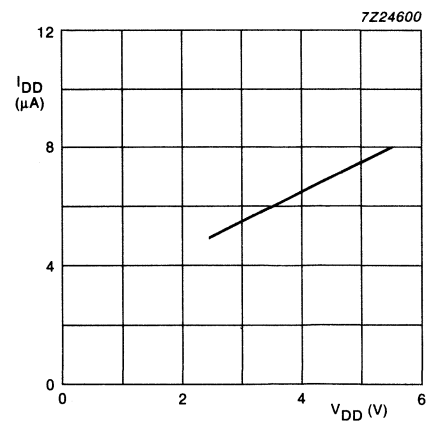



Fig.24 Typical supply current ($I_{DD}$) in STOP mode with LCD oscillator stopped as a function of logic supply voltage ($V_{DD}$).

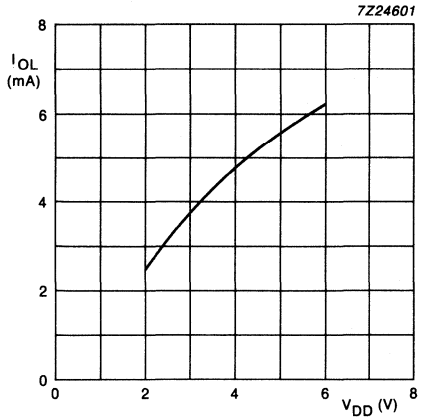
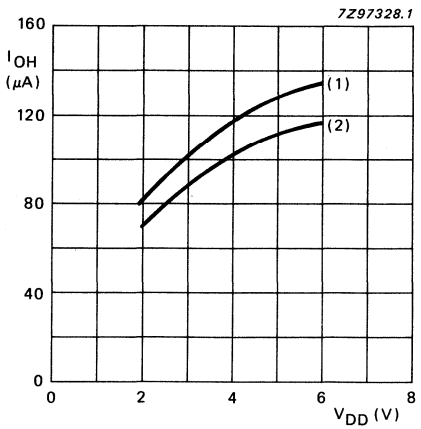



Fig.25 Typical port output sink current ($I_{OL}$) as a function of logic supply voltage ($V_{DD}$); $V_O$ = 0.4 V.




(1) $V_O = V_{SS}$
(2) $V_O = 0.7\ V_{DD}$

Fig.26 Typical port output source current ($-I_{OH}$) as a function of logic supply voltage ($V_{DD}$).

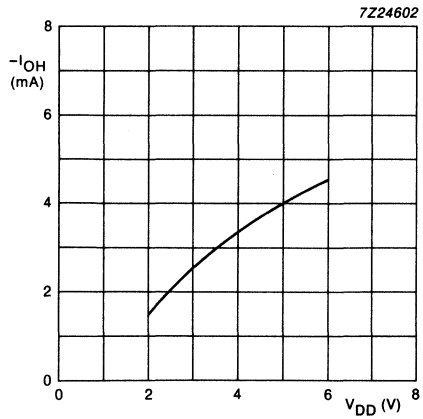



Fig.27 Typical push-pull port output source current ($-I_{OH}$) as a function of logic supply voltage ($V_{DD}$); $V_O = V_{DD} - 0.4$ V.

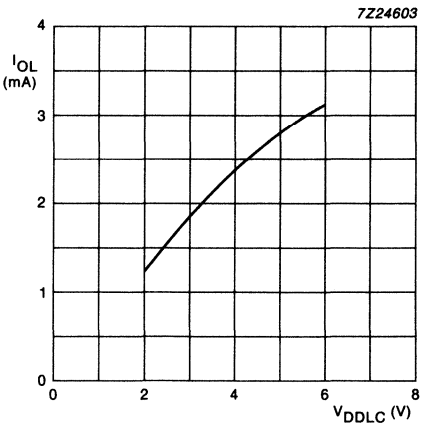



Fig.28 Typical LCD output sink current ($I_{OL}$) as a function of LCD supply voltage ($V_{DDLC}$); $V_O$ = 0.4 V.

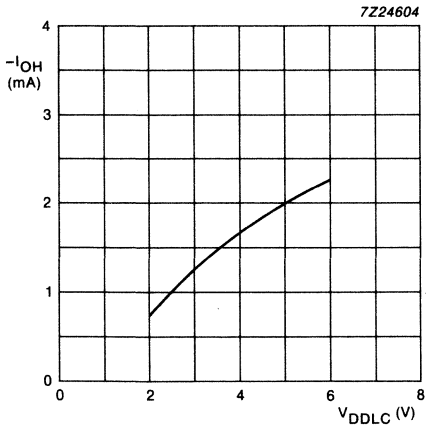



Fig.29 Typical LCD output source current ($-I_{OH}$) as a function of LCD supply voltage ($V_{DDLC}$); $V_O = V_{DDLC} - 0.4$ V.

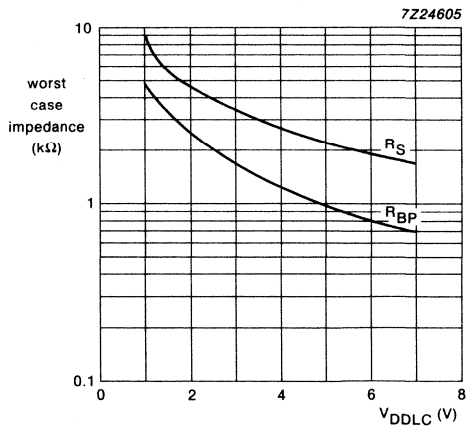



Fig.30 Typical LCD output impedance at 1/3, 1/2 and 2/3 $V_{DDLC}$ as a function of LCD supply voltage ($V_{DDLC}$).

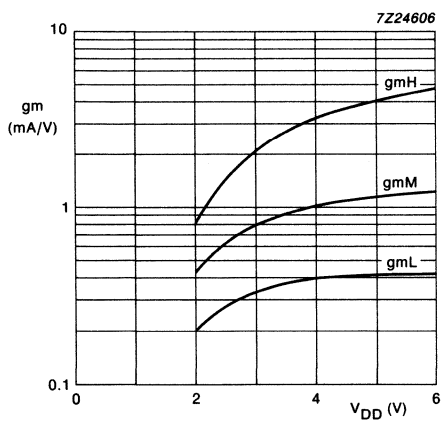



Fig.31 Typical transconductance values as a function of supply voltage ($V_{DD}$) for the options gmL, gmM and gmH.

# Microcontroller for TV tuning control and OSD application

# PCA84C846

CONTENTS

# Microcontroller for TV tuning control and OSD application

# PCA84C846

## 1 FEATURES PCF84CXXXA KERNEL

- 8-bit CPU, ROM, RAM, I/O and derivative logic in one package
- Over 80 instructions
- All instructions of 1 or 2 cycles
- Quasi bidirectional standard I/O port lines (P0, P1)
- Configuration of I/O lines individually selected by mask
- External interrupt $\overline{INT}$/T0
- 2 direct testable inputs T0, T1
- 8-bit timer/event counter
- Single level vectored interrupt: external ($\overline{INT}$), counter/timer, I²C-bus and $\overline{VSYNC}$
- On-chip oscillator clock frequency ranges 1 MHz to 10 MHz
- Power on reset and low-voltage detector
- Single power supply
- Low standby voltage and current in Idle and Stop modes
- Operating temperature range: –20 to +70 °C
- Operating voltage range: 4.5 to 5.5 V
- Configuration of optimal on-chip oscillator transconductance by mask.

## 2 VST AND OSD DERIVATIVE FEATURES

- 8K × 8 system ROM, 192 × 8 system RAM
- A multi-master I²C-bus interface
- One 14-bit PWM output for VST
- Three AFC inputs with 4-bit DAC and comparator
- Four 6-bit PWM and four 7-bit PWM outputs (DACs for analog controls)
- Eight port lines with 10 mA LED drive (at ≤ 1.2 V) capability
- Programmable $\overline{VSYNC}$/$\overline{HSYNC}$ active level
- Display RAM: 64 × 10 bit
- Display character fonts: 64 (62 customized + 2 special reserved codes)
- Display starting position: 64 different positions by software control, both vertical and horizontal
- Character size: 4 different character sizes, line-by-line basis, 1 dot = 1H/1V, 2H/2V, 3H/3V, 4H/4V. (H: OSD clock period, V: number of horizontal scan line height)
- Character matrix: 12 × 18 with no spacing between characters
- Foreground colours: 8, character-by-character basis
- Background colours: 8, word-by-word basis. Available when background is either in 'north-west shadowing', 'box shadowing' and 'frame shadowing' mode
- Background/shadowing modes: 4, no background, north-west shadowing, box shadowing, frame shadowing (raster blanking), frame basis
- Built-in oscillator for On Screen Display (OSD) function
- Character blinking rate: 1 : 1, 1 : 3, 3 : 1 (frequency: $1/16$, $1/32$, $1/64$ or $1/128$ of $f_{VSYNC}$, programmable), character basis
- Display format: flexible display format by using CR (carriage return) code
- Spacing between lines: 4 different choices, from 0, 4, 8 or 12 horizontal scan lines
- Auto display character RAM address post increment when writing data
- On-chip power-on reset
- $\overline{VSYNC}$ leading edge can generate interrupt (programmable enable/disable by software)
- 8-bit counter triggered by external pulse input.

## 3 GENERAL DESCRIPTION

The PCA84C846 is an 8-bit microcontroller with enhanced OSD and VST functions. The PCA84C846 is a member of the PCA84C640 CMOS microcontroller family. It is enhanced in display on screen function also in TV control function. It includes the PCF84CXXXA processor core, 8192 bytes of ROM, 192 bytes of RAM, a multimaster I²C-bus interface, 2 direct testable lines, 13 general purpose bidirectional I/O lines plus 16 function combined I/O lines, one 14-bit PWM 'analog' control and 3 AFC inputs (4-bit DAC + comparator) for VST, four 6-bit PWM and four 7-bit 'analog' control outputs and an enhanced on screen display (OSD) facility for flexible screen format (maximum of 64 character types).

The built-in internal oscillator for OSD operation considerably reduces the radiation generated by the RC or LC oscillator. An 8-bit timer is integrated on the chip with a 5-bit prescaler. Another 8-bit counter with Schmitt-trigger as its input for clock/timer function application.

## 4 BLOCK DIAGRAM

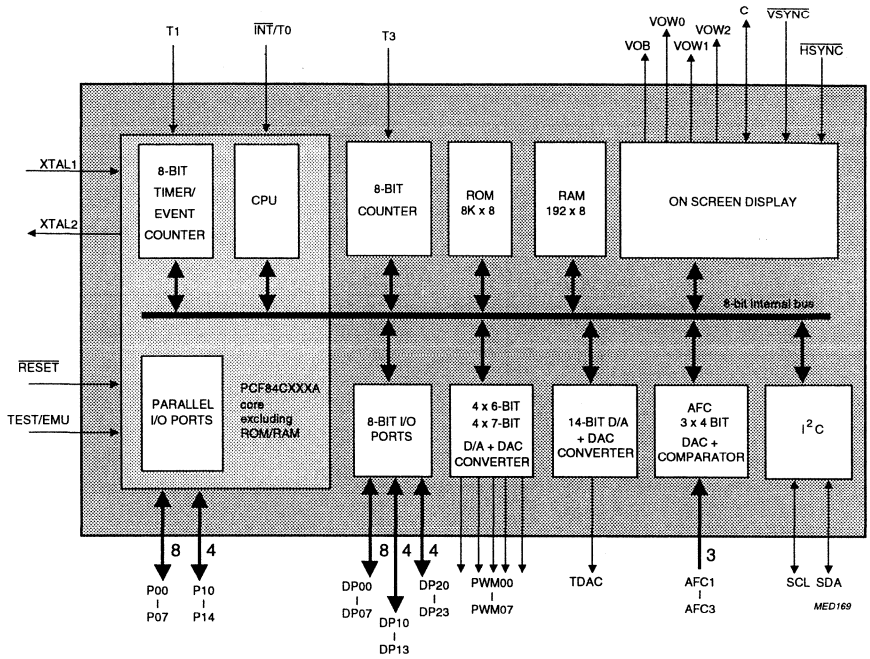



Fig.1 Block diagram.

## 5 ORDERING INFORMATION

| TYPE NUMBER | PACKAGE | | | |
|---|---|---|---|---|
| | PINS | PIN POSITION | MATERIAL | CODE |
| PCA84C846P | 42 | SDIP42 | plastic | SOT270-1 |

## 6 PINNING INFORMATION

### 6.1 Pinning

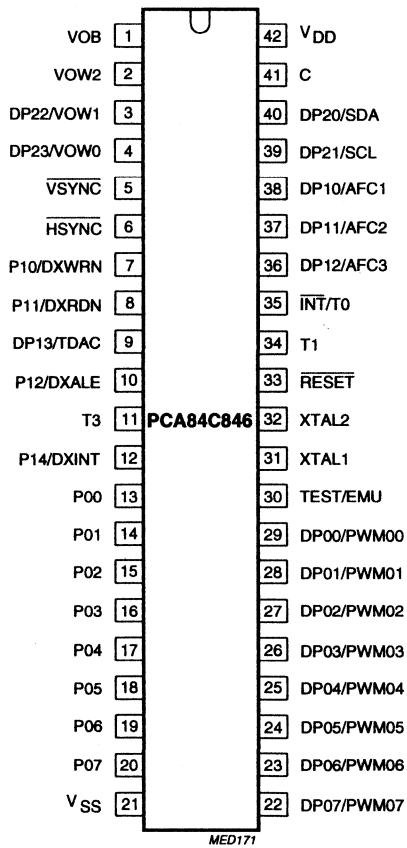



Fig.2 Pin configuration (SDIP42) PCA84C846P.

### 6.2 Pin description

**Table 1** See Fig. 2.

| SYMBOL | PIN | DESCRIPTION |
|---|---|---|
| VOB | 1 | Video fast blanking output signal. |
| VOW2 | 2 | Video character outputs or derivative port lines. |
| DP22/VOW1 | 3 | |
| DP23/VOW0 | 4 | |
| $\overline{\text{VSYNC}}$ | 5 | Vertical synchronization signal input, active LOW. |
| $\overline{\text{HSYNC}}$ | 6 | Horizontal synchronization signal input, active LOW. |
| P10/$\overline{\text{DXWR}}$ | 7 | Port line 10 or emulation $\overline{\text{DXWR}}$ signal input. |
| P11/$\overline{\text{DXRD}}$ | 8 | Port line 11 or emulation $\overline{\text{DXRD}}$ signal input. |
| DP13/TDAC | 9 | Derivative I/O port or 14-bit D/A PWM. |
| P12/DXALE | 10 | Port line 12 or emulation DXALE signal input. |
| T3 | 11 | Secondary 8-bit counter input pin (Schmitt-trigger). |
| P14/DXINT | 12 | Port line 14 or emulation DXINT signal input. |
| P00 to P03 | 13, 14, 15, 16 | General I/O port lines (10 mA). |
| P04 to P07 | 17, 18, 19, 20 | |
| $V_{SS}$ | 21 | Ground. |
| DP00 to 07/PWM00 to 07 | 22, 23, 24, 25, 26, 27, 28, 29 | Derivative I/O port, 6-bit D/A PWM (PWM04 to 07) or 7-bit PWM (PWM00 to 03). |
| TEST/EMU | 30 | Control input of testing and emulation mode, normally LOW. |
| XTAL1 | 31 | Oscillator input terminal for system clock. |
| XTAL2 | 32 | Oscillator output terminal for system clock. |
| $\overline{\text{RESET}}$ | 33 | Initialize input, active LOW. |
| T1 | 34 | Direct testable pin and event counter input. |
| $\overline{\text{INT}}$/T0 | 35 | External interrupt/direct testable pin. |
| DP12/AFC3 | 36 | Derivative I/O port or comparator input with 4-bit D/A. |
| DP11/AFC2 | 37 | |
| DP10/AFC1 | 38 | |
| DP21/SCL | 39 | Derivative port line or $I^2C$-bus clock line. |
| DP20/SDA | 40 | Derivative port line or $I^2C$-bus data line. |
| C | 41 | External capacitor input for on chip PLL OSD oscillator. |
| $V_{DD}$ | 42 | Power supply. |

## 7 RESET

The $\overline{\text{RESET}}$ pin is used as an active-LOW initialize input. A passive reset is generated by the RC-circuit in Fig.3.

To avoid overload of the internal diode, an external diode should be added in parallel if $C_{RESET} > 0.2\ \mu F$.

An active reset can be generated by driving the $\overline{\text{RESET}}$ pin from an external logic device. Such an active reset pulse should not fall off before $V_{DD}$ has reached its $f_{xtal}$ dependent on minimum operating voltage.

### 7.1 Reset trip level

The $\overline{\text{RESET}}$ trip-voltage level is masked to 1.3 V in the PCA84C846.

### 7.2 Reset status

- Derivative register status (see register map Fig.58 for details)
- Program counter 0
- Memory bank
- Register bank 0
- Stack pointer 0
- All interrupt disabled
- Timer/event counter 1 stopped and cleared
- Timer prescaler modulo-32 (PS = 0)
- Timer flag cleared
- Serial I/O interface disabled (ESO = 0) and in slave receiver mode
- Idle and Stop mode cleared.

## 8 ANALOG CONTROL: 6-BIT/7-BIT PWM DACS

The PCA84C846-VST has eight PWM DAC outputs for analog controls of e.g. volume, balance, brightness and saturation. These PWM outputs generate pulse patterns with a repetition rate of $1/64 f_{PWM}$ or $1/128 f_{PWM}$. The analog value is determined by the ratio of the HIGH-time and the repetition time. A DC voltage proportional to the PWM control setting is obtained by means of an external integration network (low-pass filter). PWM00 to 03 is 7-bit while PWM04 to 07 is 6-bit.

The polarity of the PWMn output is selected by bit P6LVL/P7LVL (in derivative register 23H, see register map Fig 58.), this P6LVL/P7LVL (polarity control bit) allows you to invert the PWMn outputs. Setting P6LVL/P7LVL to 1 inverts all eight PWMn outputs. If P6LVL/P7LVL = 0 then PWMn outputs are not inverted.

The high time of a PWMn output is:

$t_{HIGH} = [PWMnDL] \times t_0$

where [PWMnDL] is the 'contents of PWMn data latch' (derivative register 10 to 17, see Fig.58) and

$t_0 = 1/f_{PWM}$ (in which $1/3 f_{xtal} = f_{PWM}$)

Figure 4 shows non-inverted PWM output patterns (with P6LVL/P7LVL = 0). The PWMn output shares the pin with a DP0n I/O line under control of a PWMnE enable bit. If PWMnE = 0 then the DP0n/PWM terminal can be used as a DP0n I/O pin. If PWMnE = 1 then the DP0n/PWM terminal can be used as a PWMn output pin.

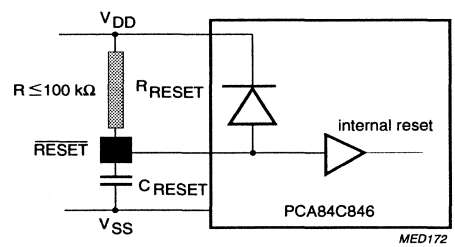



Fig.3 External components for $\overline{\text{RESET}}$ pin.

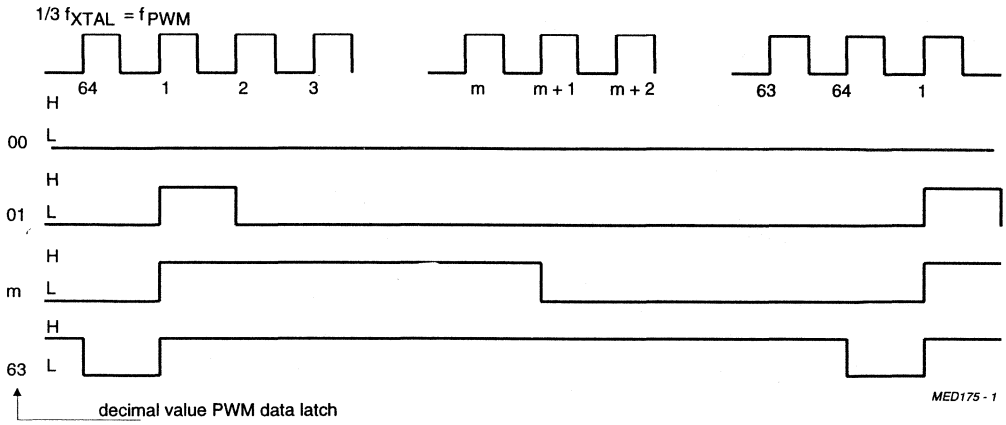



Fig.4 Example PWMn (6-bit) output patterns (H = HIGH level, L = LOW level, P6LVL = 0).

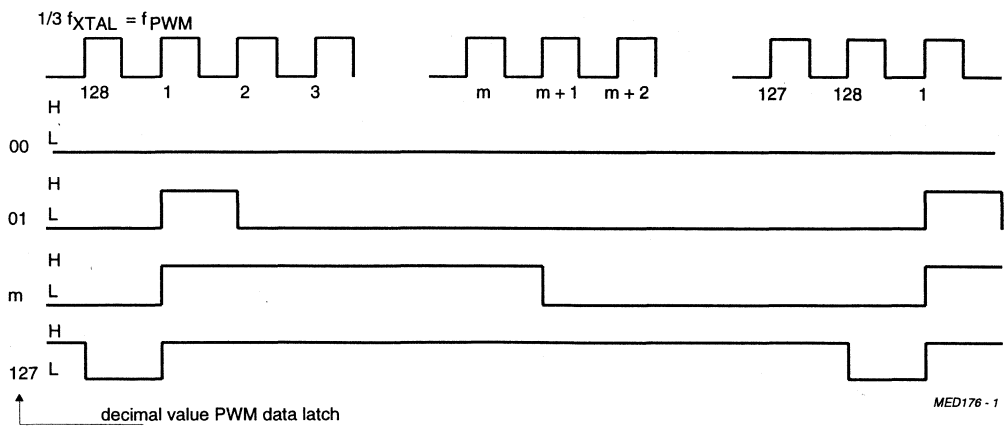



Fig.5 Example PWMn (7-bit) output patterns (H = HIGH level, L = LOW level, P7LVL = 0).

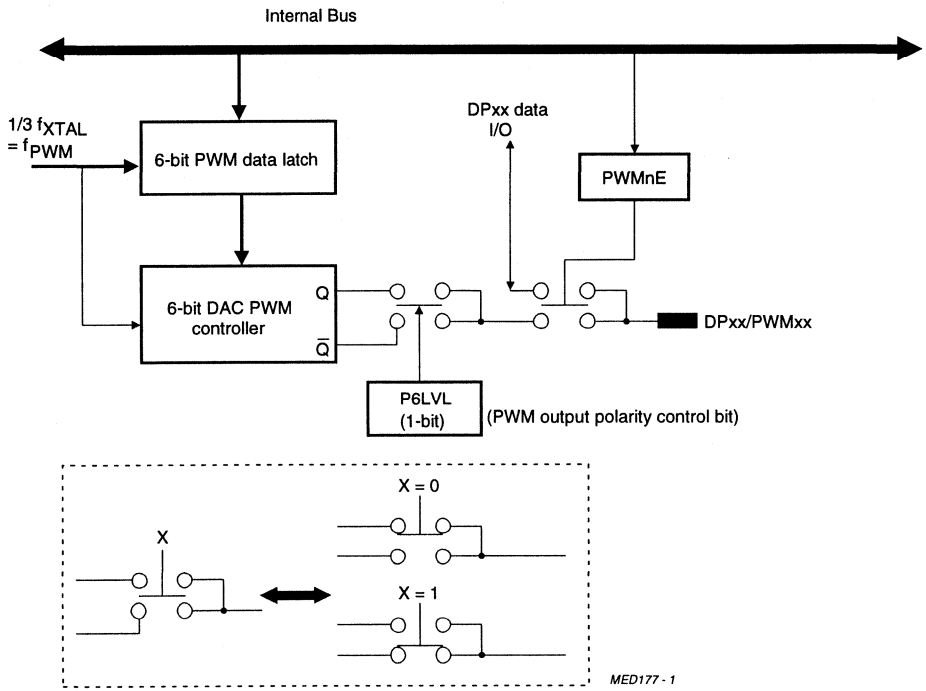



Fig.6 Block diagram of 6-bit PWM DAC.

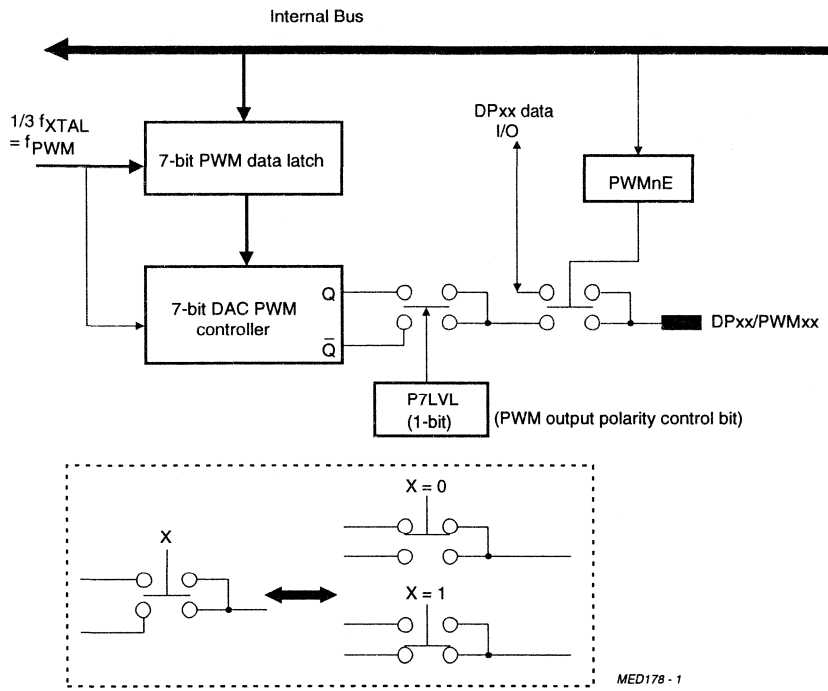



Fig.7 Block diagram of 7-bit PWM DAC.

## 9 VST CONTROL 14-BIT PWM DAC

The PCA84C846 has one PWM DAC output with a resolution of 16384 levels for voltage synthesized tuning, the VST DAC (see Fig.8).

The PWM DAC consists of:

- Latches (VSTH, VSTL, derivative registers 18H and 19H; see Fig.58)
- Two 7-bit DAC interface latches (VSTH, VSTL)
- One 14-bit DAC data latch (VSTREG)
- 14 bit counter
- Pulse control.

The contents of the data latch (VSTREG) defines the high time. This content is transferred from the interface latches (VSTH, VSTL) to the data latch (VSTREG). The upper seven bits are used for coarse adjustment, while the lower seven bits are used for fine adjustment.

Repetition time:

- Upper 7-bits (VSTH): $f_{TH} = {}^{1}/_{128} f_{TDAC}$
- Lower 7-bits (VSTL): $f_{TL} = {}^{1}/_{128} f_{TH} = {}^{1}/_{16384} f_{TDAC}$.

The data from 'VSTH', 'VSTL' is latched into 'VSTREG' at the beginning of the first $T_{sub}$ after 'VSTL' is written.

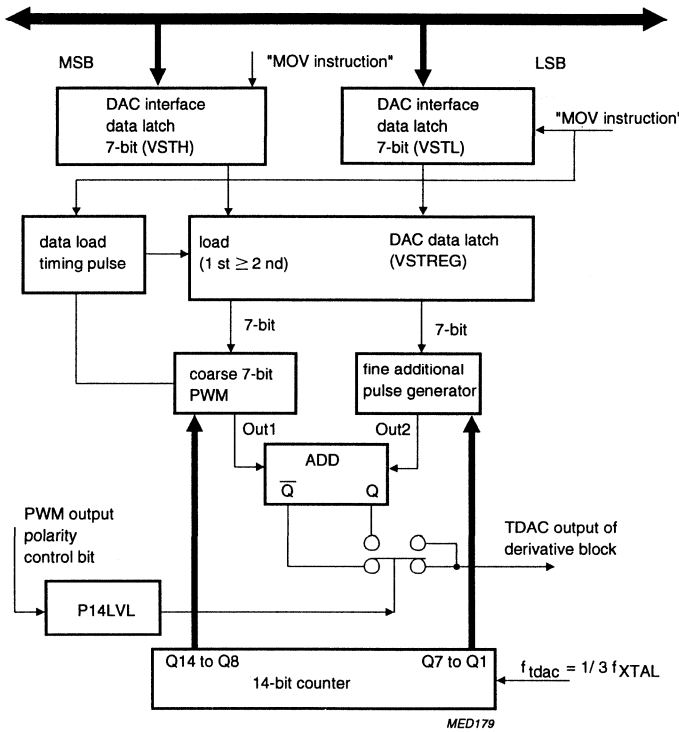
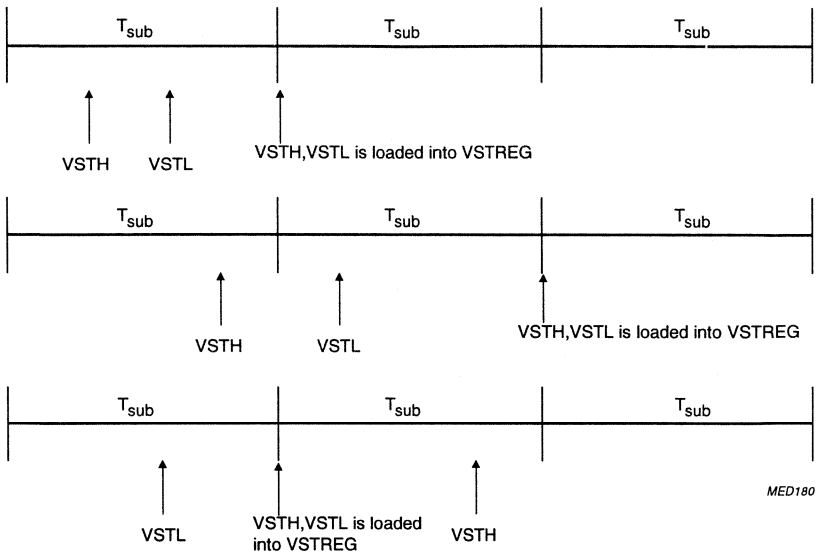



Fig.8 Block diagram of 14-bit PWM DAC.

After VSTH, VSTL is latched into VSTREG, it takes one $T_{sub}$ to generate the appropriate pulse pattern. So to ensure correct D/A conversion, one should wait for two $T_{sub}$ before changing VST(H)(L) again. So, in order to be sure that the correct data is latched into 'VSTREG', 'VSTH' must contain the correct value before 'VSTL' is written, (see the note in Fig.9).

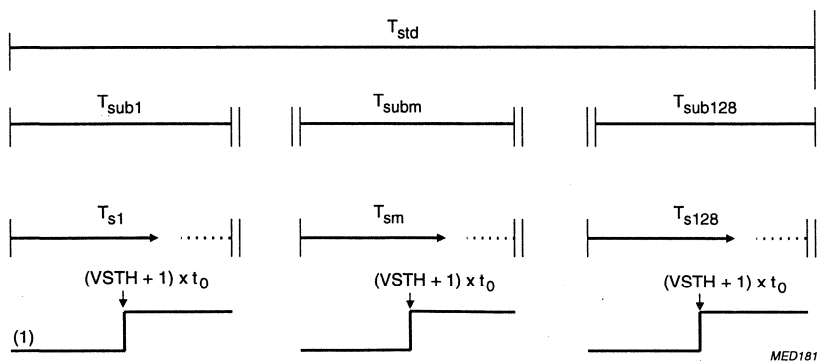
### 9.1 Coarse adjustment (VSTH)

Coarse adjustment (Fig.10) is carried out in periods of $T_{sub}$. During 1 $T_{sub}$, the coarse adjustment output (Out1 of Fig.8) is low while $t_S \leq (VSTH) \times t_0$. The coarse adjustment output is high while $(VSTH) \times t_0 < t_S < 128 \times t_0$. Where $t_S$ is the time within $T_{sub}$.




In case 1 and case 2, a new value for VSTH, VSTL is latched into VSTREG. In case 3, VSTL, together with an old value of VSTH are latched into VSTREG).

Fig.9 Latching VSTH, VSTL into VSTREG.




(1) (VSTH) = contents of register VSTH.

Fig.10 Coarse adjustment output Out1.

### 9.2 Fine adjustment (VSTL)

Fine adjustment is carried out with additional pulses, generated in particular sub-periods. The contents of VSTL indicates in which periods an additional pulse is generated. During $T_{std}$, 0 (VSTL = 1111 111) to 127 (VSTL = 0000 000) additional pulses can be generated. Table 2 shows the relation between the contents of the VSTL and the sub-periods ($T_{sub}$) during which an additional pulse is generated (pulse distribution).

If, for example VSTH = 0011101, VSTL = 1111101 and P14LVL = 0, then TDAC is as shown in Fig.13. The additional pulses are generated in sub-periods of 32 and 96.

TDACE bit (in derivative register 22H, see Fig.58) is the bit to control the DP13/TDAC pin as DP13 function (= 0) or as 14-bit PWM output (= 1).

**Table 2** Additional pulse distribution.

| LOWER 7-BIT DATA (VSTL) | ADDITIONAL PULSE IN SUB PERIODS $T_{subn}$ | BINARY POSITION[1] |
|---|---|---|
| 111 1110 | 64 | 1 0 0 0 0 0 0 |
| 111 1101 | 32, 96 | X 1 0 0 0 0 0 |
| 111 1011 | 16, 48, 80, 112 | X X 1 0 0 0 0 |
| 111 0111 | 8, 24, 40, 56, 72, 88, 104, 120 | X X X 1 0 0 0 |
| 110 1111 | 4, 12, 20, 28, 36, 44, 52, 60....116, 124 | X X X X 1 0 0 |
| 101 1111 | 2, 6, 10, 14, 18, 22, 26, 30....122, 126 | X X X X X 1 0 |
| 011 1111 | 1, 3, 5, 7, 9, 11, 13, 15, 17....125, 127 | X X X X X X 1 |

**Note**

1. X = don't care.

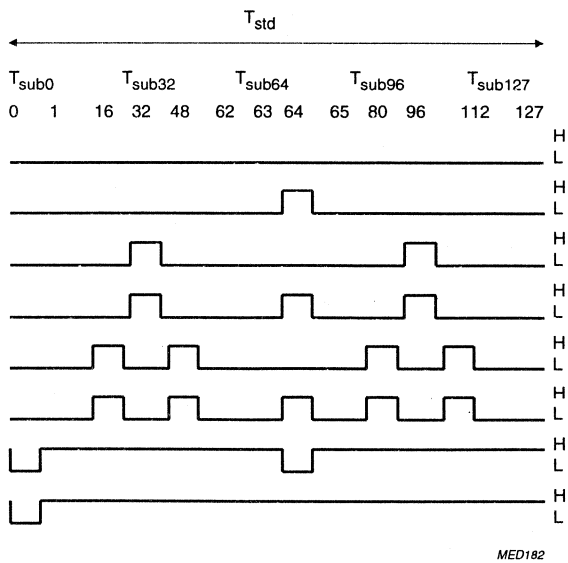



Fig.11 Lower 7-bit D/A converter output pulse (Out2).

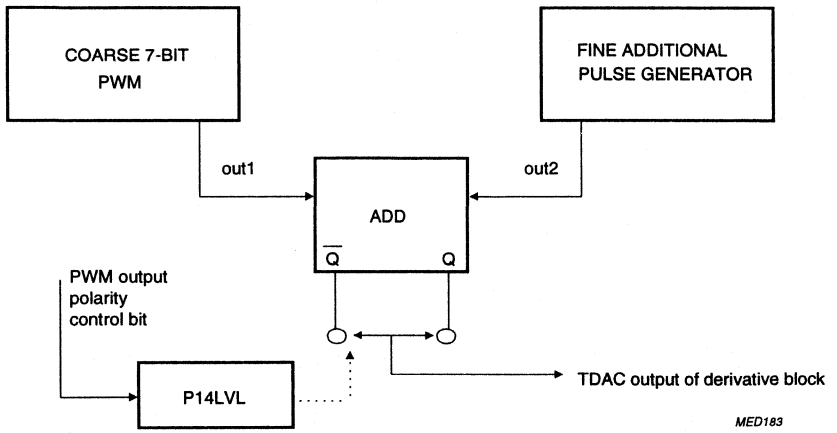
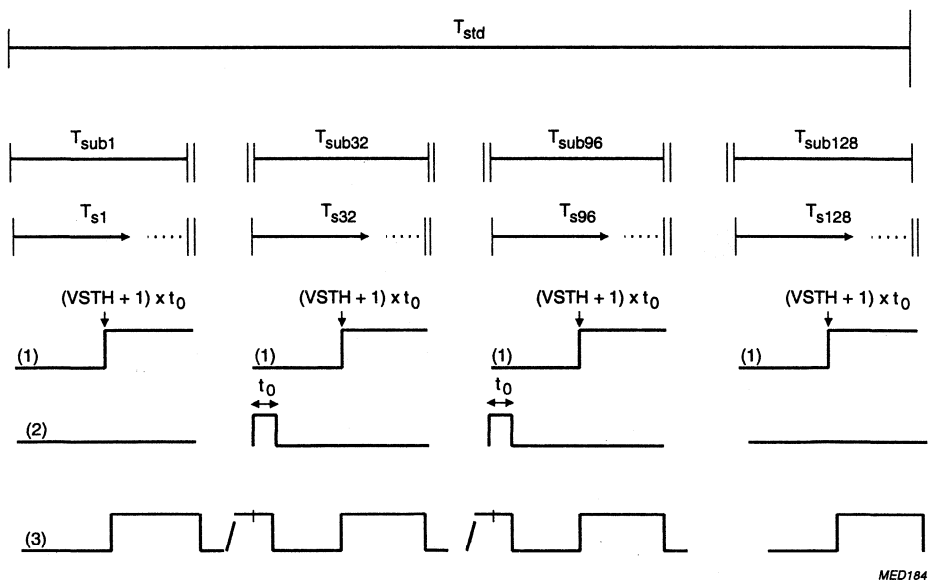



Fig.12 How a TDAC signal is built up (see also Fig.8).




(1) Coarse output (Out1, Fig.8)
(2) Fine adjustment output (Out2, Fig.8)
(3) TDAC output (Out1, Out2)

Fig.13 Example TDAC output pulse where VSTH = 0011101 and VSTL = 1111101, additional pulses in sub periods 32 and 96.

## 10 AFC INPUT

The AFC input is intended to measure the level of the Automatic Frequency Control signal. This is done by comparing the AFC signal with the output of a 4-bit D/A converter as shown in Fig.14. The DAC analog switches select one of the 16 resistor taps that is connected between $V_{DD}$ and $V_{SS}$ (controlled by AFC3, AFC2, AFC1, AFC0 bits of derivative register 20H). The AFCC signal (in derivative register 20H) then can be tested to check whether the AFC input is higher or lower than the DAC level.

The AFC input shares the pin with a DP10, DP11 and DP12 I/O lines under control of the AFCE0, AFCE1, AFCE2 bits (AFC enable/disable bit, in derivative register 22H).

AFCEi (i = 0, 1, 2) '0', disable the corresponding comparator and 4-bit D/A, enable DP1i (i = 0, 1, 2) I/O buffer.

AFCEi (i = 0, 1, 2) '1', enable the corresponding comparator and 4-bit D/A, disable DP1i (i = 0, 1, 2) I/O buffer.

AFCH1, AFCH0 (in derivative register 20H) select one out of three AFC input to the comparator as shown in Table 3 (the corresponding AFCEi must be enabled to get the correct comparison).

**Table 3** AFC input selection by bits AFCH1 and AFCH0.

| AFCH1 | AFCH0 | SELECT |
|---|---|---|
| 0 | 0 | AFC Channel 0 |
| 0 | 1 | AFC Channel 1 |
| 1 | 0 | AFC Channel 2 |
| 1 | 1 | reserved |

Conversion time of AFC is larger than 6 µs while smaller than 9 µs. A 'NOP' instruction is recommended to add in between the instruction which changes the reference voltage or channel and the instruction which reads the AFCC register bit.

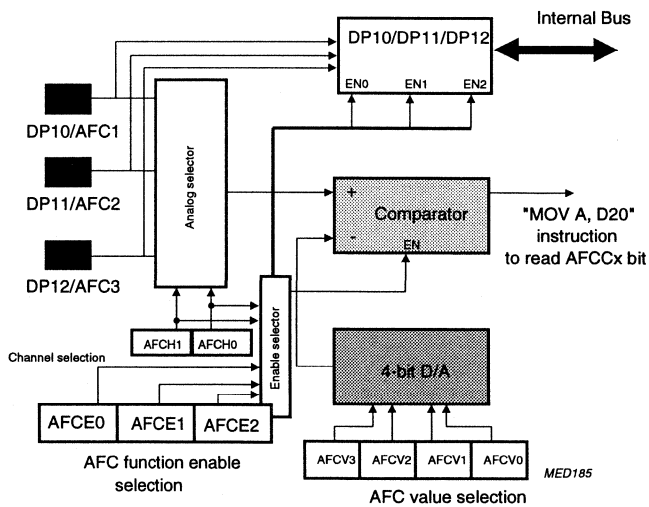



Fig.14 AFC circuit.

**Table 4** $V_{ref}$ as a function of AFC3 to AFC0.

| AFC3 | AFC2 | AFC1 | AFC0 | $V_{ref}$ | $V_{ref}$ ($V_{DD}$ = 5.0 V) |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | $V_{DD} \times 1/16$ | 0.31 V |
| 0 | 0 | 0 | 1 | $V_{DD} \times 2/16$ | 0.62 V |
| 0 | 0 | 1 | 0 | $V_{DD} \times 3/16$ | 0.93 V |
| 0 | 0 | 1 | 1 | $V_{DD} \times 4/16$ | 1.25 V |
| 0 | 1 | 0 | 0 | $V_{DD} \times 5/16$ | 1.56 V |
| 0 | 1 | 0 | 1 | $V_{DD} \times 6/16$ | 1.87 V |
| 0 | 1 | 1 | 0 | $V_{DD} \times 7/16$ | 2.18 V |
| 0 | 1 | 1 | 1 | $V_{DD} \times 8/16$ | 2.50 V |
| 1 | 0 | 0 | 0 | $V_{DD} \times 9/16$ | 2.81 V |
| 1 | 0 | 0 | 1 | $V_{DD} \times 10/16$ | 3.12 V |
| 1 | 0 | 1 | 0 | $V_{DD} \times 11/16$ | 3.43 V |
| 1 | 0 | 1 | 1 | $V_{DD} \times 12/16$ | 3.75 V |
| 1 | 1 | 0 | 0 | $V_{DD} \times 13/16$ | 4.06 V |
| 1 | 1 | 0 | 1 | $V_{DD} \times 14/16$ | 4.37 V |
| 1 | 1 | 1 | 0 | $V_{DD} \times 15/16$ | 4.68 V |
| 1 | 1 | 1 | 1 | $V_{DD}$ | 5.00 V |

## 11 INPUT/OUTPUT

There are 3 different port options available for the port pins in the 84CXXX derivatives (see Figs 15, 16 and 17).
The output stage consists of 4 transistors:

TR1: N - channel transistor for 'sink'

TR2: P - channel transistor for 'boost-up'

TR3: P - channel transistor for 'pull-up'

TR4: P - channel transistor for 'constant current'.

See Tables 7 and 8 for PCA84C846 possible port option list.

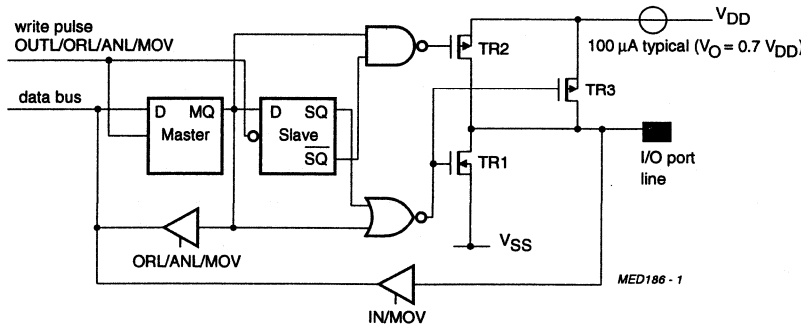



Fig.15 Standard output with switched pull-up current source (Option 1).

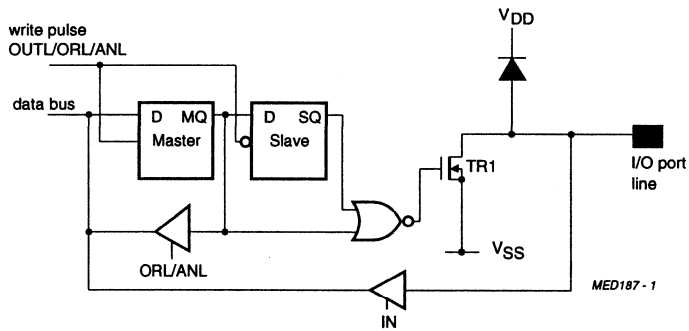



Fig.16 Open drain output (Option 2).

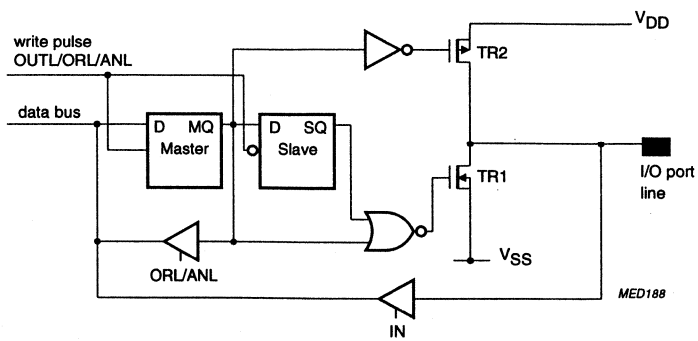



Fig.17 Push-pull output (Option 3).

## 12 OSD (ON-SCREEN DISPLAY) FUNCTION

### 12.1 Features

- Display RAM: 64 (characters) × 10 (bits)
- Display character fonts: 64 (in which 62 customized + 2 special reserved codes)
- Display starting position of the first character: 64 different position by software control, both vertical and horizontal
- Character size: 4 different character size, line-by-line basis, 1 dot = 1H/1V, 2H/2V, 3H/3V, 4H/4V
- Character matrix: 12 × 18 with no spacing between characters
- Foreground colours: 8, combination of Red, Green, Blue; character-by-character basis
- Background/shadowing modes: 4, 'no background', 'box shadowing', 'north west shadowing', 'frame shadowing (raster blanking)', frame basis
- Background colours: 8, combination of Red, Green, Blue; word-by-word basis. Available when background mode is either in 'box shadowing' or 'north west shadowing' and 'frame shadowing' mode
- Built-in on-chip OSD oscillator
- Character blinking rate: 1 : 1, 1 : 3, 3 : 1 (frequency: $^1/_{16}$, $^1/_{32}$, $^1/_{64}$ or $^1/_{128}$ of $f_{VSYNC}$, programmable, e.g. NTSC: $^{60}/_{16}$ Hz, PAL: $^{50}/_{64}$ Hz etc.); character basis
- Display format: flexible display format by using CR (carriage return) code, maximum number of characters per line is flexible and depending on the OSD clock
- Spacing between lines: 4 different choices from 0, 4, 8 or 12 horizontal scan lines
- Display character RAM auto-address-post-increment when writing data
- Programmable HSYNC and VSYNC active input polarity
- Programmable G (VOW1), B (VOW2), R (VOW0) and FB (VOB) output polarity.

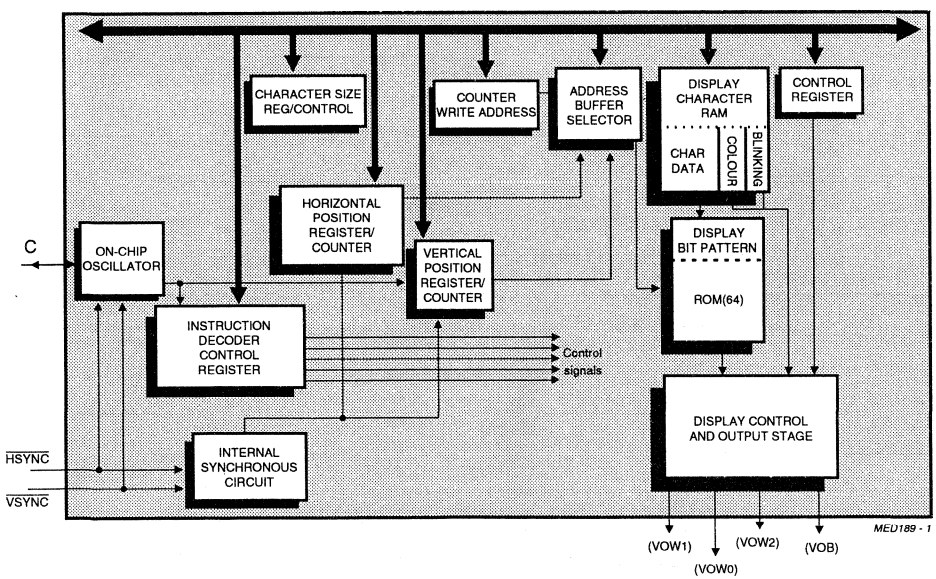



Fig.18 OSD block diagram.

### 12.2 Horizontal display position control

The horizontal position counter is increased every OSDC (OSD clock) cycle after the programmed level of HSYNC occurs at the HSYNC pin and is reset when the opposite polarity of the HSYNC is reached.

Horizontal start position is controlled by derivative register 36H (see Fig.58).

The starting position is calculated as:

$HPx = [4 \times (H5\ to\ nH0)_{DECIMAL} + 5]$ (OSD clock)

in which $(H5\ to\ H0) \geq 10_{DECIMAL}$.

### 12.3 Vertical display position control

The vertical position counter is increased every HSYNC cycle and is reset by the VSYNC signal.

Vertical start position is controlled by derivative register 35H (see Fig.58).

The starting position is calculated as:

$VPx = [4 \times (V5\ to\ V0)_{DECIMAL}]$ (horizontal scan lines)

in which $(V5\ to\ V0) \geq 0_{DECIMAL}$.

### 12.4 Clock generator

The clock generator is made of a PLL (Phased Lock Loop) circuitry. The VCO (Voltage Controlled Oscillator) outputs a constant clock with frequency ranges in between 8 MHz to 20 MHz (see Fig.20).

The input frequency, $f_1$, is the normal HSYNC frequency. Depending on the value programmed in the 7-bit counter:

the $f_{VCO} = f_1 \times 16 \times$ (value of 7-bit counter)

with $16_{DECIMAL} <$ (value of 7-bit counter) $< 48_{DECIMAL}$.

The value 16 is the 4-bit prescaler which gives the output of the VCO to increase/decrease in step of $16 \times f_1$ (Hz). Given an example of HSYNC = 15.750 kHz,
The $f_{VCO}$ is then increased or decreased in steps of $16 \times 15.750$ kHz = 252 kHz = 0.25 MHz.

The $f_{VCO}$ is then feed into the buffer to generate the DOSC (OSD dot clock frequency) signal which gives the DOSC in a range of 4 to 12 MHz. Decreasing the OSDC frequency gives broader characters. Typical recommended OSD clock frequency is 6 to 8 MHz.

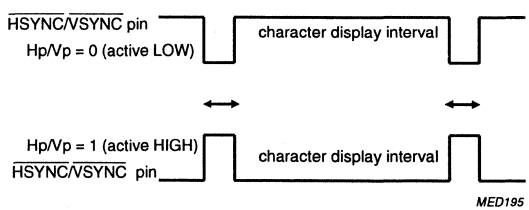
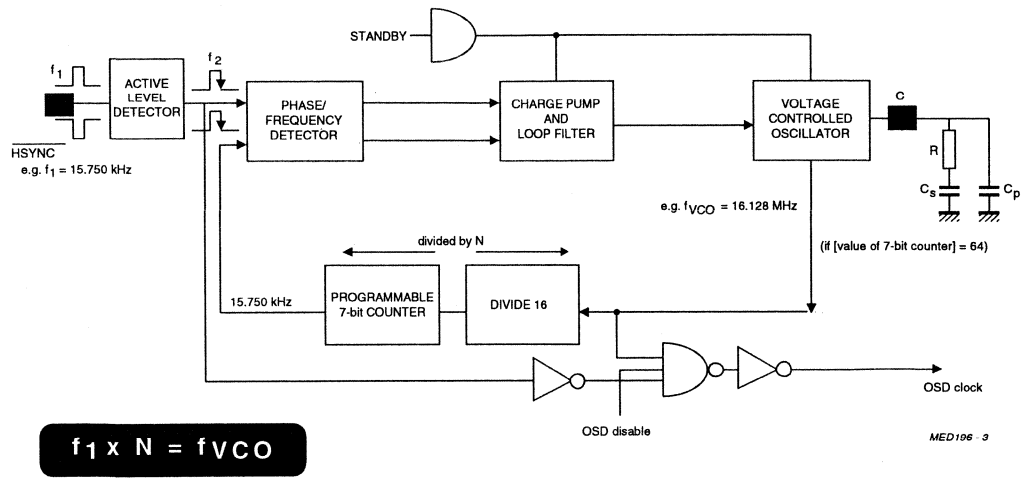



Fig.19 Hp/Vp bits to decide the active level of HSYNC/VSYNC signal.




R = 3.9 kΩ to 15 kΩ; typ. 10 kΩ
$C_s$ = 100 nF to 470 nF; typ. 220 nF
$C_p = 1/10\ C_s$

Fig.20 On-chip OSD oscillator.

The VCO is synchronized with the HIGH-to-LOW edge of the $f_2$ clock. The programmable active level detector is to either pass the $\overline{\text{HSYNC}}$ signal (if $\overline{\text{HSYNC}}$ is active HIGH) or to invert the $\overline{\text{HSYNC}}$ signal (if the $\overline{\text{HSYNC}}$ is active LOW). It makes $f_2$ always active HIGH.

Figure 20 illustrates the block diagram of on-chip OSD clock generator circuitry.

The on-chip oscillator is always active after power on. No large current is sourced or sunk in and out of the clock pin as the RC or LC type of oscillator normally is, the radiation generated by the OSD clock source can be reduced considerably.

Disabling of the OSD circuitry is controlled by the bit 0 of derivative register 34H. When the OSD circuitry is disabled, the DOSC is not active (kept LOW) while the on chip OSD clock generator is still active. This would reduce the transient time needed from the clock being stopped up to it can lock into the external $\overline{\text{HSYNC}}$ signal.

The 7-bit PLL programmable counter control register is in 25H (see Fig.58).

## 13 DISPLAY RAM ORGANIZATION

The display character RAM is organized as 64 × 10-bits. See Fig.21 for details. Bit < 9 : 4 > is the character data which determines 1 out of 64 different fonts; (62 customized + 2 reserved codes, i.e. carriage return code and space code).

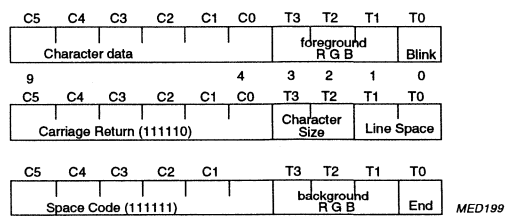



Fig.21 Display RAM organization overview.

| VOW2 | VOW1 | VOW0 | PIN |
|---|---|---|---|
| T3 | T2 | T1 | |
| R | G | B | colour |
| 0 | 0 | 0 | black |
| 0 | 0 | 1 | blue |
| 0 | 1 | 0 | green |
| 0 | 1 | 1 | cyan |
| 1 | 0 | 0 | red |
| 1 | 0 | 1 | magenta |
| 1 | 1 | 0 | yellow |
| 1 | 1 | 1 | white |

| T0 |
|---|
| Blink |
| 0 : No blinking of this character<br>1 : Blinking of this character at the frequency of VSYNC/16, VSYNC/32, VSYNC/64 or VSYNC/128 rate |

MED200 - 2

Fig.22 Display RAM organization overview.

| T3 | T2 | |
|---|---|---|
| Size | | |
| 0 | 0 | 1 dot = 1H/1V |
| 0 | 1 | 1 dot = 2H/2V |
| 1 | 0 | 1 dot = 3H/3V |
| 1 | 1 | 1 dot = 4H/4V |

| T1 | T0 | |
|---|---|---|
| Line spacing between two rows | | |
| 0 | 0 | 0H line |
| 0 | 1 | 4H line |
| 1 | 0 | 8H line |
| 1 | 1 | 12H line |

MED201

Fig.23 Attribute for carriage return code character size of next row and line spacing.

| T3 | T2 | T1 | |
|---|---|---|---|
| R | G | B | colour |
| 0 | 0 | 0 | black |
| 0 | 0 | 1 | blue |
| 0 | 1 | 0 | green |
| 0 | 1 | 1 | cyan |
| 1 | 0 | 0 | red |
| 1 | 0 | 1 | magenta |
| 1 | 1 | 0 | yellow |
| 1 | 1 | 1 | white |

| T0 |
|---|
| End of display control |
| 0: continue display of next character<br>1: end of display |

MED203

Fig.24 Background colour and end of display control.

### 13.1 Bits in the Display RAM

There are three different kinds of bit < 9 : 4 > configurations to be considered:

1. When bit < 9 : 4 > indicates a **customized character**, bit < 3 : 1 > determines the colour (1 out of 8) of this character. Blinking of this character is controlled by bit < 0 >.

   Blinking duty cycle and frequency control (see Section 14.1.3).

   Figure 22 gives the colour table and blinking control.

2. **Carriage return code** (bit < 9 : 4 > = 111110) is a special code that will finish the current display line and start to display the character next to this code in the beginning of next line (row) with a spacing of number of horizontal scan lines between lines (rows) defined by bit < 1 : 0 >.

   It displays a transparent pattern on the screen.

   Character size of the next line is determined by bit < 3 : 2 >. 1 dot is composed of 1H/1V, 2H/2V, 3H/3V or 4H/4V (H: DOSC clock period, V: Horizontal scan line). Figure 23 gives the details.

3. **Space code** is a code which displays a transparent pattern on the screen and lasts for 1 character width. When the bit < 9 : 4 > = 111111, bit < 3 : 1 > determines the background colour of the characters/words from this space code in 'box shadowing' and/or 'north-west shadowing' modes (see Section 14.1.2 for detail). Bit < 0 > is the 'end of display' bit which is to indicate the end of the display of current screen before exhaustion of display RAM (i.e. before the 64th RAM location).

   Figure 24 gives the details and Fig.25 illustrates an example of the timing of FB, R, G, and B pulses when displaying a line of dots stream in a character (FB = VOB; R, G, B = VOW0, VOW1, VOW2).

Figure 26 shows an example of the screen which includes some CR and SPACE code in.

In summary:

1. **Carriage return** code is to control
   a) character size of next line
   b) space (in terms of number of horizontal scan lines) between current character line and next line.
2. **Space** code is to control
   a) the background colour of the characters from this space code
   b) end of character display.
3. Display of the characters always starts from RAM address 0. Space and carriage return code is to change the attribute and the position of the characters. The last character displayed on the TV screen is either the 64th RAM location or a Space code with end-of-display attribute set to One.

### 13.2 How to load the character data into the display RAM

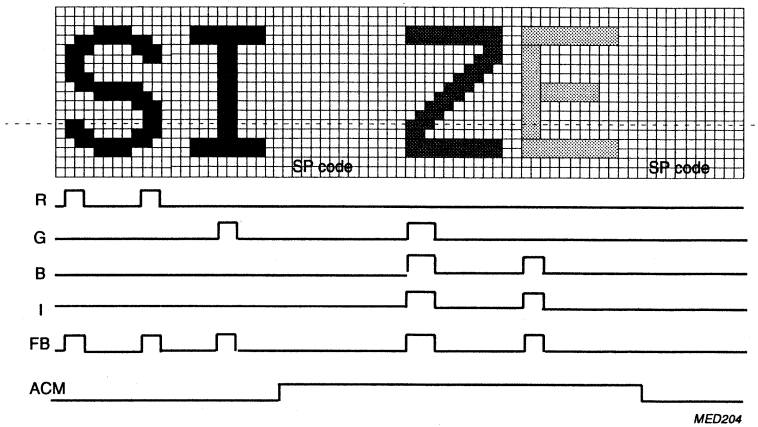
There are 3 registers to control/address/buffer the display characters.

**DCRAR(5 : 0), derivative register 30H:** is the address register which points to the location of the display RAM (1 out of 64 address) for which data is to be written into.

**DCRTR(3 : 0), derivative register 31H:** is the attribute register which store the attribute data accompanied with the data in derivative register 32H.

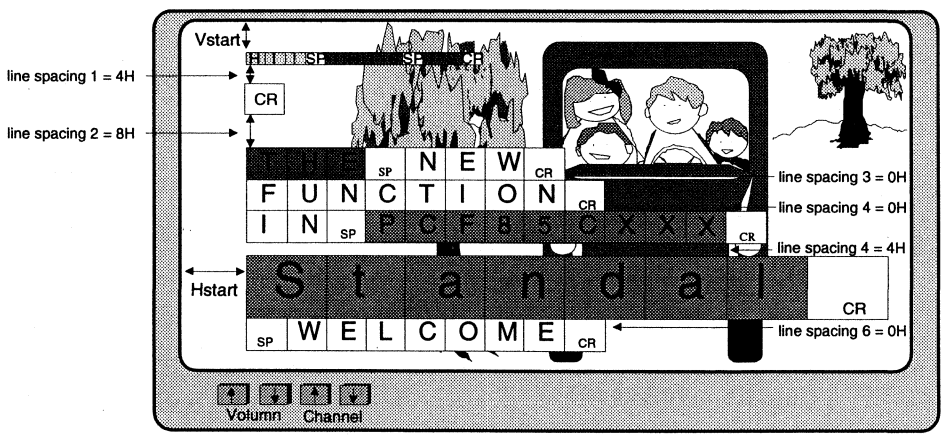
**DCRCR(5 : 0), derivative register 32H:** is the character data register to store the character which is going to be loaded into the RAM location pointed by DCRAR.

Figure 27 shows the configuration of these three registers.




"S" : red colour
"I" : green colour
"Z" : G+B+I colour
"E" : B+I colour
1st SP code : ACM = on
2nd SP code : ACM = off

Fig.25 R, G, B and FB timing.




Four different background colours : (in box shadowing mode)

Black
Red
Green
Blue

Fig.26 On-screen-display (an example).

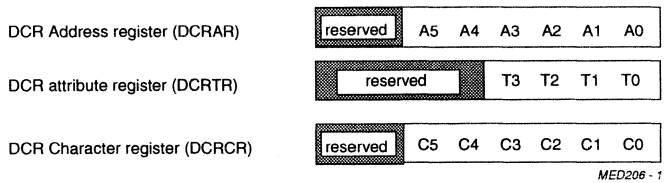



Fig.27 Control register for display characters.

### 13.3 The procedure to write character data into display RAM

1. Initialize the starting address of the display RAM by writing data into DCRAR(5 : 0).
2. Write DCRTR(3 : 0) to determine the attribute of the character. The meaning of these 4 bits is dependent on the contents of the next command, the data in the DCRCR(5 : 0) (i.e. carriage return code, space code or normal user's code).
3. Write to the Derivative register 32H, DCRCR(5 : 0), is to write the character code to be displayed on the screen. When the write to Derivative register 32H operation is finished, the data stored within the DCRTR and DCRCR are loaded into the RAM location pointed by the DCRAR.
4. Post increment operation is executed in the DCRAR (i.e. DCRAR ≤ DCRAR + 1) to make it point to the next RAM location.

Overflow of the DCRAR, i.e. overflow from 63 to 64, makes it reset to **zero**. It is step 3 which triggers the load of DCRCR and DCRTR into RAM and the post increment operation.

If the attributes of a series of displayed characters are the same, only DCRCR has to be updated.

Figures 28 and 29 show the post-increment operation.

Figure 29 shows how DCRAR is incremented and advanced.

Overflow of the post-increment just makes the registers reset to **zero**.

Initial value (after master $\overline{\text{RESET}}$) of DCRAR, DCRTR and DCRCR are all **zero**.

After the instruction MOV D32H, A

is finished, the post increment operation is performed automatically

Auto-post-increment operation:

```
Begin
  (DCRAR) ≤ (DCRAR) + 1
  If (DCRAR) > 63 then (DCRAR) ≤ 0
End
```

MED207

Fig.28 Auto post increment operation.

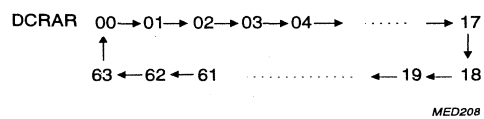
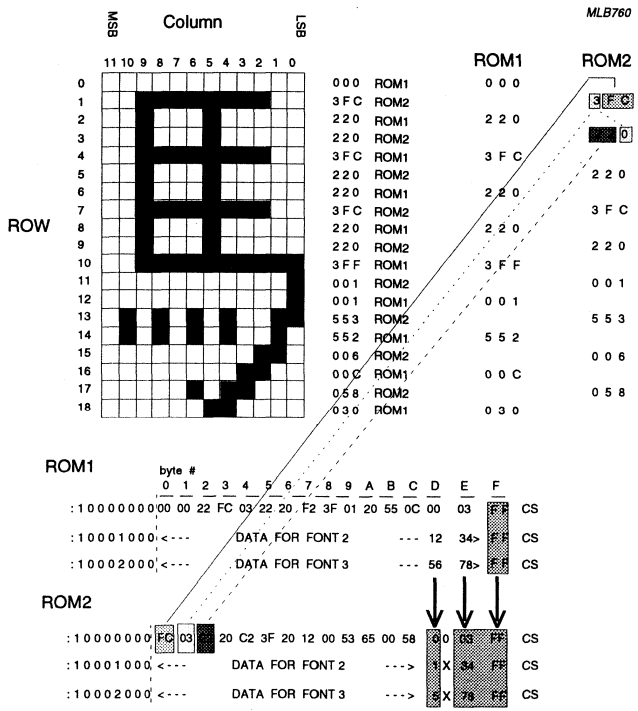



Fig.29 DCRAR increment cycle.




The ROW 0 of each font is reserved for vertical combination of two fonts. In vertical emergence, ROW 0 of the lower font has the same code as ROW 18 of the upper font.

Binary 1 stands for dots in PCA8510 and PCA84C846 (in PCA84C640 binary 0 stands for dots).

The ROM1 and ROM2 data files are in INTEL hex format with byte basis. Each byte is structured with higher nibble followed by lower nibble.

The unused last byte for each font in ROM1 must be filled with 'FF'.

The unused last 2 and half bytes in ROM2 must be filled with the same data with corresponding address locations in ROM1.

The DATA bytes of last two reserved fonts (RETURN and SPACE codes) should be filled with all '00'.

For PCA8510, the lower 16 characters which are reserved for soft RAM fonts in the character ROM address must be filled with all 'FF'.

X stands for the last data nibble in font 2/3 of ROM2; while CS represents for checksum.

Fig.30 Font pattern stored in character ROM1 and ROM2.

### 13.4 Default value of character size and background colour

The default value, after master $\overline{\text{RESET}}$, of the display characters are as follows:

- Background colour = blue (R = 0, G = 0, B = 1)
- Character size = 1V/1H
- End of display control = 0
- If another set-up is needed, the first character should be space code and second character is CR code to define the character size and background colour.

## 14 CHARACTER ROM

The character displayed on the screen is in 12 × 18 dot matrix format while it is in 12 × 19 format stored in the on-chip character ROM. The character ROM is actually divided into two part ROM1 and ROM2.

The bit pattern stored in ROM1 and ROM2 is shown in Fig.30.

ROM 1 to 18 is the bit pattern you see on the TV screen.

ROW 0 is for use only in the north-west shadowing mode (details see Section 14.1.2). It is when two character cells are combined in vertical direction to formulate a new pattern, this ROW 0 contains the same bit pattern of ROW 18 of the character above it.

If no combined character in vertical direction is intended for this character, ROW 0 should be filled with all zeros.

The file format to submit to Philips to make customized character sets is demonstrated in Fig.30.
Philips provides a software under MS DOS environment (IBM/PC or compatible) to help customer to design the character font on the screen and to generate the bit pattern HEX decimal file automatically. Contact your local Philips Sales Organization for details.

### 14.1 OSD control registers

The functions of the OSD circuitry are controlled by the derivative registers 22H, 33H and 34H.

#### 14.1.1 Derivative register 22H

The VOW1E, VOW0E bits in the D22 are to enable the VOW1/DP22 and VOW0/DP23 pins as VOWi (i = 1 or 0) output or not (= 0).

#### 14.1.2 Derivative register 34H

1. EN bit is to disable (= 0) or enable (= 1) the OSD function.
2. Bp bit is to set the output polarity of VOB, VOW0, VOW1 and VOW2 to active HIGH (= 1) or active LOW (= 0); default = 1 (see Fig.31).
3. Hp is for the Horizontal Sync input polarity (see Fig.19).
4. Hp = 0, active LOW.
5. Hp = 1, active HIGH; default = 0.
6. Vp is similar to Hp, it is for the Vertical Sync input (see Fig.19).
7. Vp = 0, active LOW.
8. Vp = 1, active HIGH; default = 0.
9. S1, S0 are for background/shadowing mode control.

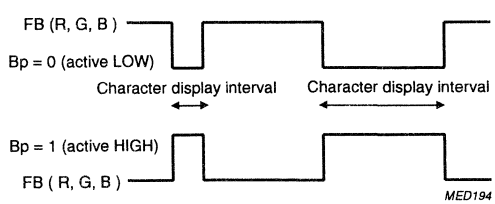



Fig.31 Bp bit to decide the active level of FB, R, G and B.

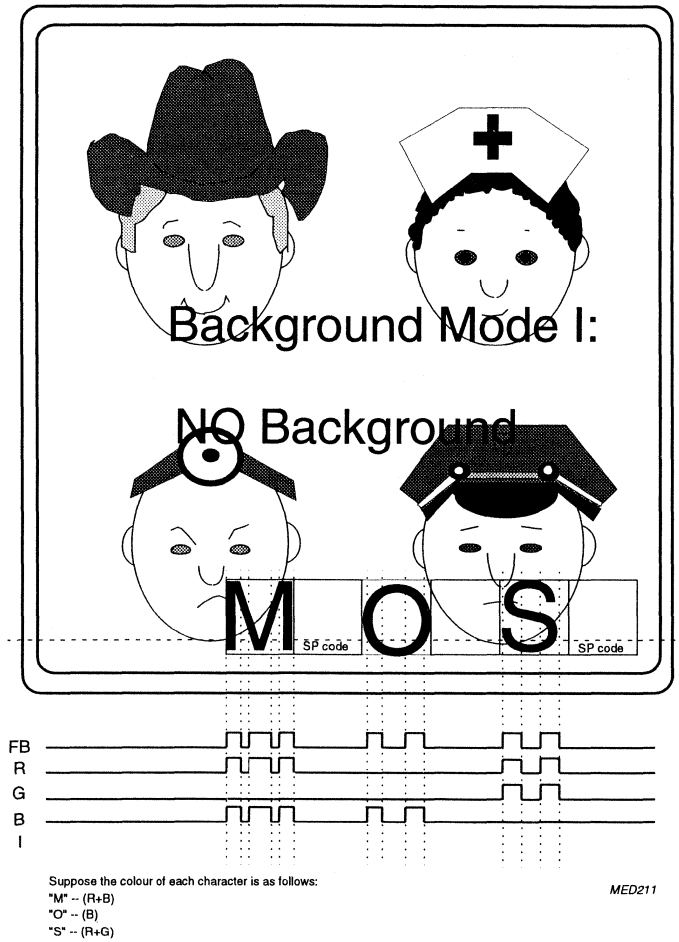
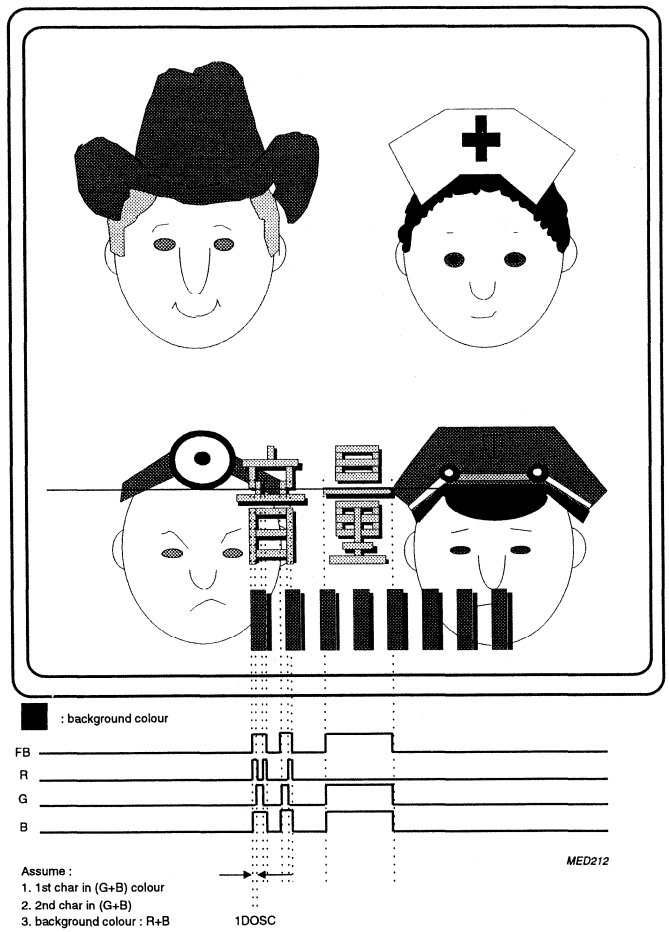



Fig.32 Background shadowing mode (I): NO Background - Superimpose.




Available only in character size 2V/2H or 4V/4H.

Fig.33 Background shadowing mode (II): North-west Shadowing.

Mode 0: (Fig.32) S1, S0 = 00, 'no background' mode.

The OSD fonts/characters are directly superimposed on the TV video signals.

Mode 1: (Fig.33) S1, S0 = 01, north-west shadowing, available only in the character size 2V/2H or 4V/4H (V: horizontal line; H: OSD clock).

The shadows of the characters are generated by placing a light source on the north-west 45 degree direction (see also Figs 36 and 37).

When designing the character bit pattern, care must be taken that the shadows generated by this mode is only within the cell boundary in vertical direction (see Figs 39 and 40 for details). But shadows generated by this mode in horizontal direction has no boundary limitation (Fig.41).

Mode 2: (Fig.34) S1, S0 = 10, box shadowing.

Box shadowing is to surround the character font by a 12 × 18 dots box in background. i.e. within the character font cell, where there is no foreground dot there is background dot (see Fig.38).

Mode 3: (Fig.35) S1, S0 = 11, Frame Shadowing (raster blanking), background colour displayed on full screen where no bit patterns are on.

The background colour is controlled by derivative register 37H and has 8 different colours.

#### 14.1.3 Derivative register 33H

Derivative register 33H is to control the character blinking related operation. BF1 and BF0 bits are to control the character blinking frequency.
It is determined by the equation:

$$\text{Blinking frequency} = \frac{f_{VSYNC}}{16 \times 2^{(BF1,\,BF0)}}\ \text{Hz},$$

where '$2^{(BF1,\,BF0)}$' is a decimal value determined by bits BF1 and BF0. For an overview of values see Table 5.

**Table 5** Blinking frequency determined by (BF1,BF0).

| BF1 | BF0 | $2^{(BF1,\,BF0)}$ | BLINKING FREQUENCY (Hz) |
|---|---|---|---|
| 0 | 0 | 1 | $1/16\ f_{VSYNC}$ |
| 0 | 1 | 2 | $1/32\ f_{VSYNC}$ |
| 1 | 0 | 4 | $1/64\ f_{VSYNC}$ |
| 1 | 1 | 8 | $1/128\ f_{VSYNC}$ (default) |

BR1, BR0 bits are to control the active ratio of the character blinking; see Table 6.

**Table 6** Active ratio determined by bits BR1 and BR0.

| BR1 | BR0 | ACTIVE RATIO |
|---|---|---|
| 0 | 0 | 3 : 1 (default) |
| 0 | 1 | 1 : 1 |
| 1 | 0 | 1 : 3 |
| 1 | 1 | reserved |

Figure 42 shows the timing diagram of character blinking frequency and blinking ratio.

#### 14.1.4 Derivative register 23H

Bit 7 and bit 6 are for $\overline{\text{VSYNC}}$ signal interrupt control purpose (bits 2 to 0 are for PWM control, see Chapters 8 and 9 for details).

The $\overline{\text{VSYNC}}$ leading edge (active level detection automatically done by PCA84C846) generates an interrupt if bit 6 of D23 is enabled (= 1) and the SIO interrupt is enabled (i.e. the I²C-bus and the $\overline{\text{VSYNC}}$ interrupt shares the same interrupt vector).

Bit 7 of D23 is to indicate if the interrupt comes from $\overline{\text{VSYNC}}$ (if = 1 and bit 6 enabled) or I²C-bus when the CPU gets interrupted by interrupt vector address 7.

### 14.2 Combination of two or more font cells to formulate a new font

The user can combine two (or more) font cells to formulate a new higher resolution pattern, (Figs 43, 44, 45 and 46).

Combination of two cells in horizontal direction needs no special care. All 4 background/shadowing modes are applicable.

But the combination of two cells in a vertical direction needs the following special care:

- Space between two rows should be programmed as 0 (bit <1 : 0> of carriage return code = 00).
- ROW 0 in the character ROM is for use in the 'north-west shadowing' mode. If this mode is intended for use by this formulated character font, the ROW 0 should contain the bit pattern of ROW 18 of the font above it (see Figs 45 and 46).

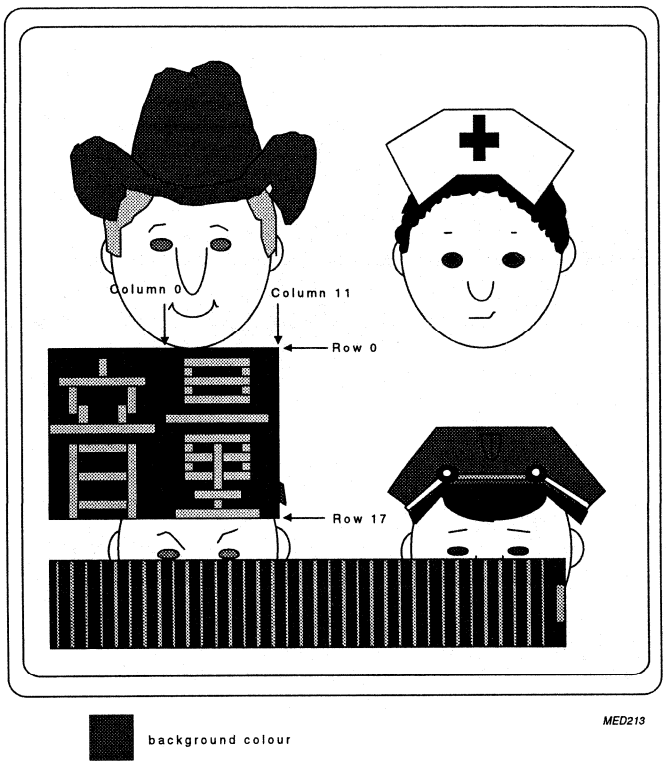



Fig.34 Background shadowing mode (III): Box Shadowing mode.

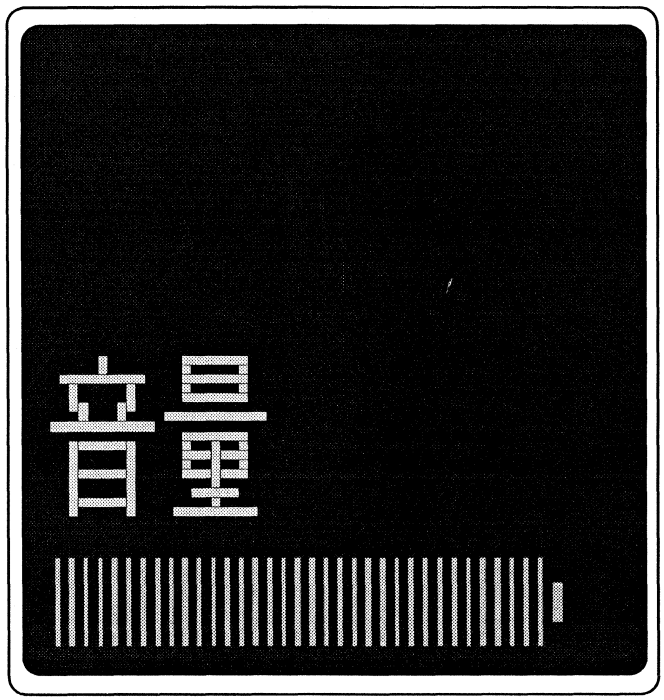



Fig.35 Background shadowing mode (IV): Frame Shadowing mode.

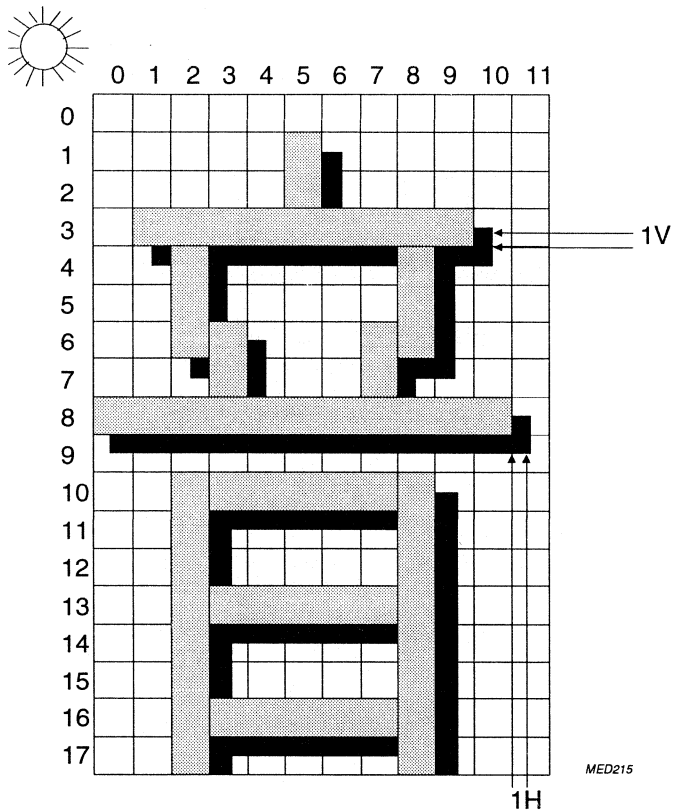



Fig.36 Example of North-west shadowing; mode size = 2V/2H.

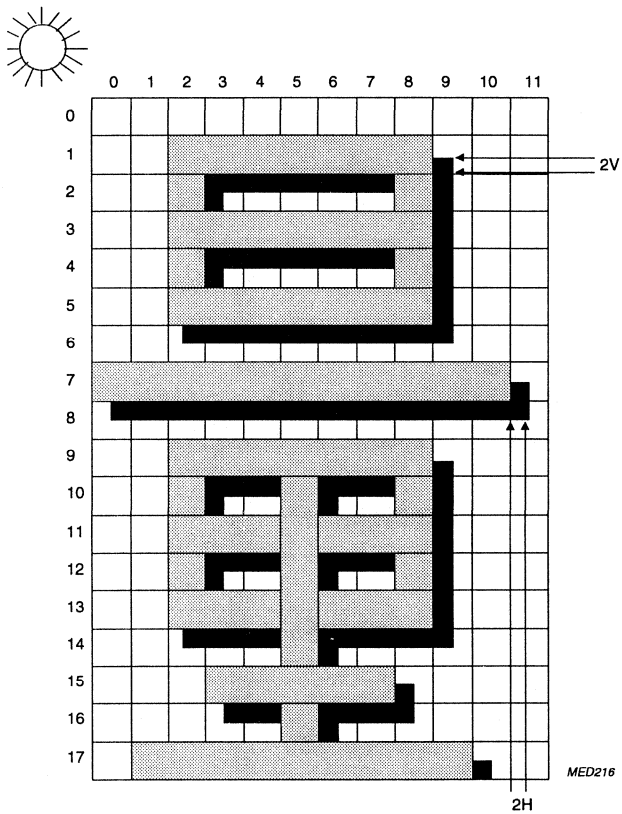



Fig.37 Example of North-west shadowing; mode size = 4V/4H.

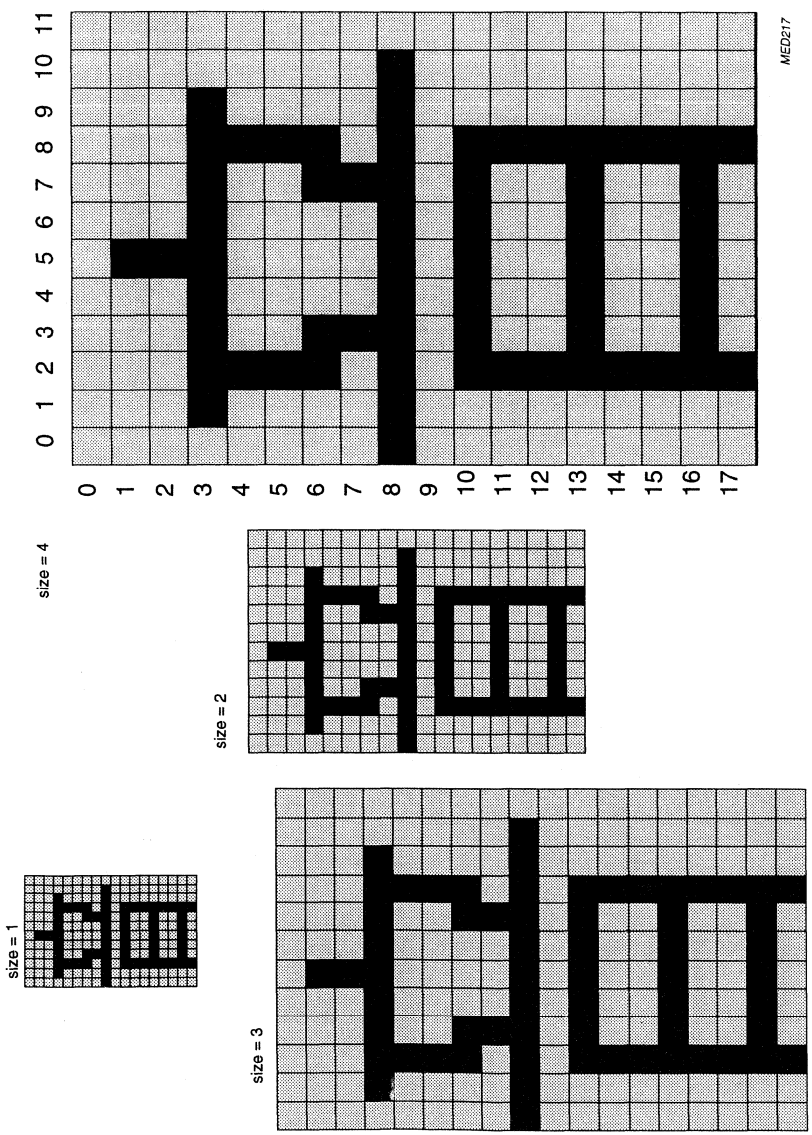



Fig.38 Example of box shadowing mode.

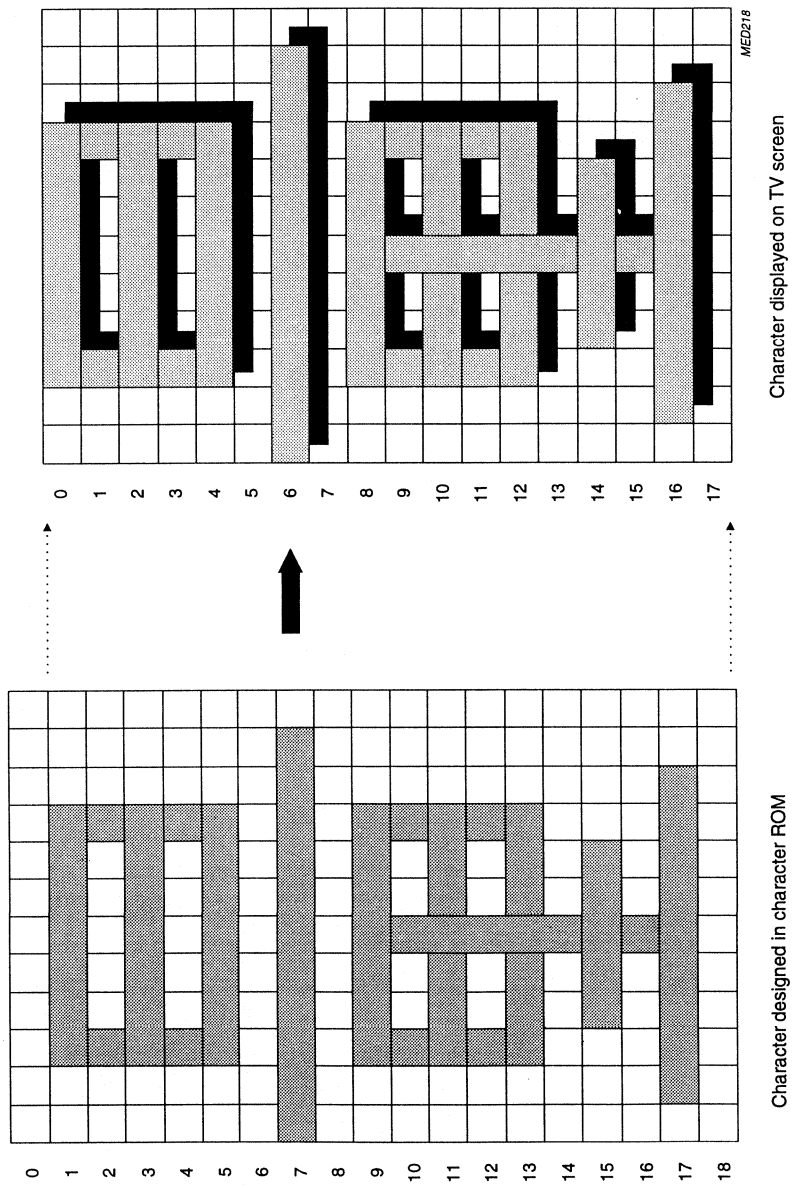



Fig.39 Example of North-west shadowing mode.

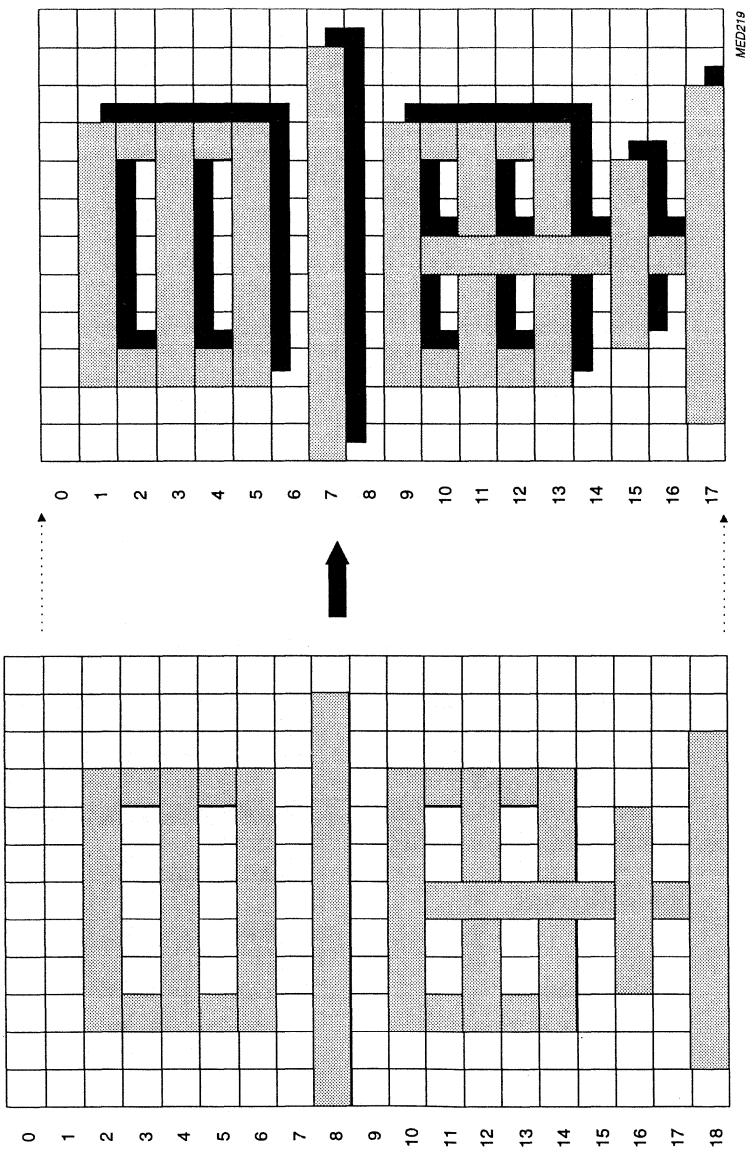



Fig.40 Example of North-west shadowing mode.

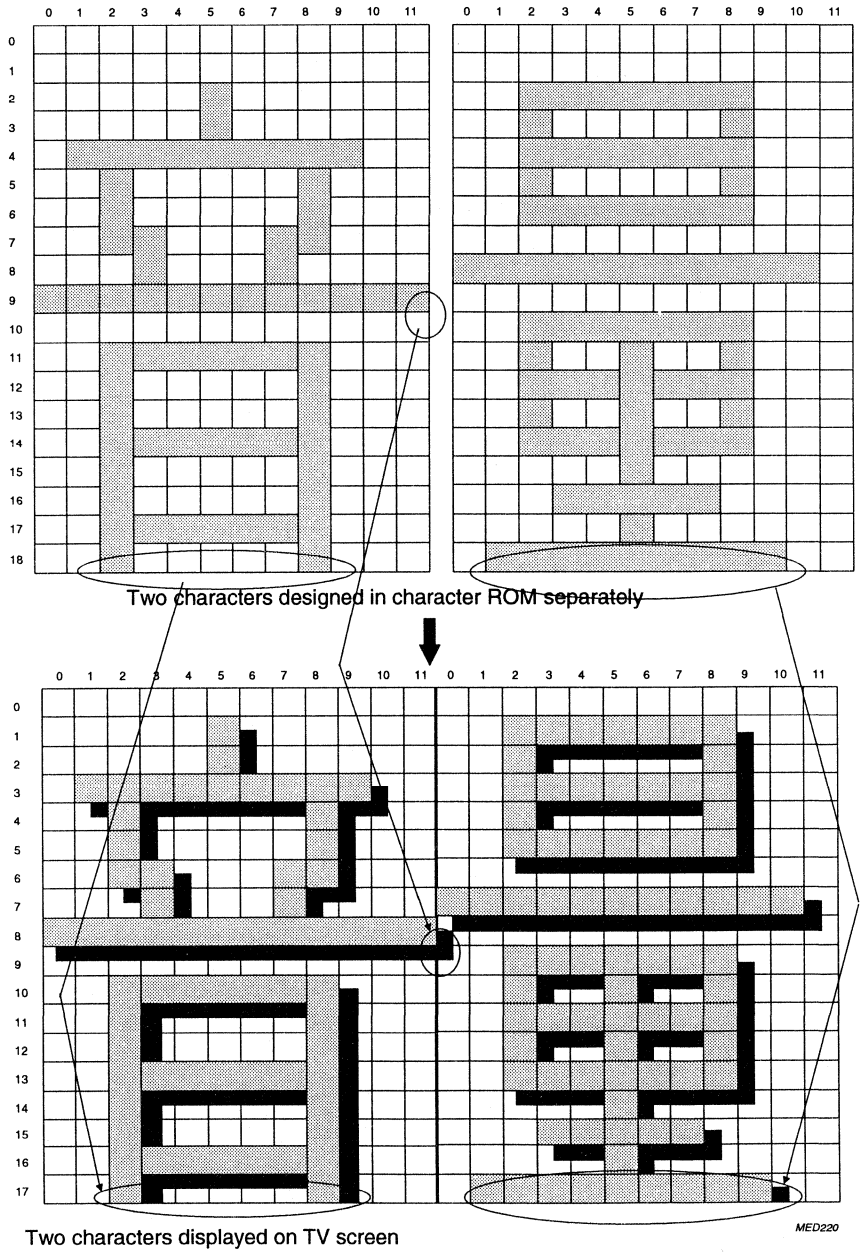



Fig.41 North-west shadowing.

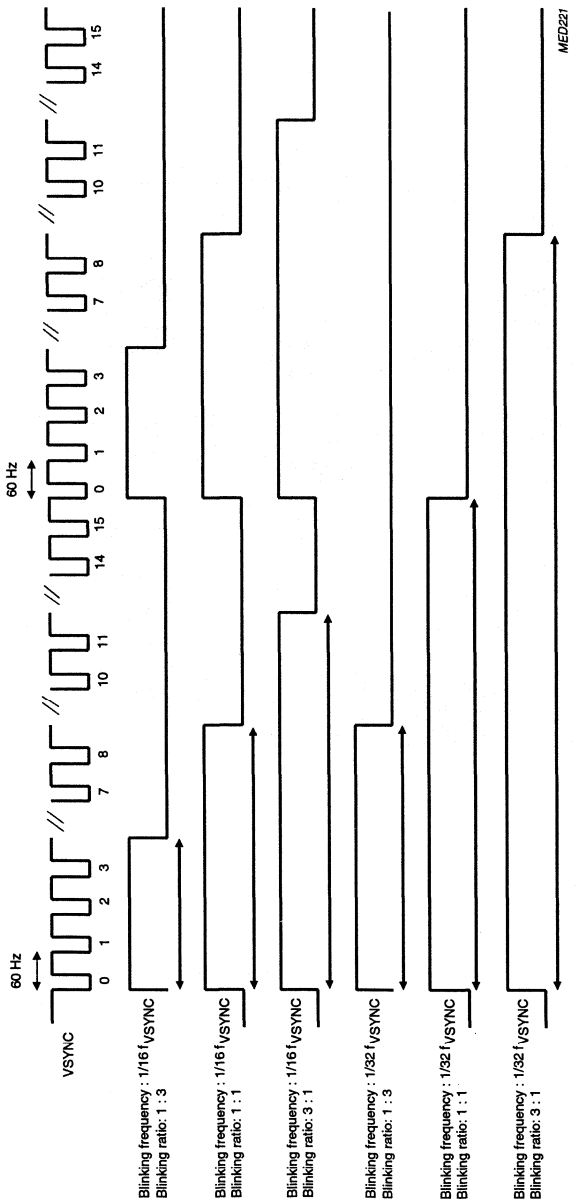



Fig.42 Example of character blinking (NTSC 525LPF/60Hz).

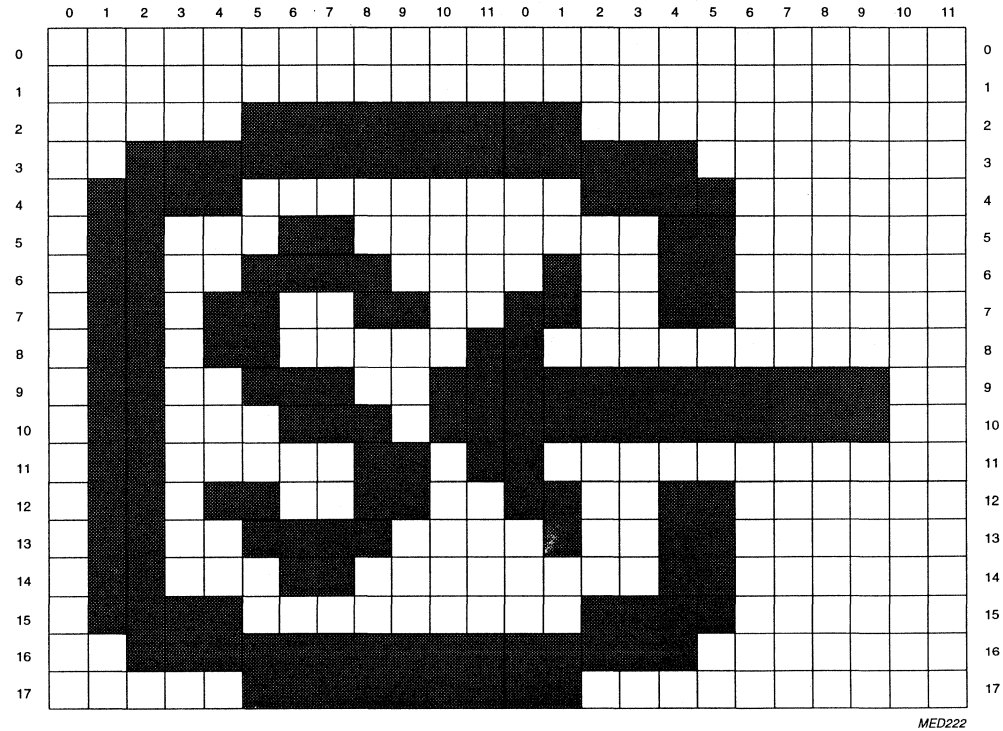



Fig.43 Combination of two character cells to form a new font (in horizontal direction).

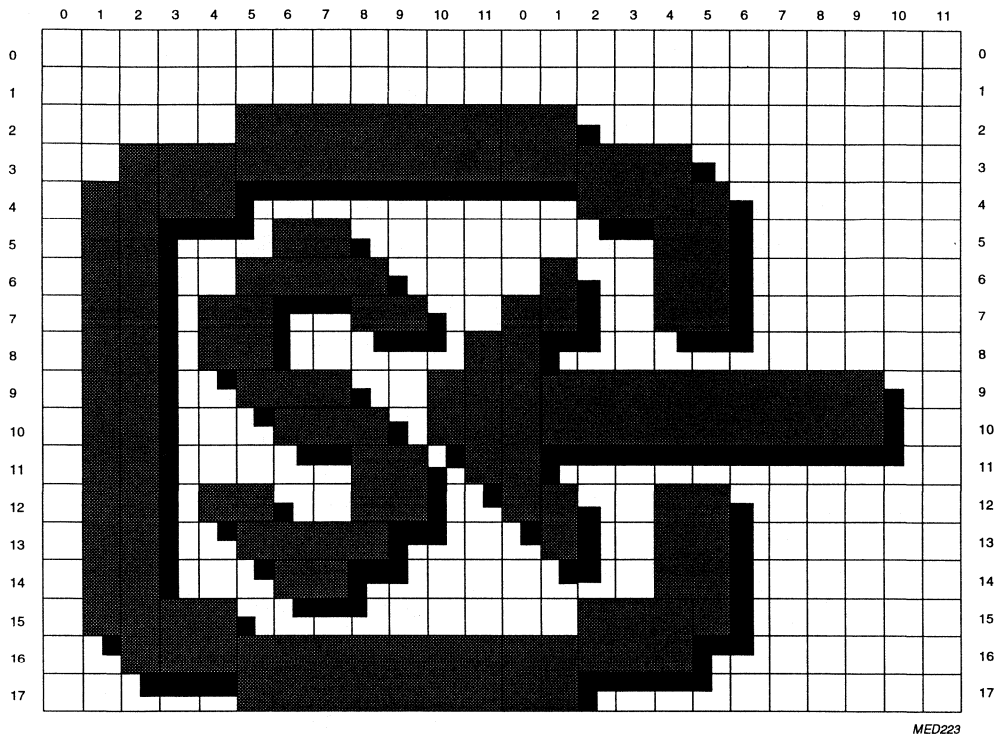



Fig.44 North-west shadowing mode in combination of two cells to form new font (in horizontal direction).

### 14.3 Space code and carriage return code in different background/shadowing modes

Figures 47, 48, 49 and 50 show the space code and carriage return code in 4 different background/shadowing modes (no blinking).

1. In mode 0, 'no background mode' - both reserved codes display a transparent (no bit) pattern with the video signal as its background (superimpose).
2. In mode 1, 'north-west shadowing' - same as mode 0.
3. In mode 2, 'box shadowing mode' - space code display a transparent pattern with background colour that it intends to change or keep (the same as the background colour of the character next to this space code). Carriage return code is a transparent pattern super imposed on the video.
4. In mode 3, 'frame shadowing' - transparent pattern with background colour as its colour.

Blinking of character in both space code and carriage return code are shown in Figs 51, 52, 53 and 54.

Figure 55 shows blinking of character is only within 12 × 18 boundary. If the shadow of the blinking character cross over the boundary of the cell of the character next to it which is not blinking, the shadow dot will still appear on the screen regardless if the blinking character is on or off.

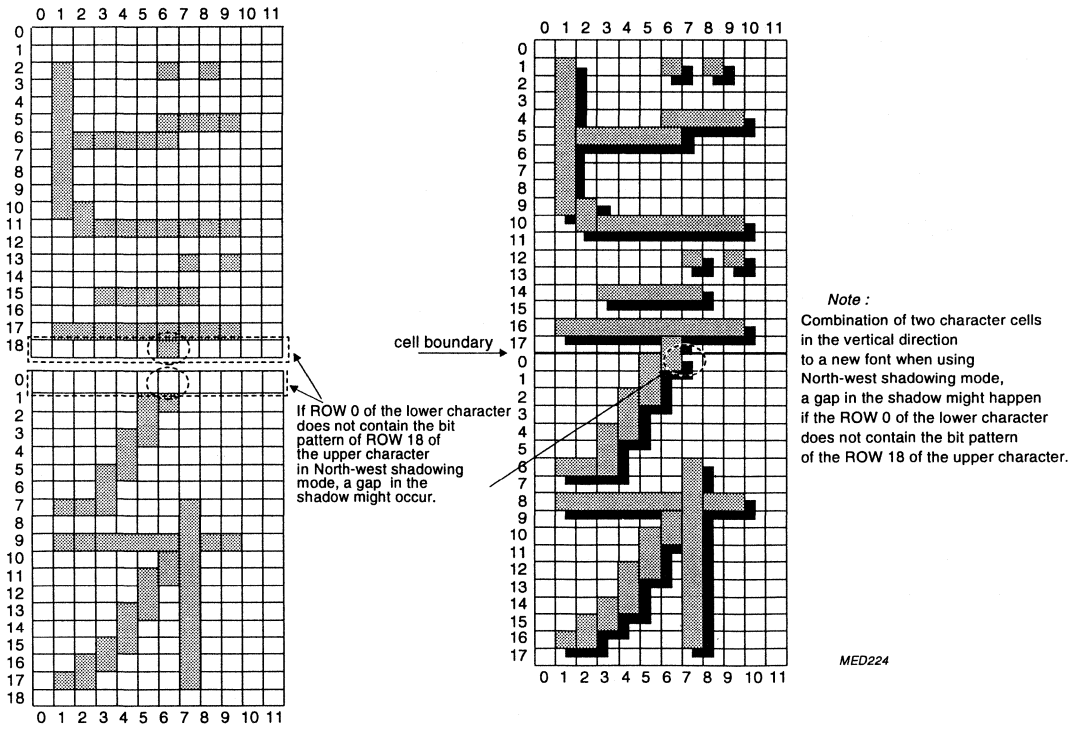



Fig.45 Combination of two characters in vertical direction to formulate a new pattern.

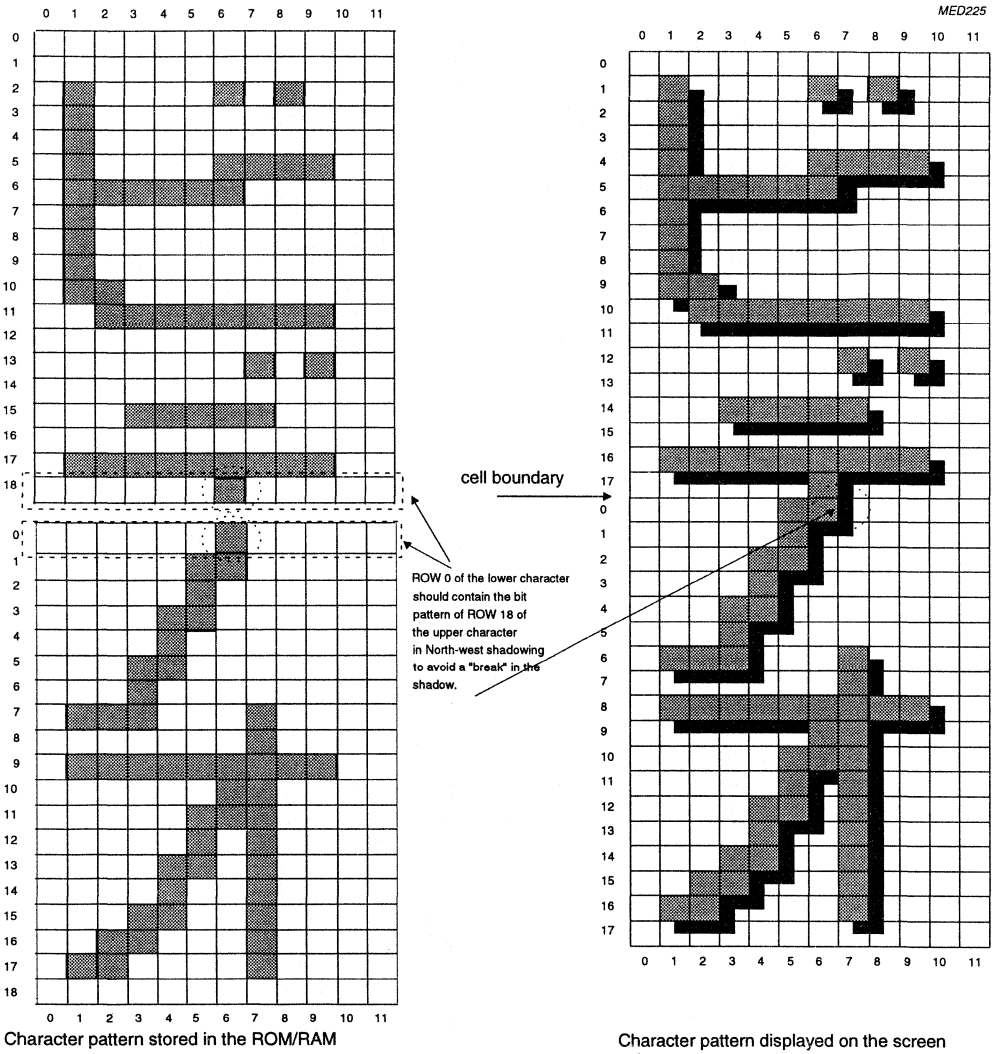



Fig.46 Combination of two characters in vertical direction to formulate a new pattern.

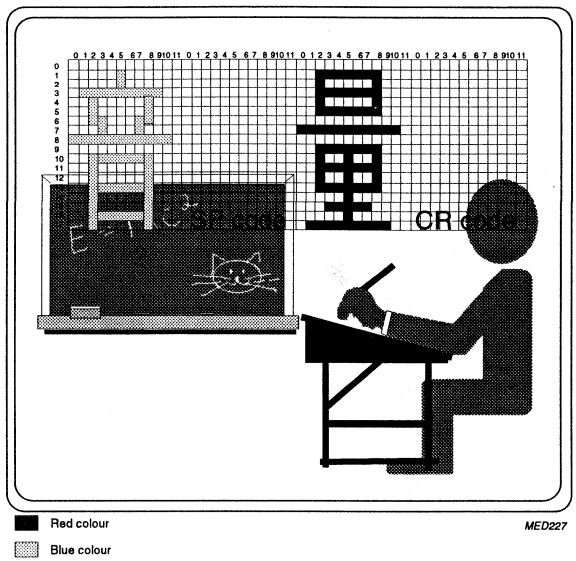



Fig.47 SP and CR codes in no background (superimpose) mode (transparent pattern).

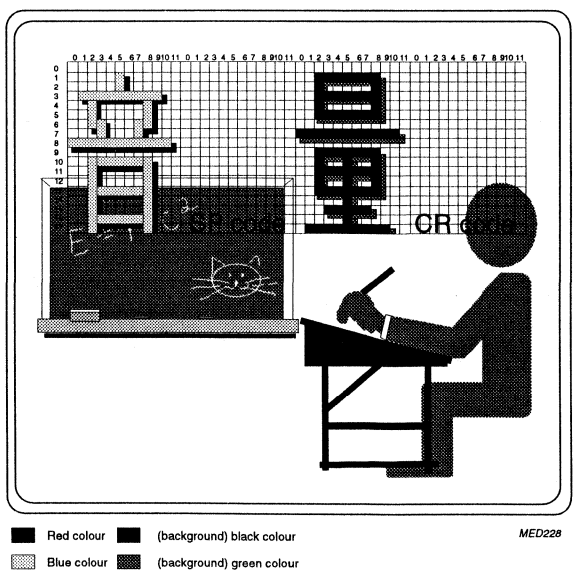
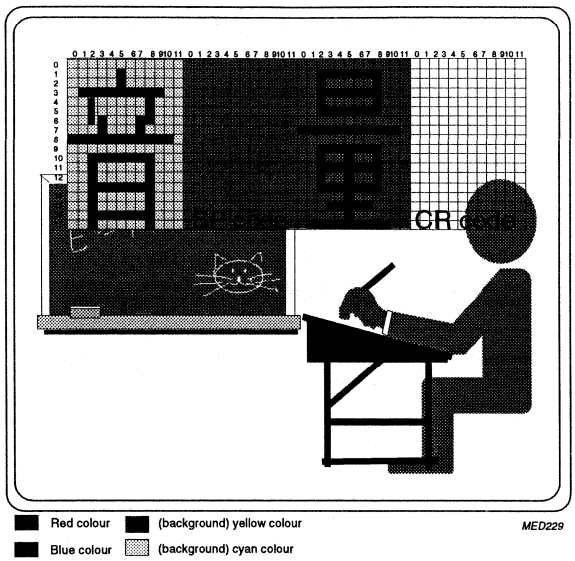
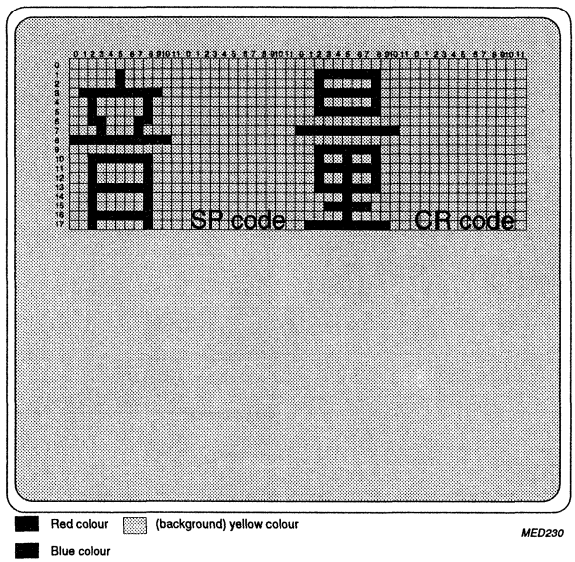



Fig.48 SP and CR codes in north-west shadowing mode (transparent pattern).




SP code is a transparent pattern with the background colour of the character it intends to change or keep.
CR code is always a transparent pattern with the video signal as its background.
SP code can change the background colour of itself and the character/word next to it (in this example: from cyan to yellow).

Fig.49 SP and CR codes in box shadowing mode.




SP and CR codes are all transparent pattern with the background colour as its colour.

Fig.50 SP and CR in frame shadowing mode.

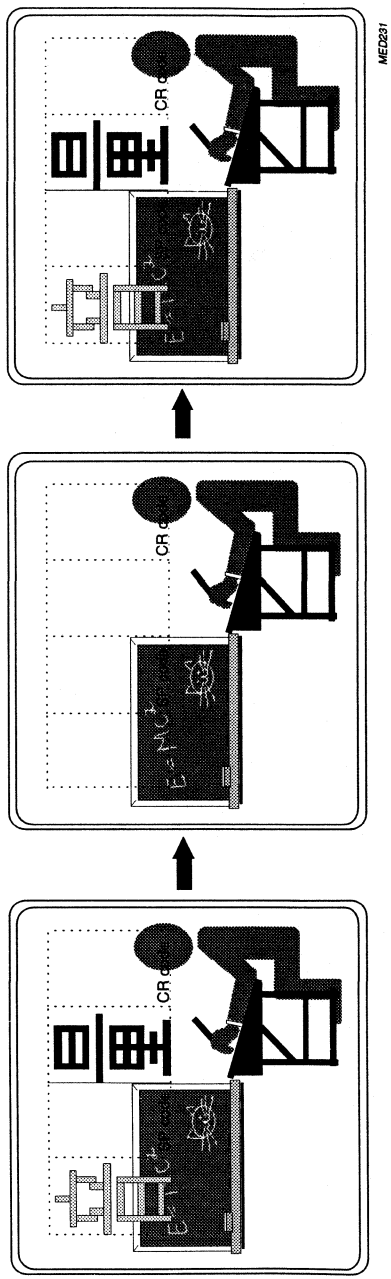



Fig.51 SP and CR in no background (superimpose) mode (transparent pattern) when blinking of character is set.

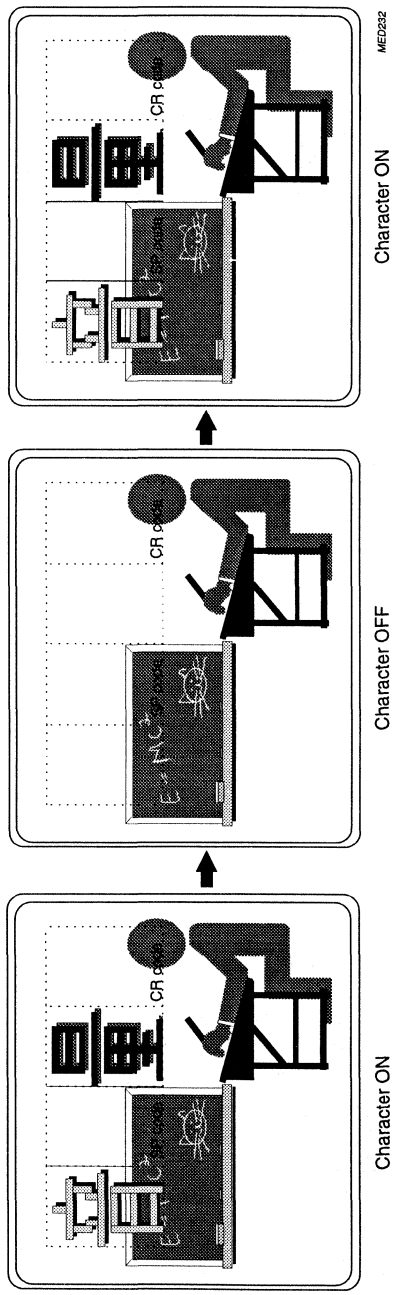



Fig.52 SP and CR codes in north-west shadowing mode (transparent pattern) when blinking of character is set to active.

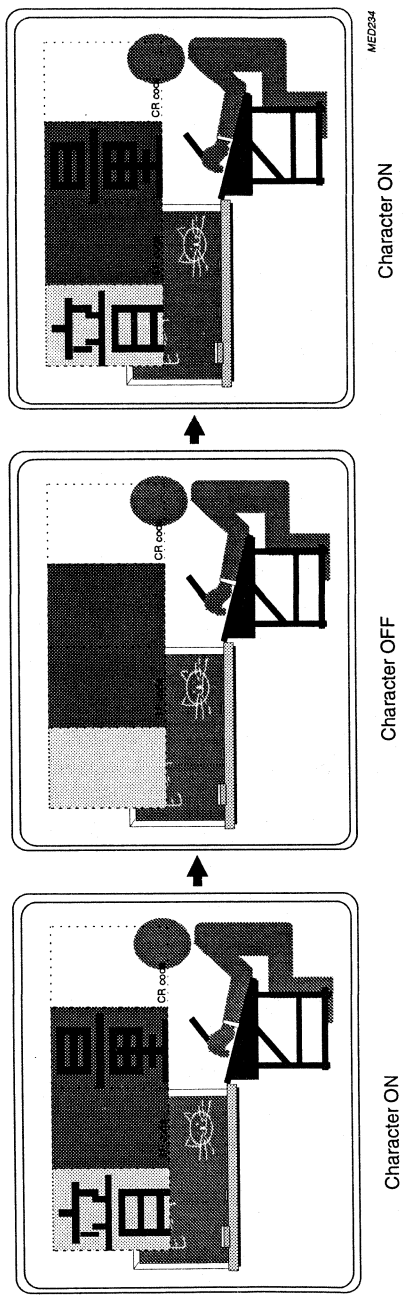



Fig.53 SP and CR codes in box shadowing mode with blinking of character set to active.

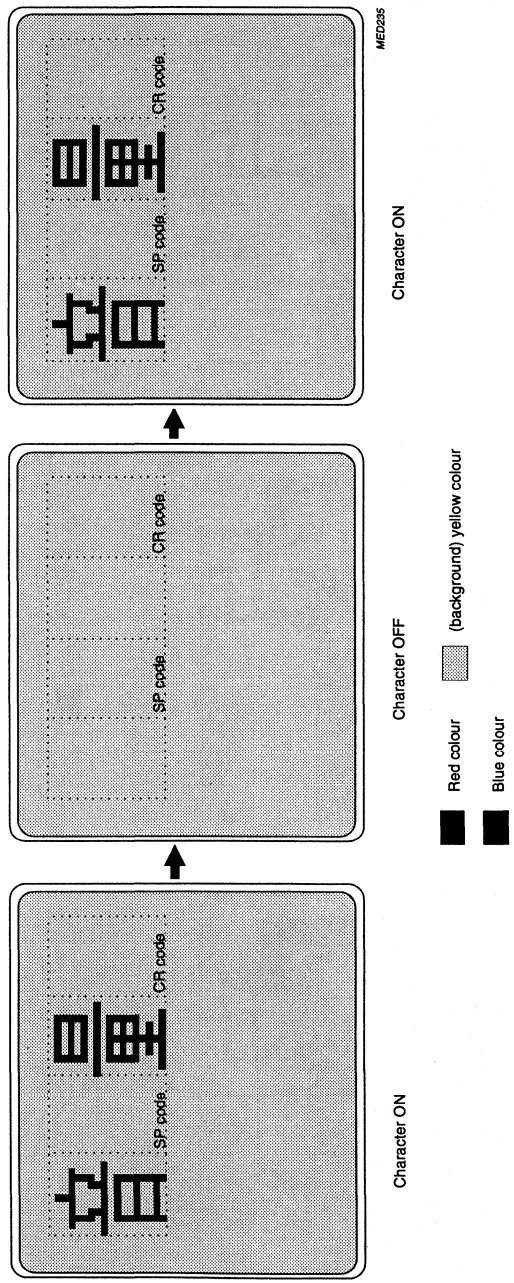
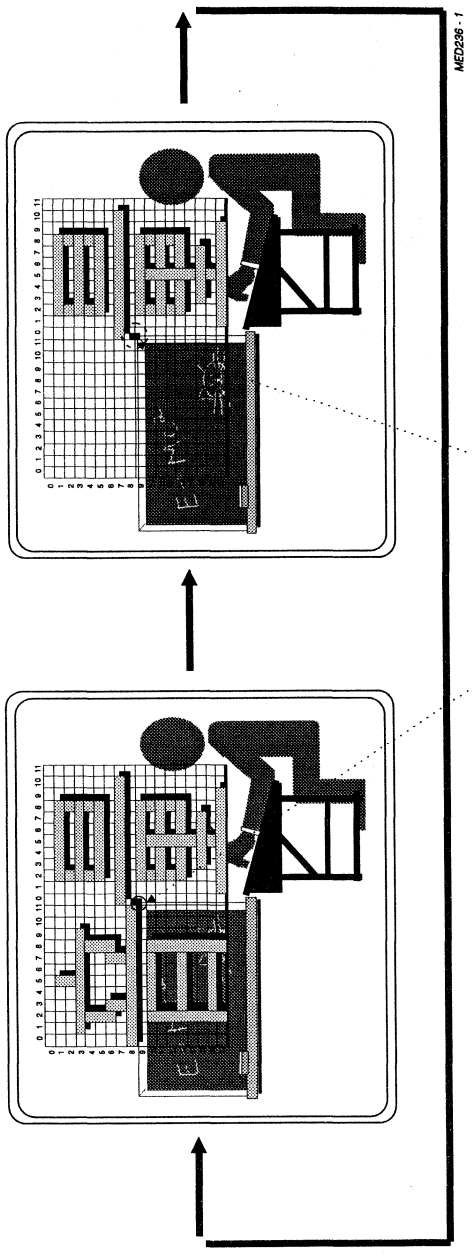



Fig.54 SP and CR codes in frame shadowing mode with blinking of character set to active.




In this example, assume
- 1st character is in blinking mode
- 2nd character is not
- the first character bit pattern is designed such that the north-west shadow of the 1st character falls into the cell of 2nd character

When the 1st character is blinking and shown up in the screen, everything is fine

But when the 1st character is off then the shadow of the 1st character falls into 2nd character cell will still remain there.

To avoid this happening, user can either
- design bit pattern in such a way that shadow does not cross the cell boundary
- make adjacent characters all blinking or all not blinking

Fig.55 Blinking of character is within the character cell only (12 × 18).

### 14.4 OSD clock in different TV standard

#### 14.4.1 Maximum number of characters/row and row/frame in different TV standard

The maximum number of characters per row is determined by the:

- DOSC (i.e. depending on $\overline{\text{HSYNC}}$ frequency and value of the 7-bit programmable counter in the on-chip oscillator)
- The TV standard.

The maximum OSD clock frequency is 12 MHz. Given an NTSC 525LPF (Line Per Frame)/60 Hz example (Fig.56).

The active video signal period of a horizontal line is 53.5 µs. In order to reduce the jittering of the screen edge, overscan is normally applied by the TV manufacturer. It causes the real visible video period of a horizontal line reduced to 53.5 µs × $^{9}/_{10}$ = 48.15 µs.

#### 14.4.2 Case 1

- OSD clock frequency = 6 MHz i.e. period = 0.166 µs
- Number of visible dots on a horizontal line $^{48.15\ \mu s}/_{0.166\ \mu s}$ = 290 dots
- The starting of first character dot is roughly 45 dots after $\overline{\text{HSYNC}}$ (see command B, C, D)
- Therefore the visible dots are 290 − 45 = 245 dots
- Each character is composed of 12 × 18 dots
- So the maximum number of characters displayed on a row with OSD clock = 6 MHz in 525LPF/60 Hz NTSC are $^{245}/_{12}$ = 20 characters
- In a 19" TV screen, the width of a horizontal line is around 370 mm which gives approximately 18.5 mm/character in width.

#### 14.4.3 Case 2

- OSD clock frequency = 10 MHz, i.e. dot period = 0.1 µs
- $^{48.15}/_{0.1}$ = 481 dots per horizontal line
- 481 − 45 = 436 visible dots in a line
- $^{436}/_{12}$ = 36 characters per row
- Again, in the 19" TV standard, the character width is $^{370\ mm}/_{36}$ = 10.3 mm per character.

#### 14.4.4 Maximum number of rows of characters per frame

- In NTSC 525/60, the active lines are roughly 241.5H to 249.5H per field (Fig.57). Take 241 as an example, the maximum number of character rows per frame is (remember a character is 12 × 18 dots): $^{241}/_{18}$ = 13 rows.
- In PAL 625/50, it is approximately $^{280}/_{18}$ = 15 rows.

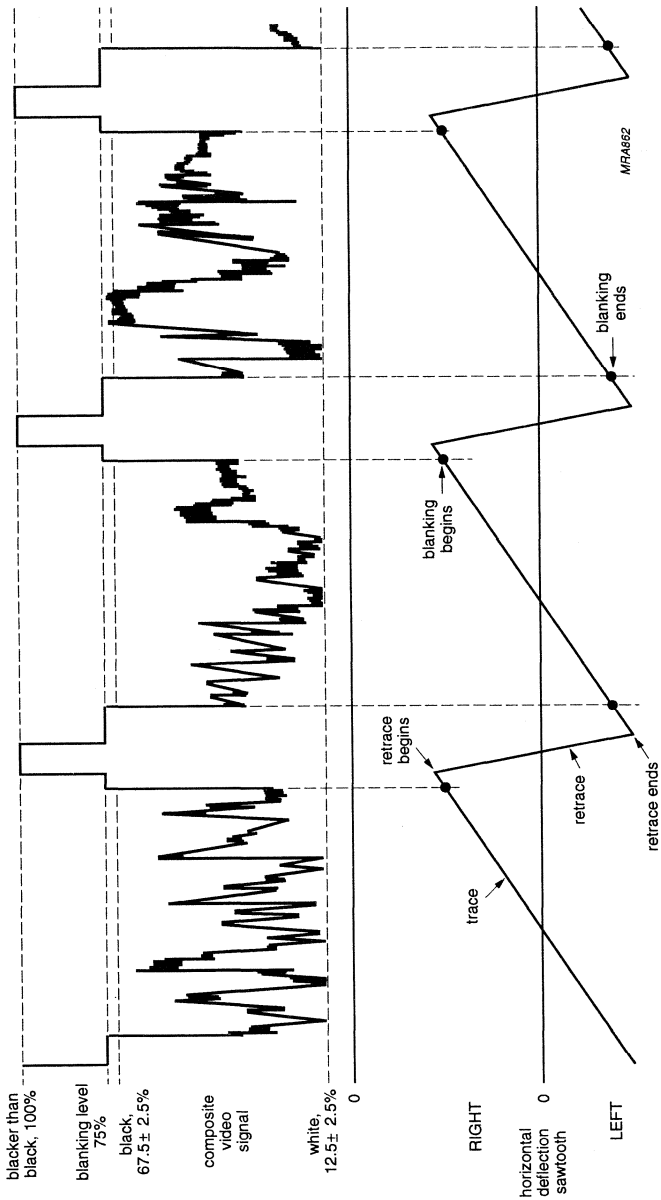



Fig.56 Composite video signal for three horizontal lines compared to three horizontal deflection sawteeth (NTSC 525LPF/60 Hz).

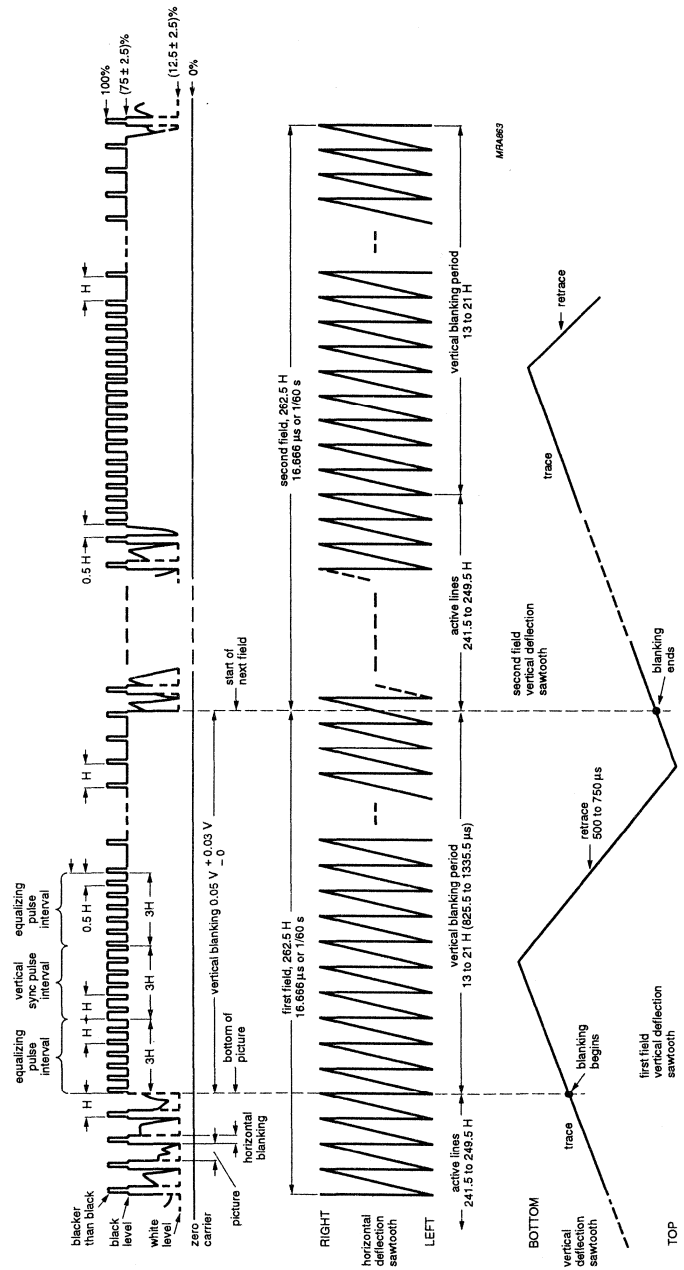



Fig.57 Vertical sync and blanking pulse intervals for one frame (NTSC 525LPF/60 Hz).

| Addr (HEX) | bit 7 | bit 6 | bit 5 | bit 4 | bit 3 | bit 2 | bit 1 | bit 0 | Register's name |
|---|---|---|---|---|---|---|---|---|---|
| 00 | DP07 -R | DP06 -R | DP05 -R | DP04 -R | DP03 -R | DP02 -R | DP01 -R | DP00 -R | DP0R (terminal) |
| 01 | -- | -- | -- | -- | DP13 -R | DP12 -R | DP11 -R | DP10 -R | DP1R (terminal) |
| 02 | -- | -- | -- | -- | DP23 -R | DP22 -R | DP21 -R | DP20 -R | DP2R (terminal) |
| 03 | DP07 1RW | DP06 1RW | DP05 1RW | DP04 1RW | DP03 1RW | DP02 1RW | DP01 1RW | DP00 1RW | DP0R (latch) |
| 04 | -- | -- | -- | -- | DP13 1RW | DP12 1RW | DP11 1RW | DP10 1RW | DP1R (latch) |
| 05 | -- | -- | -- | -- | DP23 1RW | DP22 1RW | DP21 1RW | DP20 1RW | DP2R (latch) |
| | | | | | | | | | |
| 10 | -- | PWM06 0RW | PWM05 0RW | PWM04 0RW | PWM03 0RW | PWM02 0RW | PWM01 0RW | PWM00 0RW | PWM0 |
| 11 | -- | PWM16 0RW | PWM15 0RW | PWM14 0RW | PWM13 0RW | PWM12 0RW | PWM11 0RW | PWM10 0RW | PWM1 |
| 12 | -- | PWM26 0RW | PWM25 0RW | PWM24 0RW | PWM23 0RW | PWM22 0RW | PWM21 0RW | PWM20 0RW | PWM2 |
| 13 | -- | PWM36 0RW | PWM35 0RW | PWM34 0RW | PWM33 0RW | PWM32 0RW | PWM31 0RW | PWM30 0RW | PWM3 |
| 14 | -- | -- | PWM45 0RW | PWM44 0RW | PWM43 0RW | PWM42 0RW | PWM41 0RW | PWM40 0RW | PWM4 |
| 15 | -- | -- | PWM55 0RW | PWM54 0RW | PWM53 0RW | PWM52 0RW | PWM51 0RW | PWM50 0RW | PWM5 |
| 16 | -- | -- | PWM65 0RW | PWM64 0RW | PWM63 0RW | PWM62 0RW | PWM61 0RW | PWM60 0RW | PWM6 |
| 17 | -- | -- | PWM75 0RW | PWM74 0RW | PWM73 0RW | PWM72 0RW | PWM71 0RW | PWM70 0RW | PWM7 |
| 18 | -- | VST06 0RW | VST05 0RW | VST04 0RW | VST03 0RW | VST02 0RW | VST01 0RW | VST00 0RW | VSTL |
| 19 | -- | VST13 0RW | VST12 0RW | VST11 0RW | VST10 0RW | VST09 0RW | VST08 0RW | VST07 0RW | VSTH |
| 20 | -- | AFCH1 0RW | AFCH0 0RW | AFC3 0RW | AFC2 0RW | AFC1 0RW | AFC0 0RW | AFCC -R | AFCCN |
| 21 | PWM7E 0RW | PWM6E 0RW | PWM5E 0RW | PWM4E 0RW | PWM3E 0RW | PWM2E 0RW | PWM1E 0RW | PWM0E 0RW | PWME |
| 22 | TDACE 0RW | SCLE 0RW | SDAE 0RW | AFCE2 0RW | AFCE1 0RW | AFCE0 0RW | VOW1E 0RW | VOW0E 0RW | CON1 |
| 23 | VINT 0R | VIEN 0RW | -- | -- | -- | P14LV 0RW | P7LVL 0RW | P6LVL 0RW | CON2 |
| 24 | T3b7 0R | T3b6 0R | T3b5 0R | T3b4 0R | T3b3 0R | T3b2 0R | T3b1 0R | T3b0 0R | T3CON |
| 25 | -- | PLL6 0RW | PLL5 0RW | PLL4 0RW | PLL3 0RW | PLL2 0RW | PLL1 0RW | PLL0 0RW | PLLCN |
| 30 | -- | -- | DCRA5 0RW | DCRA4 0RW | DCRA3 0RW | DCRA2 0RW | DCRA1 0RW | DCRA0 0RW | DCRAR |
| 31 | -- | -- | -- | -- | DCRT3 1W | DCRT2 1W | DCRT1 1W | DCRT0 1W | DCRTR |
| 32 | -- | -- | DCRC5 1W | DCRC4 1W | DCRC3 1W | DCRC2 1W | DCRC1 1W | DCRC0 1W | DCRCR |
| 33 | -- | -- | -- | -- | BR1 0RW | BR0 0RW | BF1 1RW | BF0 1RW | CON3 |
| 34 | -- | -- | S1 0RW | S0 0RW | Hp 0RW | Vp 0RW | Bp 1RW | EN 0RW | CON4 |
| 35 | -- | -- | V5 1W | V4 1W | V3 1W | V2 1W | V1 1W | V0 1W | VPOS (Vertical Position) |
| 36 | -- | -- | H5 0W | H4 0W | H3 0W | H2 0W | H1 0W | H0 0W | HPOS (Horizontal Position) |
| 37 | -- | -- | -- | -- | -- | BCR 0W | BCG 0W | BCB 1W | BCC Background color control |

*MED244*

A
BC

A: Bit name ('--' = not used)
B: Initial value ('0', '1' or '-' if not guaranteed)
C: Read, Write or Read Write

Fig.58 PCA84C846 register map.

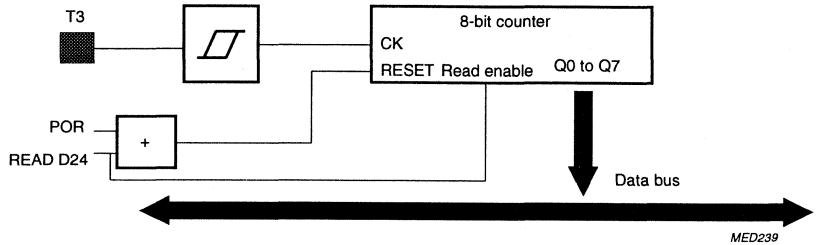



Fig.59 T3: 8-bit counter block diagram.

### 14.5 I²C-bus master slave transceiver

The I²C-bus master and slave transceiver is integrated. Control register 22H, SCLE and SDAE bit enable (= 1) the pins DP20/SDA and DP21/SCL as I²C-bus data and clock lines. If these two bits were disabled (= 0), these two pins become normal derivative port lines. Only port option 2 is available for these two pins.

### 14.6 T3: 8-bit counter

Figure 56 shows the block diagram of the 8-bit counter. A Schmitt-trigger input pin shapes the slow slope of the input signal into a square wave. The rising edge of the signal increases the (ripple) counter by 1.

The data in the counter can be read by instruction 'MOV A, D24H' (derivative register 24H). As soon as data is read, this counter is reset to **zero**. Overflow or power on $\overline{\text{RESET}}$ both reset the counter value to **zero**.

Minimum distance between two successive pulses (rising edges) is 30 µs.

## 15 OPTION LISTS

### 15.1 Port option

For the port options (1, 2 and 3) see Figs 15, 16 and 17.

**Table 7** Port options for making piggyback.
Only the port pins whose options are left blank, e.g. DP00, can be user mask programmable.

| PORT | PIN | OPTION | | PORT/DPORT | PIN | OPTION | | DPORT | PIN | OPTION | | DPORT | PIN | OPTION | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| P00 | 13 | 1 | S[1] | P10 | 7 | 1 | S | DP00 | 29 | | | DP20 | 40 | 2 | S |
| P01 | 14 | 1 | S | P11 | 8 | 1 | S | DP01 | 28 | | | DP21 | 39 | 2 | S |
| P02 | 15 | 1 | S | P12 | 10 | 1 | S | DP02 | 27 | | | DP22 | 3 | | |
| P03 | 16 | 1 | S | P14 | 12 | 1 | S | DP03 | 26 | | | DP23 | 4 | | |
| P04 | 17 | 1 | S | DP10 | 38 | | | DP04 | 25 | | | | | | |
| P05 | 18 | 1 | S | DP11 | 37 | | | DP05 | 24 | | | | | | |
| P06 | 19 | 1 | S | DP12 | 36 | | | DP06 | 23 | | | VOB[2] | 1 | | |
| P07 | 20 | 1 | S | DP13 | 9 | | | DP07 | 22 | | | VOW2[2] | 2 | | |

**Notes**

1. S = SET (and R = RESET), initial H or L after power-on reset.
2. Option 2 or 3 only (i.e. output only).

**Table 8** Port options for production.
Only the port pins whose options are left blank, e.g. DP00, can be user mask programmable.

| PORT | PIN | OPTION | | PORT/DPORT | PIN | OPTION | | DPORT | PIN | OPTION | | DPORT | PIN | OPTION | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| P00 | 13 | | | P10 | 7 | | | DP00 | 29 | | | DP20 | 40 | 2 | S[1] |
| P01 | 14 | | | P11 | 8 | | | DP01 | 28 | | | DP21 | 39 | 2 | S |
| P02 | 15 | | | P12 | 10 | | | DP02 | 27 | | | DP22 | 3 | | |
| P03 | 16 | | | P14 | 12 | | | DP03 | 26 | | | DP23 | 4 | | |
| P04 | 17 | | | DP10 | 38 | | | DP04 | 25 | | | | | | |
| P05 | 18 | | | DP11 | 37 | | | DP05 | 24 | | | | | | |
| P06 | 19 | | | DP12 | 36 | | | DP06 | 23 | | | VOB[2] | 1 | | |
| P07 | 20 | | | DP13 | 9 | | | DP07 | 22 | | | VOW2[2] | 2 | | |

**Notes**

1. S = SET (and R = RESET), initial H or L after power-on reset.
2. Option 2 or 3 only (i.e. output only).

### 15.2 On-chip oscillator transconductance

| OPTION | typ. $g_m$ at 5 V | $f_{osc}$ FOR QUARTZ | $f_{osc}$ FOR PXE |
|---|---|---|---|
| LOW ($g_{mL}$) | 0.4 mS | 1 to 6 MHz | not allowed |
| MEDIUM ($g_{mM}$) | 1.6 mS | 4 to 10 MHz | 1 to 6 MHz |
| HIGH ($g_{mH}$) | 4.5 mS | not allowed | 3 to 10 MHz |

## 16 LIMITING VALUES

In accordance with the Absolute Maximum Rating System (IEC 134).

| SYMBOL | PARAMETER | MIN. | MAX. | UNIT |
|---|---|---|---|---|
| $V_{DD}$ | supply voltage | −0.3 | +7 | V |
| $V_I$ | all input voltages | −0.3 | $V_{DD}$ + 0.3 | V |
| $P_{tot}$ | total power dissipation | – | 1 | W |
| $I_{OH}$ | maximum source current for all port lines | – | −10 | mA |
| $I_{OL}$ | maximum sink current for all port lines | – | 30 | mA |
| $T_{stg}$ | storage temperature | −55 | +125 | °C |
| $T_{amb}$ | ambient operating temperature | −20 | +70 | °C |

## 17 DC CHARACTERISTICS

$V_{DD}$ = 5 V; $V_{SS}$ = 0 V; $T_{amb}$ = −20 to +70 °C; all voltages with respect to $V_{SS}$; unless otherwise specified.

| SYMBOL | PARAMETER | CONDITIONS | MIN. | TYP. | MAX. | UNIT |
|---|---|---|---|---|---|---|
| $V_{DD}$ | operating supply voltage | | 4.5 | 5.0 | 5.5 | V |
| $I_{DD}$ | operating supply current | $f_{DOSCRC}$, $f_{DOSCLC}$ = $f_{xtal}$ | | | | |
| | | $f_{xtal}$ = 10 MHz | – | 5 | 10 | mA |
| | | $f_{xtal}$ = 6 MHz | – | 3.5 | 7 | mA |
| | | $f_{DOSCRC}$, $f_{DOSCLC}$ = stop | | | | |
| | | $f_{xtal}$ = 10 MHz | – | 3 | 6 | mA |
| | | $f_{xtal}$ = 6 MHz | – | 1.5 | 4 | mA |
| **Input Ports P00, P01, DP00, DP01 and DP02** | | | | | | |
| $V_{IL}$ | LOW level input voltage | $V_{DD}$ = 4.5 V to 5.5 V | 0 | – | 0.3$V_{DD}$ | V |
| $V_{IH}$ | HIGH level input voltage | | 0.7$V_{DD}$ | – | $V_{DD}$ | V |
| $I_{LI}$ | input leakage current | $V_{DD}$ = 4.5 V to 5.5 V; $V_{SS} < V_I < V_{DD}$ | – | – | ±10 | µA |
| **Output Port P00** | | | | | | |
| $V_{OL}$ | LOW level output voltage | $I_{OL}$ = 10 mA | – | – | 1.2 | V |
| $I_{OH}$ | HIGH level pull-up output source current | $V_O$ = 0.7$V_{DD}$ | −40 | −100 | – | µA |
| | HIGH level pull-up output source current | $V_O$ = $V_{SS}$ | – | −140 | −400 | µA |
| | HIGH level push-pull output source current | $V_O$ = $V_{DD}$ − 0.4 V | −3.0 | −7.0 | – | mA |
| **DP00/PWM00 to DP07/PWM07 as derivative Port** | | | | | | |
| $I_{OL}$ | LOW level output sink current | $V_{OL}$ = 0.4 V | 5.0 | 12.0 | – | mA |
| $I_{OH}$ | HIGH level pull-up output source current | $V_O$ = 0.7$V_{DD}$ | −40 | −100 | – | µA |
| | HIGH level pull-up output source current | $V_O$ = $V_{SS}$ | – | −140 | −400 | µA |
| | HIGH level push-pull output source current | $V_O$ = $V_{DD}$ − 0.4 V | −3.0 | −7.0 | – | mA |

| SYMBOL | PARAMETER | CONDITIONS | MIN. | TYP. | MAX. | UNIT |
|---|---|---|---|---|---|---|
| **DP00/PWM00 to DP07/PWM07 as PWM output Port** | | | | | | |
| $I_{OL}$ | LOW level output sink current | $V_{OL}$ = 0.4 V | 0.7 | 1.5 | – | mA |
| $I_{OH}$ | HIGH level pull-up output source current | $V_O = 0.7V_{DD}$ | –40 | –100 | – | µA |
| | HIGH level pull-up output source current | $V_O = V_{SS}$ | – | –140 | –400 | µA |
| | HIGH level push-pull output source current | $V_O = V_{DD} - 0.4$ V | –0.7 | –1.5 | – | mA |
| **Port P10 to P13 outputs** | | | | | | |
| $I_{OL}$ | LOW level output sink current | $V_{OL}$ = 0.4 V | 5.0 | 12.0 | – | mA |
| $I_{OH}$ | HIGH level pull-up output source current | $V_O = 0.7V_{DD}$ | –40 | –100 | – | µA |
| | HIGH level pull-up output source current | $V_O = V_{SS}$ | – | –140 | –400 | µA |
| | HIGH level push-pull output source current | $V_O = V_{DD} - 0.4$ V | –3.0 | –7.0 | – | mA |
| **Outputs VOB and VOW2** | | | | | | |
| $I_{OL}$ | LOW level output sink current | $V_{OL}$ = 0.4 V | 1.4 | 3 | – | mA |
| $I_{OH}$ | HIGH level pull-up output source current | $V_O = 0.7V_{DD}$ | –40 | –100 | – | µA |
| | HIGH level pull-up output source current | $V_O = V_{SS}$ | – | –140 | –400 | µA |
| | HIGH level push-pull output source current | $V_O = V_{DD} - 0.4$ V | –1.4 | –3 | – | mA |
| **DP10/AFC1, DP11/AFC2 and DP12/AFC3 as derivative output Port** | | | | | | |
| $I_{OL}$ | LOW level output sink current | $V_{OL}$ = 0.4 V | 5.0 | 12.0 | – | mA |
| $I_{OH}$ | HIGH level pull-up output source current | $V_O = 0.7V_{DD}$ | –40 | –100 | – | µA |
| | HIGH level pull-up output source current | $V_O = V_{SS}$ | – | –140 | –400 | µA |
| | HIGH level push-pull output source current | $V_O = V_{DD} - 0.4$ V | –3.0 | –7.0 | – | mA |
| **DP20/SDA and DP21/SCL outputs** | | | | | | |
| $I_{OL}$ | LOW level output sink current | $V_{OL}$ = 0.4 V | 3.0 | – | – | mA |
| $I_{OH}$ | HIGH level pull-up output source current | $V_O = 0.7V_{DD}$ | –40 | –100 | – | µA |
| | HIGH level pull-up output source current | $V_O = V_{SS}$ | – | –140 | –400 | µA |
| | HIGH level push-pull output source current | $V_O = V_{DD} - 0.4$ V | – | –7.0 | – | mA |
| **DP22/VOW1, DP23/VOW0 and DP13/TDAC as derivative output Port** | | | | | | |
| $I_{OL}$ | LOW level output sink current | $V_{OL}$ = 0.4 V | 5.0 | 12.0 | – | mA |
| $I_{OH}$ | HIGH level pull-up output source current | $V_O = 0.7V_{DD}$ | –40 | –100 | – | µA |
| | HIGH level pull-up output source current | $V_O = V_{SS}$ | – | –140 | –400 | µA |
| | HIGH level push-pull output source current | $V_O = V_{DD} - 0.4$ V | –3.0 | –7.0 | – | mA |
| **DP22/VOW1 and DP23/VOW0 as VOWi output** | | | | | | |
| $I_{OL}$ | LOW level output sink current | $V_{OL}$ = 0.4 V | 1.4 | 3.0 | – | mA |
| $I_{OH}$ | HIGH level pull-up output source current | $V_O = 0.7V_{DD}$ | –40 | –100 | – | µA |
| | HIGH level pull-up output source current | $V_O = V_{SS}$ | – | –140 | –400 | µA |
| | HIGH level push-pull output source current | $V_O = V_{DD} - 0.4$ V | –1.4 | –3.0 | – | mA |

| SYMBOL | PARAMETER | CONDITIONS | MIN. | TYP. | MAX. | UNIT |
|---|---|---|---|---|---|---|
| **DP13/TDAC as TDAC output** | | | | | | |
| $I_{OL}$ | LOW level output sink current | $V_{OL}$ = 0.4 V | 1.4 | 3.0 | – | mA |
| $I_{OH}$ | HIGH level pull-up output source current | $V_O = 0.7V_{DD}$ | –40 | –100 | – | µA |
| | HIGH level pull-up output source current | $V_O = V_{SS}$ | – | –140 | –400 | µA |
| | HIGH level push-pull output source current | $V_O = V_{DD} - 0.4$ V | –1.4 | –3.0 | – | mA |
| **EMU/TEST, $\overline{\text{RESET}}$, $\overline{\text{INT}}$/T0, T1, $\overline{\text{HSYNC}}$, $\overline{\text{VSYNC}}$ and T3 (Schmitt-trigger input)** | | | | | | |
| $V_{IL}$ | LOW level input voltage | $V_{DD}$ = 4.5 V to 5.5 V | 0 | – | $0.3V_{DD}$ | V |
| $V_{IH}$ | HIGH level input voltage | | $0.7V_{DD}$ | – | $V_{DD}$ | V |
| $I_{LI}$ | input leakage current | $V_{DD}$ = 4.5 V to 5.5 V; $V_{SS} < V_I < V_{DD}$ | –1.0 | – | +1.0 | µA |

## 18 AC CHARACTERISTICS

$V_{DD}$ = 5 V; $T_{amb}$ = –20 to +70 °C; all voltages with respect to $V_{SS}$; unless otherwise specified.

| SYMBOL | PARAMETER | CONDITIONS | MIN. | TYP. | MAX. | UNIT |
|---|---|---|---|---|---|---|
| **Oscillator** | | | | | | |
| $f_{xtal}$ | crystal frequency (note 1) | | 1 | – | 10.0 | MHz |
| $f_{osc\text{-}XTAL}$ | oscillator frequency; option 1 | $g_m$ = 0.4 mS (typ.) | 1 | – | 6.0 | MHz |
| $f_{osc\text{-}PXE}$ | | | not allowed | | | MHz |
| $f_{osc\text{-}XTAL}$ | oscillator frequency; option 2 | $g_m$ = 1.6 mS (typ.) | 4.0 | – | 10.0 | MHz |
| $f_{osc\text{-}PXE}$ | | | 1.0 | – | 6.0 | MHz |
| $f_{osc\text{-}XTAL}$ | oscillator frequency; option 3 | $g_m$ = 4.5 mS (typ.) | not allowed | | | MHz |
| $f_{osc\text{-}PXE}$ | | | 3.0 | – | 10.0 | MHz |
| $C_{XTAL1}$ | external capacitance at XTAL1 | | | | | |
| | with XTAL resonator | | not required | | | pF |
| | with PXE resonator | | – | 30 | 100 | pF |
| $C_{XTAL2}$ | external capacitance at XTAL2 | | | | | |
| | with XTAL resonator | | not required | | | pF |
| | with PXE resonator | | – | 30 | 100 | pF |
| $f_{DOSC}$ | on-screen-display clock frequency | | 4.0 | 8.0 | 12.0 | MHz |

**Note**

1. Oscillator with three (3) options for optimum use.

## 19 AFC CHARACTERISTICS

| SYMBOL | PARAMETER | MIN. | TYP. | MAX. | UNIT |
|---|---|---|---|---|---|
| $T_{AFC}$ | conversion time (from any change in the AFC: channel number, voltage level, enable/disable) with $f_{xtal}$ = 10 MHz | – | – | 7 | µs |
| **DP10/AFC1, DP11/AFC2 and DP12/AFC3 comparator input** | | | | | |
| $V_{AI}$ | comparator analog input voltage | $V_{SS}$ | – | $V_{DD}$ | V |
| $V_{AE}$ | conversion error range | – | – | ±0.5 | LSB |

# SECTION 4 33XX DERIVATIVES

CONTENTS

## 1 FEATURES

- 8-bit CPU, ROM, RAM, I/O in a single 28-lead package
- 1536 bytes ROM
- 160 bytes RAM
- Over 100 instructions (based on MAB8048) all of 1 or 2 cycles
- 20 quasi-bidirectional I/O port lines
- 8-bit programmable timer/event counter 1
- 2 single-level vectored interrupts:
  - external
  - 8-bit programmable timer/event counter 1
- Two test inputs, one of which also serves as the external interrupt input
- Power-on reset
- Stop and Idle modes
- Logic supply voltage: $V_{DD}$ = 1.8 to 6 V
- Low stand-by voltage: $V_{DD}$ = 1 V
- Low stand-by current: $I_{DD}$ = 1.2 µA typical at 1.8 V and 25 °C
- Clock frequency: 1 to 16 MHz
- Operating temperature: –25 to +70 °C
- Manufactured in silicon gate CMOS process.

## 2 GENERAL DESCRIPTION

This data sheet details the specific properties of the PCD3315A. The shared characteristics of the PCD33XXA family of microcontrollers are described in the *"PCD33XXA family data sheet"* which should be read in conjunction with this publication.

The PCD3315A is a microcontroller intended for telecom applications. It provides 1.5 kbytes of program memory, 160 bytes of RAM and 20 I/O lines. The instruction set is based on that of the MAB8048 and is software compatible with the PCD33XXA family.

## 3 ORDERING INFORMATION

| TYPE NUMBER | PACKAGE | | | |
|---|---|---|---|---|
| | PINS | PIN POSITION | MATERIAL | CODE |
| PCD3315AP | 28 | DIP | plastic | SOT117-1 |
| PCD3315AT | 28 | SO28L | plastic | SOT136-1 |

## 4 BLOCK DIAGRAM

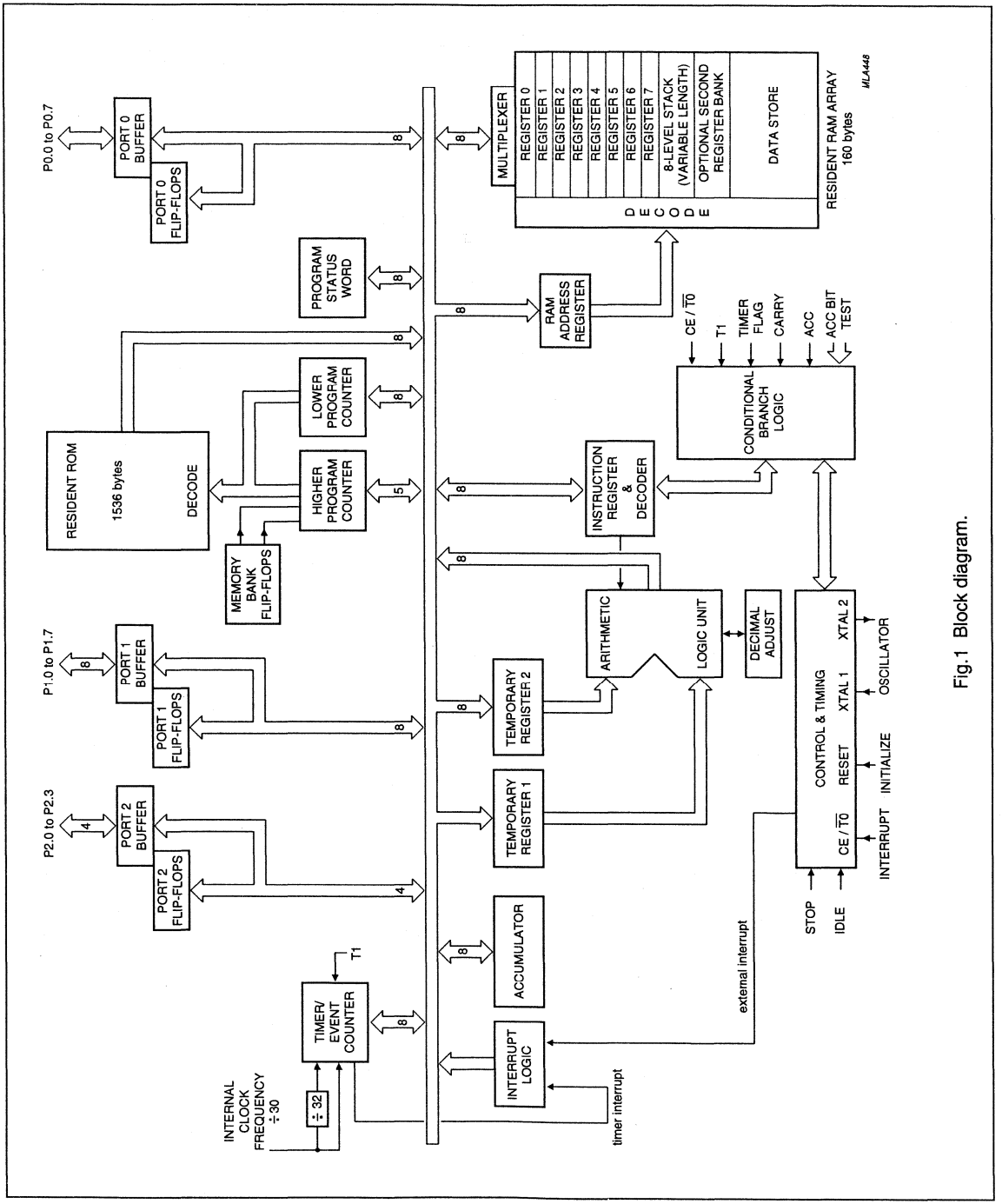



Fig.1 Block diagram.

## 5 PINNING INFORMATION

### 5.1 Pinning

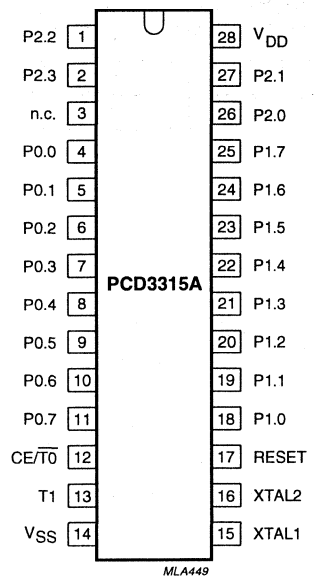



Fig.2 Pin configuration.

### 5.2 Pin description

**Table 1** DIP28 and SO28L packages.

| SYMBOL | PIN | TYPE | DESCRIPTION |
|---|---|---|---|
| P2.2 to P2.3 | 1 and 2 | I/O | Port 2: quasi-bidirectional I/O lines |
| n.c. | 3 | – | not connected |
| P0.0 to P0.7 | 4 to 11 | I/O | Port 0: quasi-bidirectional I/O lines |
| CE/$\overline{T0}$ | 12 | I | Chip enable/Test 0 |
| T1 | 13 | I | Test 1/count input of 8-bit timer/event counter 1 |
| $V_{SS}$ | 14 | P | ground |
| XTAL1 | 15 | I | crystal oscillator/ external clock |
| XTAL2 | 16 | O | crystal oscillator output |
| RESET | 17 | I | Reset input |
| P1.0 to P1.7 | 18 to 25 | I/O | Port 1: quasi-bidirectional I/O lines |
| P2.0 to P2.1 | 26 to 27 | I/O | Port 2: quasi-bidirectional I/O lines |
| $V_{DD}$ | 28 | P | Positive supply |

## 6 PARALLEL PORTS

All standard quasi-bidirectional I/O ports are available:

- Port 0 parallel port of 8 lines (P0.0 to P0.7)
- Port 1 parallel port of 8 lines (P1.0 to P1.7)
- Port 2 parallel port of 4 lines (P2.0 to P2.3)

Since no serial I/O interface is provided, P2.3 is a general purpose line without restrictions, i.e. the reset option as well as mask options 1 and 3 are also available.

## 7 INSTRUCTION SET

Since no serial I/O interface nor derivative logic is provided, the serial input/output instructions are not available.

Since the ROM space is restricted to 1536 bytes, the SEL MB1/2/3 instructions refer to non-existing program memory banks and should therefore be avoided.

Since the RAM space is restricted to 160 bytes, care should be taken to avoid accesses to non-existing RAM locations.

## 8 SUMMARY OF MASK OPTIONS

**Table 2** Port mask options (see *"PCD33XXA family data sheet"*).

| PORT | PORT OUTPUT[1] | | | PORT STATE AFTER RESET | |
|---|---|---|---|---|---|
| | OPTION 1 | OPTION 2 | OPTION 3 | SET | RESET |
| P0.0 to P0.7 | X | X | X | X | X |
| P1.0 to P1.7 | X | X | X | X | X |
| P2.0 to P2.3 | X | X | X | X | X |

**Note**

1. Option 1: normal port
   Option 2: open drain
   Option 3: push-pull.

**Table 3** Mask options.

| FEATURE | DESCRIPTION |
|---|---|
| ROM Code: program/data | Any mix of instructions and data up to ROM size of 1536 bytes. |
| Power-on reset voltage level: $V_{ref}$ | 1.2 to 3.6 V in increments of 100 mV; OFF |
| Oscillator transconductance: $g_m$ | LOW transconductance: $g_{mL}$ |
| | MEDIUM transconductance: $g_{mM}$ |
| | HIGH transconductance: $g_{mH}$ |

## 9 LIMITING VALUES

In accordance with the Absolute Maximum Rating System (IEC 134).

| SYMBOL | PARAMETER | MIN. | MAX. | UNIT |
|---|---|---|---|---|
| $V_{DD}$ | supply voltage | −0.5 | 7 | V |
| $V_I$ | all input voltages | −0.5 | $V_{DD}$ + 0.5 | V |
| $I_{DD}$ | positive supply current | −50 | +50 | mA |
| $I_I$ | DC input current | −10 | +10 | mA |
| $I_O$ | DC output current | −10 | +10 | mA |
| $P_{tot}$ | total power dissipation | – | 125 | mW |
| $P_O$ | power dissipation per output | – | 30 | mW |
| $I_{SS}$ | ground supply current | −50 | +50 | mA |
| $T_{stg}$ | storage temperature | −65 | +150 | °C |
| $T_j$ | operating junction temperature | – | 90 | °C |

## 10 HANDLING

Inputs and outputs are protected against electrostatic discharge in normal handling. However, it is good practice to take normal precautions appropriate to handling MOS devices (see *"Handling MOS devices"*).

## 11 DC CHARACTERISTICS

$V_{DD}$ = 1.8 to 6 V; $V_{SS}$ = 0 V; $T_{amb}$ = −25 to +70 °C; all voltages with respect to $V_{SS}$; $f_{xtal}$ = 3.579545 MHz ($g_{mL}$); $R_X \leq 100\ \Omega$; unless otherwise specified.

| SYMBOL | PARAMETER | CONDITIONS | MIN. | TYP. | MAX. | UNIT |
|---|---|---|---|---|---|---|
| **Supply (see Figs 3 to 6)** | | | | | | |
| $V_{DD}$ | operating supply voltage | note 1 | 1.8 | – | 6 | V |
| $V_{DD}$ | RAM data retention Stop mode | | 1.0 | | 6 | V |
| $I_{DD}$ | operating supply current | $V_{DD}$ = 3 V; $f_{xtal}$ = 3.579545 MHz ($g_{mL}$); note 1 | – | 0.3 | 0.6 | mA |
| | | $V_{DD}$ = 5 V; $f_{xtal}$ = 10 MHz ($g_{mL}$); note 1 | – | 1.1 | 3.0 | mA |
| | | $V_{DD}$ = 5 V; $f_{xtal}$ = 16 MHz ($g_{mM}$); note 1 | – | 1.7 | 5.0 | mA |
| | | $V_{DD}$ = 5 V; $f_{xtal}$ = 16 MHz ($g_{mH}$); note 1 | – | 2.5 | 6.0 | mA |
| $I_{DD(ID)}$ | supply current Idle mode | $V_{DD}$ = 3 V; $f_{xtal}$ = 3.579545 MHz ($g_{mL}$); note 1 | – | 0.2 | 0.4 | mA |
| | | $V_{DD}$ = 5 V; $f_{xtal}$ = 10 MHz ($g_{mL}$); note 1 | – | 0.8 | 1.6 | mA |
| | | $V_{DD}$ = 5 V; $f_{xtal}$ = 16 MHz ($g_{mM}$); note 1 | – | 1.2 | 4.0 | mA |
| | | $V_{DD}$ = 5 V; $f_{xtal}$ = 16 MHz ($g_{mH}$); note 1 | – | 1.7 | 5.0 | mA |
| $I_{DD(ST)}$ | supply current Stop mode | $V_{DD}$ = 1.8 V; $T_{amb}$ = 25 °C; note 2 | – | 1.2 | 2.5 | µA |
| | | $V_{DD}$ = 1.8 V; $T_{amb}$ = 70 °C; note 2 | – | – | 10 | µA |
| **Inputs** | | | | | | |
| $V_{IL}$ | LOW level input voltage | | 0 | – | $0.3V_{DD}$ | V |
| $V_{IH}$ | HIGH level input voltage | | $0.7V_{DD}$ | – | $V_{DD}$ | V |
| $I_{LI}$ | Input leakage current | $V_{SS} \leq V_I \leq V_{DD}$ | −1 | – | +1 | µA |
| **Outputs (see Figs 7 to 9)** | | | | | | |
| $I_{OL}$ | LOW level port sink current | $V_{DD}$ = 3 V; $V_O$ = 0.4 V | 0.7 | 8 | – | mA |
| $I_{OH}$ | HIGH level port pull-up source current | $V_{DD}$ = 3 V; $V_O$ = 2.7 V | −10 | −20 | – | µA |
| | | $V_{DD}$ = 3 V; $V_O$ = 0 V | – | −100 | −300 | µA |
| $I_{OH}$ | HIGH level port push-pull source current | $V_{DD}$ = 3 V; $V_O$ = 2.6 V | −0.7 | −4 | – | mA |
| $\Delta V_{POR}$ | power-on reset level variation around chosen $V_{POR}$ | note 3 | −0.5 | 0 | +0.5 | V |

**Notes**

1. $V_{IL}$ = $V_{SS}$; $V_{IH}$ = $V_{DD}$; open drain outputs connected to $V_{SS}$; all other outputs open. Maximum values: external clock at XTAL1; XTAL2 open. Typical values: $T_{amb}$ = 25 °C; crystal connected between XTAL1 and XTAL2.
2. $V_{IL}$ = $V_{SS}$; $V_{IH}$ = $V_{DD}$; open drain outputs connected to $V_{SS}$; RESET, T1 and CE/$\overline{T0}$ at $V_{SS}$; crystal connected between XTAL1 and XTAL2; all other outputs open.
3. $V_{POR}$ is an option chosen by the user. Depending on its value, it may restrict the supply voltage range.

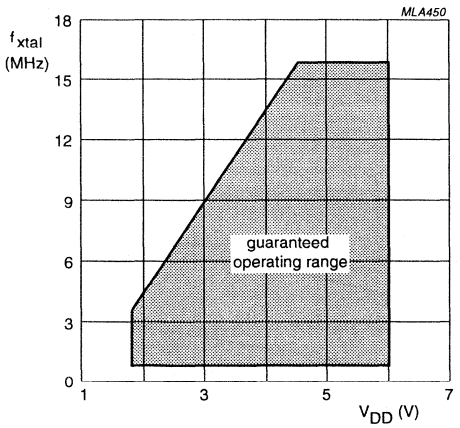



Fig.3 Maximum clock frequency ($f_{xtal}$) as a function of supply voltage ($V_{DD}$).

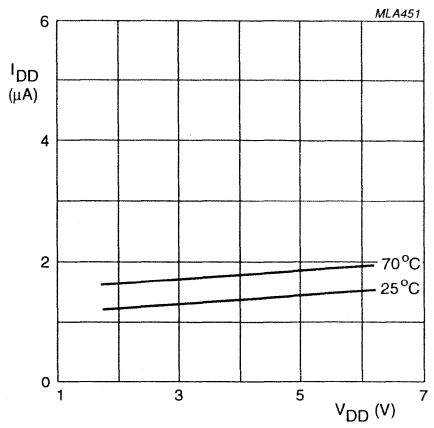



Fig.4 Typical supply current ($I_{DD}$) in Stop mode as a function of supply voltage ($V_{DD}$).

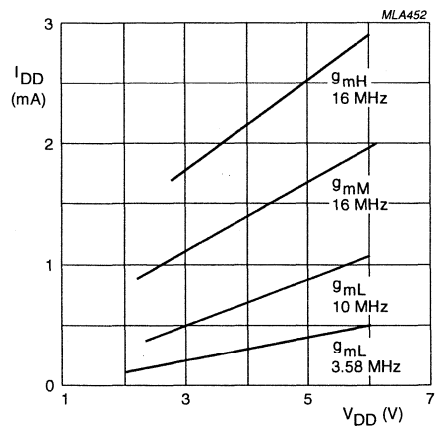



Fig.5 Typical operating supply current ($I_{DD}$) as a function of supply voltage ($V_{DD}$).

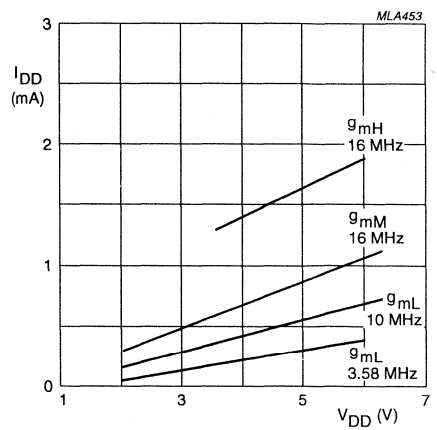
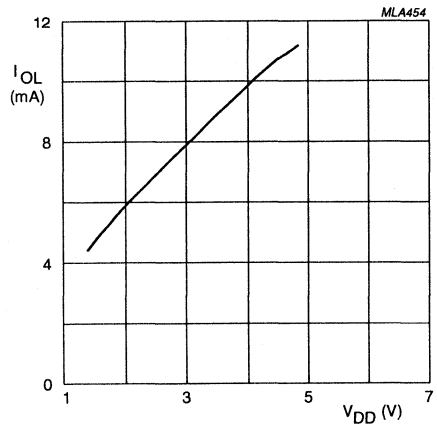



Fig.6 Typical supply current ($I_{DD}$) in Idle mode as a function of supply voltage ($V_{DD}$).




$V_O$ = 0.4 V.

Fig.7 Typical LOW level port output sink current ($I_{OL}$) as a function of supply voltage ($V_{DD}$).

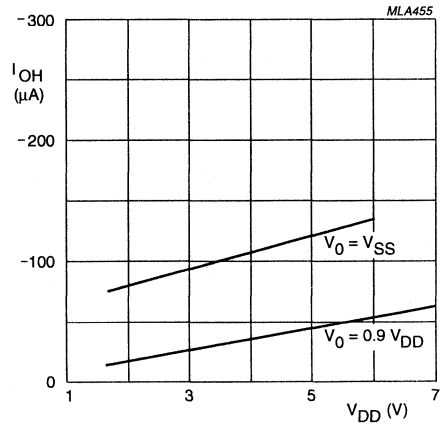
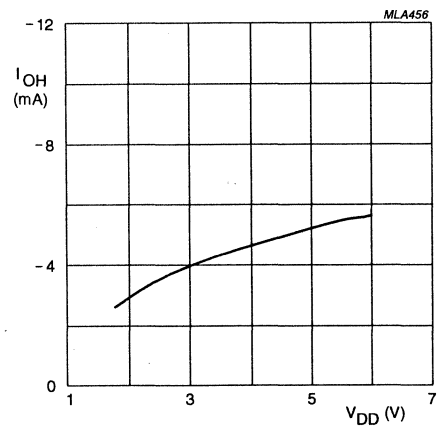



Fig.8 Typical HIGH level output pull-up source current ($I_{OH}$) as a function of supply voltage ($V_{DD}$).




$V_O = V_{DD} - 0.4$ V.

Fig.9 Typical HIGH level push-pull output source current ($I_{OH}$) as a function of supply voltage ($V_{DD}$).

## 12 AC CHARACTERISTICS

$V_{DD}$ = 1.8 to 6 V; $V_{SS}$ = 0 V; $T_{amb}$ = −25 to +70 °C; all voltages with respect to $V_{SS}$; unless otherwise specified. $V_{IL} = V_{SS}$; $V_{IH} = V_{DD}$; open drain outputs connected to $V_{SS}$; all other outputs open. Maximum values: external clock at XTAL1; XTAL2 open. Typical values: $T_{amb}$ = 25 °C; crystal connected between XTAL1 and XTAL2.

| SYMBOL | PARAMETER | CONDITIONS | MIN. | TYP. | MAX. | UNIT |
|---|---|---|---|---|---|---|
| $t_r$ | rise time all outputs | $V_{DD}$ = 5 V; $T_{amb}$ = 25 °C; $C_L$ = 50 pF | – | 30 | – | ns |
| $t_f$ | fall time all outputs | $V_{DD}$ = 5 V; $T_{amb}$ = 25 °C; $C_L$ = 50 pF | – | 30 | – | ns |
| $f_{xtal}$ | clock frequency | see Fig.3 | 1 | – | 16 | MHz |
| **Oscillator (see Fig.10)** | | | | | | |
| $g_{mL}$ | LOW transconductance | $V_{DD}$ = 5 V | 0.2 | 0.4 | 1.0 | mS |
| $g_{mM}$ | MEDIUM transconductance | $V_{DD}$ = 5 V | 0.9 | 1.6 | 3.2 | mS |
| $g_{mH}$ | HIGH transconductance | $V_{DD}$ = 5 V | 3.0 | 4.5 | 9.0 | mS |
| $R_F$ | feedback resistor | | 0.3 | 1.0 | 3.0 | MΩ |

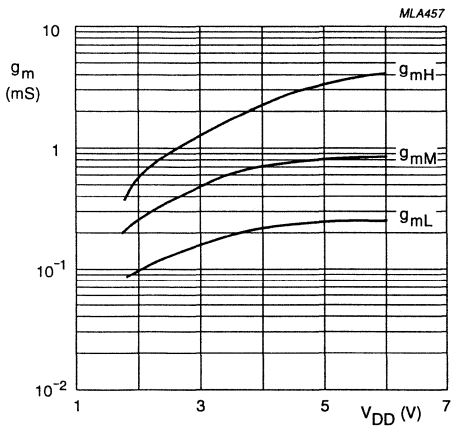



Fig.10 Typical transconductance as a function of supply voltage ($V_{DD}$) for the options $g_{mL}$, $g_{mM}$ and $g_{mH}$.

# Telecom microcontroller with serial I/O interface

# PCD3343A PCD3348A

CONTENTS

# Telecom microcontroller with serial I/O interface

# PCD3343A PCD3348A

## 1 FEATURES

- 8-bit CPU, ROM, RAM, I/O in a single 28-lead package
- 3 kbytes ROM, 224 bytes RAM (PCD3343A)
- 8 kbytes ROM, 256 bytes RAM (PCD3348A)
- Serial I/O interface with multi-master capability
- Over 100 instructions (based on MAB8048) all of 1 or 2 cycles
- 20 quasi-bidirectional I/O port lines
- 8-bit programmable timer/event counter 1
- 3 single-level vectored interrupts:
  - external
  - 8-bit programmable timer/event counter 1
  - SIO/derivative
- Two test inputs, one of which also serves as the external interrupt input
- Power-on reset
- Stop and Idle modes
- Logic supply voltage: $V_{DD}$ = 1.8 to 6 V
- Low stand-by voltage: $V_{DD}$ = 1 V
- Low stand-by current: $I_{DD}$ = 2 μA typical
- Clock frequency: 1 to 16 MHz
- Oscillator with output drive for peripherals (e.g. PCD3312 DTMF generator)
- Operating temperature: –25 to +70 °C
- Manufactured in silicon gate CMOS process.

## 2 GENERAL DESCRIPTION

This data sheet details the specific properties of the PCD3343A and PCD3348A. The shared characteristics of the PCD33XXA family of microcontrollers are described in the *"PCD33XXA family data sheet"*, which should be read in conjunction with this publication.

The PCD3343A and PCD3348A are microcontrollers intended for telecom applications. They provide 3 and 8 kbytes of program memory and 224 and 256 bytes of RAM, respectively. In addition to 20 I/O port lines, the microcontrollers provide an on-chip serial I/O interface.

This two-line serial bus extends the microcontroller capabilities when implemented with the powerful $I^2C$-bus devices of the PCF85XX, PCD33XX and 'Clips' peripheral families. These include liquid crystal display drivers, pulse and/or DTMF diallers, ringers, AD/DA converters, clock/calendar circuits, EEPROM and RAM.

The instruction set is based on that of the MAB8048 and is software compatible with the PCD33XXA family.

## 3 ORDERING INFORMATION

| TYPE NUMBER | PACKAGE | | | |
|---|---|---|---|---|
| | PINS | PIN POSITION | MATERIAL | CODE |
| PCD3343AP | 28 | DIP | plastic | SOT117-1 |
| PCD3348AP | 28 | DIP | plastic | SOT117-1 |
| PCD3343AT | 28 | SO28L | plastic | SOT136-1 |
| PCD3348AT | 28 | SO28L | plastic | SOT136-1 |

## 4 BLOCK DIAGRAM

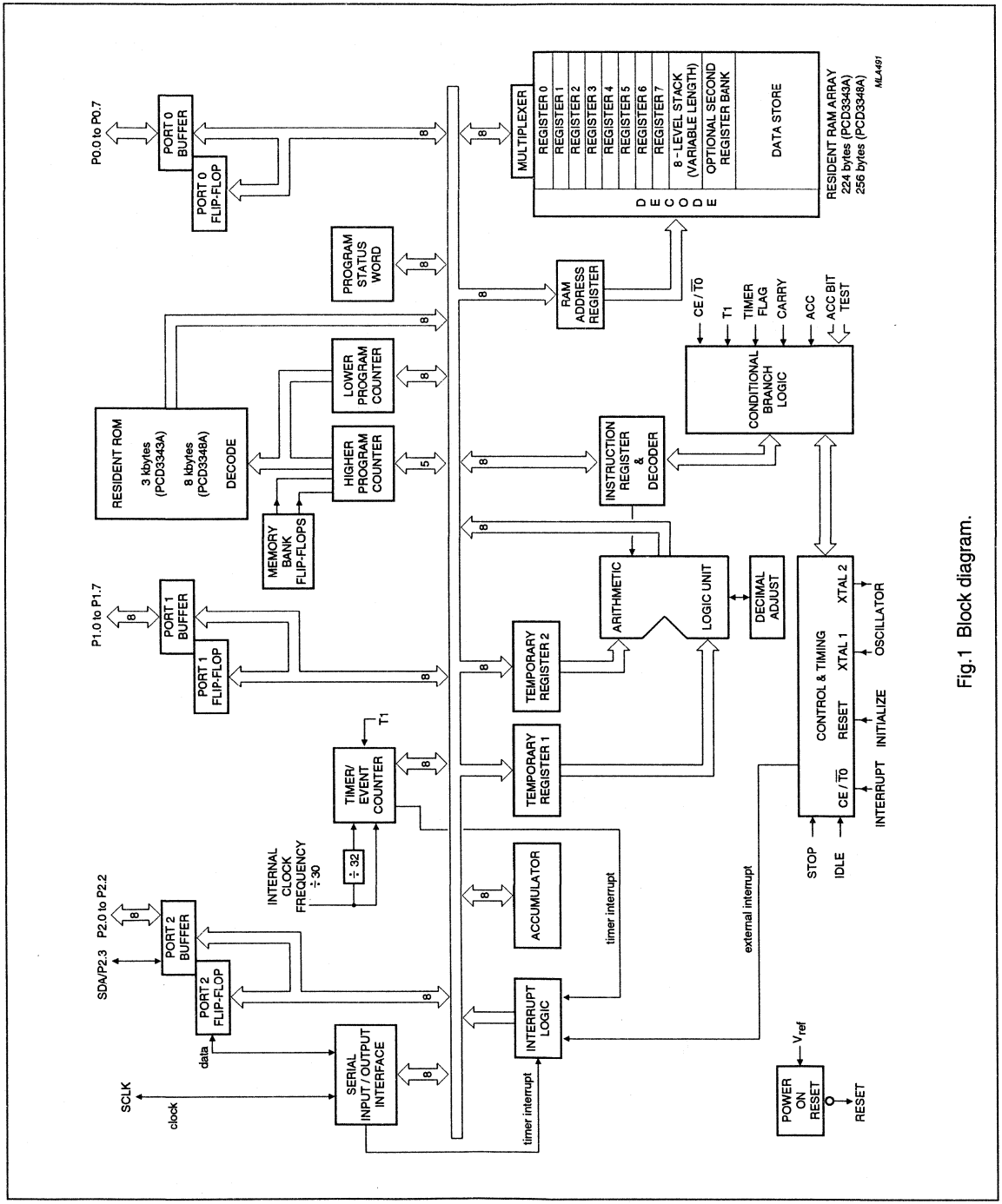



Fig.1 Block diagram.

## 5 PINNING INFORMATION

### 5.1 Pinning

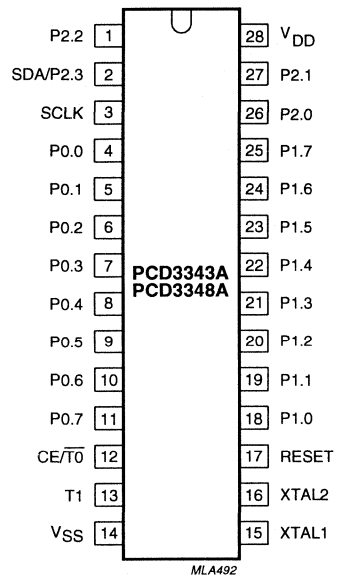



Fig.2 Pin configuration.

### 5.2 Pin description

**Table 1** DIP28 and SO28L packages.

| SYMBOL | PIN | TYPE | DESCRIPTION |
|---|---|---|---|
| P2.2 | 1 | I/O | Port 2: quasi-bidirectional I/O line |
| SDA/P2.3 | 2 | I/O | bidirectional data line of the serial I/O interface/ Port 2: quasi-bidirectional I/O line |
| SCLK | 3 | I/O | bidirectional clock line of the serial I/O interface |
| P0.0 to P0.7 | 4 to 11 | I/O | Port 0: quasi-bidirectional I/O lines |
| CE/$\overline{T0}$ | 12 | I | Chip Enable/Test 0 |
| T1 | 13 | I | Test 1/count input of 8-bit timer/event counter 1 |
| $V_{SS}$ | 14 | P | ground |
| XTAL1 | 15 | I | crystal oscillator/external clock |
| XTAL2 | 16 | O | crystal oscillator output |
| RESET | 17 | I | Reset input |
| P1.0 to P1.7 | 18 to 25 | I/O | Port 1: quasi-bidirectional I/O lines |
| P2.0 to P2.1 | 26 and 27 | I/O | Port 2: quasi-bidirectional I/O lines |
| $V_{DD}$ | 28 | P | positive supply |

## 6 INSTRUCTION SET

The PCD3343A has ROM space restricted to 3 kbytes. The instructions SEL MB1/2/3 would therefore define non-existing program memory banks and should be avoided. Additionally, RAM space is restricted to 224 bytes for the PCD3343A, so care should be taken to avoid accesses to non-existing RAM locations.

## 7 SUMMARY OF MASK OPTIONS

**Table 2** Port mask options (see *"PCD33XXA family data sheet"*).

| PORT | PORT OUTPUT[1] | | | PORT STATE AFTER RESET | |
|---|---|---|---|---|---|
| | OPTION 1 | OPTION 2 | OPTION 3 | SET | RESET |
| P0.0 to P0.7 | X | X | X | X | X |
| P1.0 to P1.7 | X | X | X | X | X |
| P2.0 to P2.3 | X | X | X | X | X |
| SDA /P2.3 | – | X | – | X | – |

**Note**

1. Option 1: normal port
   Option 2: open drain
   Option 3: push-pull.

**Table 3** Mask options.

| FEATURE | DESCRIPTION |
|---|---|
| ROM Code: program/data | Any mix of instructions and data up to ROM size of 3 kbytes (PCD3343A) and 8 kbytes (PCD3348A). |
| Power-on reset voltage level: $V_{ref}$ | 1.2 to 3.6 V in increments of 100 mV; OFF |
| Oscillator transconductance: $g_m$ | LOW transconductance: $g_{mL}$ |
| | MEDIUM transconductance: $g_{mM}$ |
| | HIGH transconductance: $g_{mH}$ |

## 8 LIMITING VALUES

In accordance with the Absolute Maximum Rating System (IEC 134).

| SYMBOL | PARAMETER | MIN. | MAX. | UNIT |
|---|---|---|---|---|
| $V_{DD}$ | supply voltage | –0.5 | +7 | V |
| $V_I$ | all input voltages | –0.5 | $V_{DD}$ + 0.5 | V |
| $I_I$ | DC input current | –10 | +10 | mA |
| $I_O$ | DC output current | –10 | +10 | mA |
| $P_{tot}$ | total power dissipation | – | 125 | mW |
| $P_O$ | power dissipation per output | – | 30 | mW |
| $I_{SS}$ | ground supply current | –50 | +50 | mA |
| $T_{stg}$ | storage temperature | –65 | +150 | °C |
| $T_j$ | operating junction temperature | – | 90 | °C |

## 9 HANDLING

Inputs and outputs are protected against electrostatic discharge in normal handling. However, it is good practice to take normal precautions appropriate to handling MOS devices (see *"Handling MOS devices"*).

## 10 DC CHARACTERISTICS

$V_{DD}$ = 1.8 to 6 V; $V_{SS}$ = 0 V; $T_{amb}$ = −25 to +70 °C; all voltages with respect to $V_{SS}$; $f_{xtal}$ = 3.579545 MHz ($g_{mL}$); $R_X \leq 100\ \Omega$; unless otherwise specified.

| SYMBOL | PARAMETER | CONDITIONS | MIN. | TYP. | MAX. | UNIT |
|---|---|---|---|---|---|---|
| **Supply (see Figs 3 to 8)** | | | | | | |
| $V_{DD}$ | operating supply voltage | note 1 | 1.8 | – | 6 | V |
| $V_{DD}$ | RAM data retention Stop mode | | 1.0 | | 6 | V |
| $I_{DD}$ | operating supply current | $V_{DD}$ = 3 V; note 1 | – | 0.3 | 0.6 | mA |
| | | $V_{DD}$ = 5 V; $f_{xtal}$ = 10 MHz ($g_{mL}$); note 1 | – | 1.1 | 3.0 | mA |
| | | $V_{DD}$ = 5 V; $f_{xtal}$ = 16 MHz ($g_{mM}$); note 1 | – | 1.7 | 5.0 | mA |
| | | $V_{DD}$ = 5 V; $f_{xtal}$ = 16 MHz ($g_{mH}$); note 1 | – | 2.5 | 6.0 | mA |
| $I_{DD(ID)}$ | supply current Idle mode | $V_{DD}$ = 3 V; note 1 | – | 0.2 | 0.4 | mA |
| | | $V_{DD}$ = 5 V; $f_{xtal}$ = 10 MHz ($g_{mL}$); note 1 | – | 0.8 | 1.6 | mA |
| | | $V_{DD}$ = 5 V; $f_{xtal}$ = 16 MHz ($g_{mM}$); note 1 | – | 1.2 | 4.0 | mA |
| | | $V_{DD}$ = 5 V; $f_{xtal}$ = 16 MHz ($g_{mH}$); note 1 | – | 1.7 | 5.0 | mA |
| $I_{DD(ST)}$ | supply current Stop mode | $V_{DD}$ = 1.8 V; $T_{amb}$ = 25 °C; note 2 | – | 1.2 | 2.5 | µA |
| | | $V_{DD}$ = 1.8 V; $T_{amb}$ = 70 °C; note 2 | – | – | 10 | µA |
| **Inputs** | | | | | | |
| $V_{IL}$ | LOW level input voltage | | 0 | – | $0.3V_{DD}$ | V |
| $V_{IH}$ | HIGH level input voltage | | $0.7V_{DD}$ | – | $V_{DD}$ | V |
| $I_{LI}$ | input leakage current | $V_{SS} \leq V_I \leq V_{DD}$ | −1 | – | +1 | µA |
| **Outputs (see Figs 9 to 12)** | | | | | | |
| $I_{OL}$ | LOW level port sink current except SDA/P2.3 and SCLK | $V_{DD}$ = 3 V; $V_O$ = 0.4 V | 0.7 | 8 | – | mA |
| $I_{OL}$ | LOW level SIO sink current SDA/P2.3 and SCLK | $V_{DD}$ = 3 V; $V_O$ = 0.4 V | 1.5 | 8 | – | mA |
| $I_{OH}$ | HIGH level port pull-up source current | $V_O$ = 2.7 V; $V_{DD}$ = 3 V | −10 | −20 | – | µA |
| | | $V_O$ = 0 V; $V_{DD}$ = 3 V | – | −100 | −300 | µA |
| $I_{OH}$ | HIGH level port push-pull source current | $V_{DD}$ = 3 V; $V_O$ = 2.6 V | −0.7 | −4 | – | mA |
| $\Delta V_{POR}$ | power-on reset level variation around chosen $V_{POR}$ | note 3 | −0.5 | 0 | +0.5 | V |

**Notes**

1. $V_{IL} = V_{SS}$; $V_{IH} = V_{DD}$; open drain outputs connected to $V_{SS}$; all other outputs open. Maximum values: external clock at XTAL1; XTAL2 open. Typical values: $T_{amb}$ = 25 °C; crystal connected between XTAL1 and XTAL2.
2. $V_{IL} = V_{SS}$; $V_{IH} = V_{DD}$; open drain outputs connected to $V_{SS}$; RESET, T1 and CE/$\overline{T0}$ at $V_{SS}$; crystal connected between XTAL1 and XTAL2; all other outputs open.
3. $V_{POR}$ is an option chosen by the user. Depending on its value, it may restrict the supply voltage range.

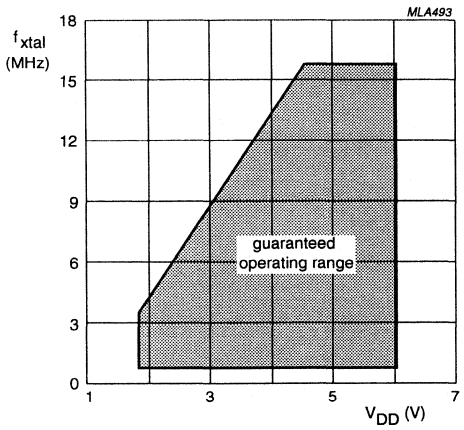



Fig.3 Maximum clock frequency ($f_{xtal}$) as a function of supply voltage ($V_{DD}$).

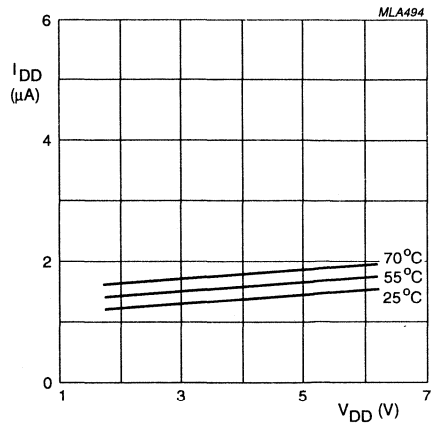



Fig.4 Typical supply current ($I_{DD}$) in Stop mode as a function of supply voltage ($V_{DD}$).

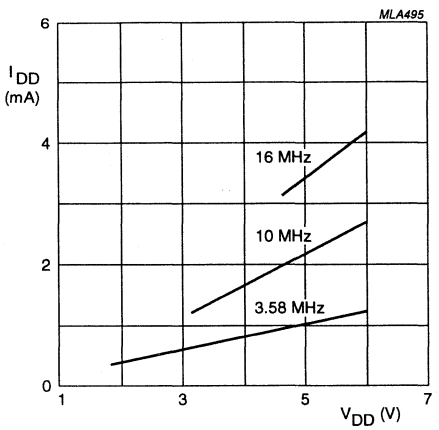



Fig.5 Typical operating supply current ($I_{DD}$) as a function of supply voltage ($V_{DD}$).

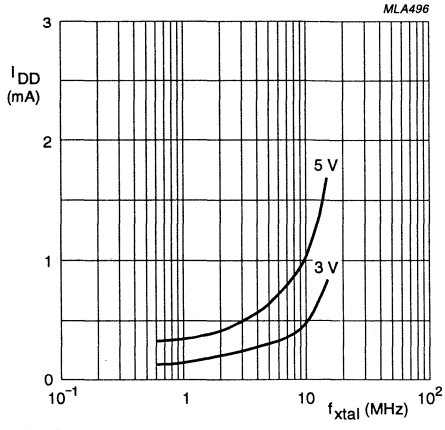



Fig.6 Typical operating supply current ($I_{DD}$) as a function of clock frequency ($f_{xtal}$).

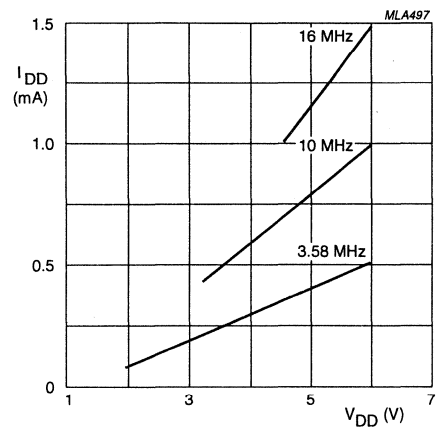



Fig.7 Typical supply current ($I_{DD}$) in Idle mode as a function of supply voltage ($V_{DD}$).

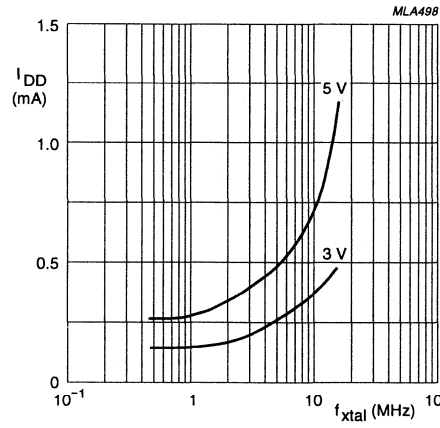
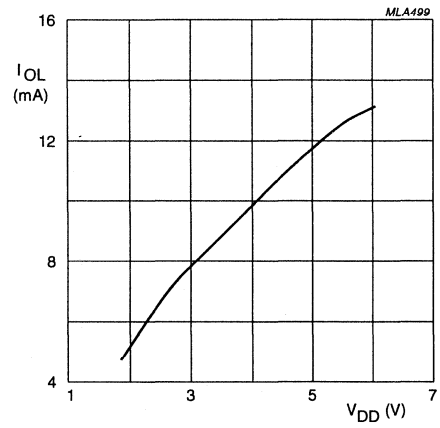
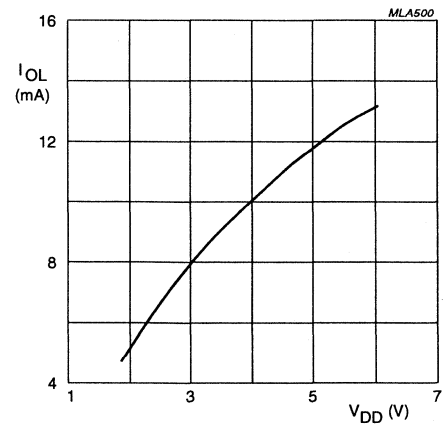



Fig.8 Typical supply current ($I_{DD}$) in Idle mode as a function of clock frequency ($f_{xtal}$).




$V_O$ = 0.4 V.

Fig.9 Typical LOW level port output sink current ($I_{OL}$) as a function of supply voltage ($V_{DD}$).




$V_O$ = 0.4 V.

Fig.10 Typical SDA/P2.3 or SCLK LOW level output sink current ($I_{OL}$) as a function of supply voltage ($V_{DD}$).

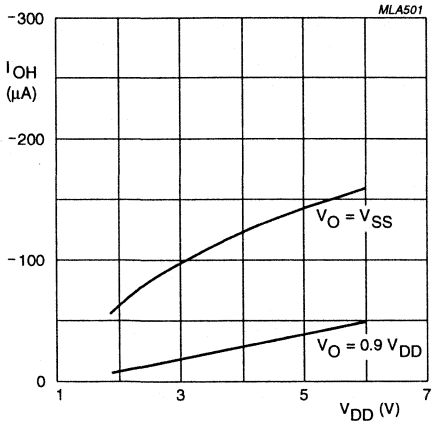
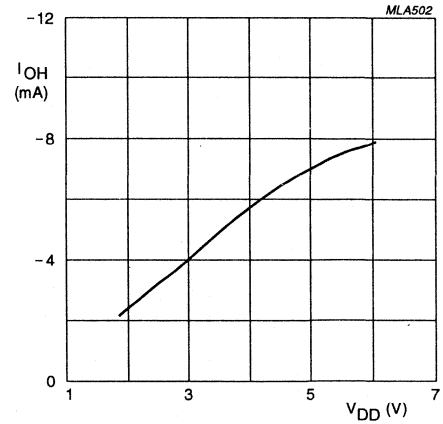



Fig.11 Typical HIGH level output pull-up source current ($I_{OH}$) as a function of supply voltage ($V_{DD}$).




$V_O = V_{DD} - 0.4$ V.

Fig.12 Typical HIGH level push-pull output source current ($I_{OH}$) as a function of supply voltage ($V_{DD}$).

## 11 AC CHARACTERISTICS

$V_{DD}$ = 1.8 to 6 V; $V_{SS}$ = 0 V; $T_{amb}$ = −25 to +70 °C; all voltages with respect to $V_{SS}$; unless otherwise specified. $V_{IL}$ = $V_{SS}$; $V_{IH}$ = $V_{DD}$; open drain outputs connected to $V_{SS}$; all other outputs open. Maximum values: external clock at XTAL1; XTAL2 open. Typical values: $T_{amb}$ = 25 °C; crystal connected between XTAL1 and XTAL2.

| SYMBOL | PARAMETER | CONDITIONS | MIN. | TYP. | MAX. | UNIT |
|---|---|---|---|---|---|---|
| $t_r$ | rise time all outputs | $V_{DD}$ = 5 V; $T_{amb}$ = 25 °C; $C_L$ = 50 pF | – | 30 | – | ns |
| $t_f$ | fall time all outputs | $V_{DD}$ = 5 V; $T_{amb}$ = 25 °C; $C_L$ = 50 pF | – | 30 | – | ns |
| $f_{xtal}$ | clock frequency | see Fig.3 | 1 | – | 16 | MHz |
| **Oscillator (see Fig.13)** | | | | | | |
| $g_{mL}$ | LOW transconductance | $V_{DD}$ = 5 V | 0.2 | 0.4 | 1.0 | mS |
| $g_{mM}$ | MEDIUM transconductance | $V_{DD}$ = 5 V | 0.9 | 1.6 | 3.2 | mS |
| $g_{mH}$ | HIGH transconductance | $V_{DD}$ = 5 V | 3.0 | 4.5 | 9.0 | mS |
| $R_F$ | feedback resistor | | 0.3 | 1.0 | 3.0 | MΩ |

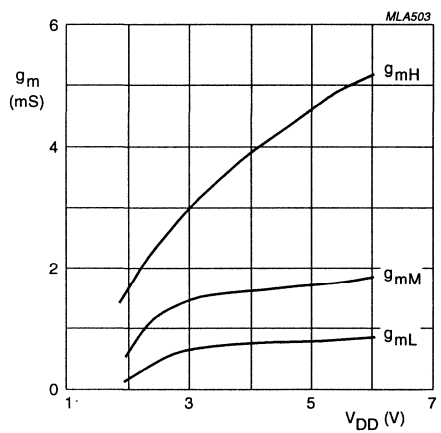



Fig.13 Typical transconductance as a function of supply voltage ($V_{DD}$) for the options $g_{mL}$, $g_{mM}$ and $g_{mH}$.

### 11.1 I²C-bus interface characteristics

**Table 4** I²C-bus timing.

| SYMBOL | PARAMETER | INPUT (see Fig.14) | OUTPUT (see Fig.15; note 1) |
|---|---|---|---|
| **SCLK** | | | |
| $t_{HD;STA}$ | START condition hold time | $\geq \frac{14}{f_{xtal}}$ | $\frac{DF+9}{2 \times f_{xtal}}$ |
| $t_{LOW}$ | SCLK LOW time | $\geq \frac{17}{f_{xtal}}$ | $\frac{DF-3}{2 \times f_{xtal}}$ ; note 2 |
| $t_{HIGH}$ | SCLK HIGH time | $\geq \frac{17}{f_{xtal}}$ | $\frac{DF+3}{2 \times f_{xtal}}$ ; note 2 |
| $t_{RC}$ | SCLK rise time | ≤ 1 µs | ≤ 1 µs; note 3 |
| $t_{FC}$ | SCLK fall time | ≤ 0.3 µs | ≤ 0.1 µs; note 4 |
| **SDA** | | | |
| $t_{BUF}$ | bus free time | $\geq \frac{14}{f_{xtal}}$ | ≥ 4.7 µs; note 5 |
| $t_{SU;DAT}$ | data set-up time | ≥ 250 ns | $\geq \frac{15}{f_{xtal}}$ ; note 6 |
| $t_{HD;DAT}$ | data hold time | ≥ 0 | $\geq \frac{9}{f_{xtal}}$ |
| $t_{RD}$ | SDA/P2.3 rise time | ≤ 1 µs | ≤ 1 µs; note 3 |
| $t_{FD}$ | SDA/P2.3 fall time | ≤ 0.3 µs | ≤ 0.1 µs; note 4 |
| $t_{SU;STO}$ | STOP condition set-up time | $\geq \frac{14}{f_{xtal}}$ | $\frac{DF-3}{2 \times f_{xtal}}$ |

**Notes**

1. DF stands for divisor of $f_{xtal}$ (see *"PCD33XXA family data sheet"*).
2. Values given for ASC = 0; for ASC =1: $t_{HIGH} = \frac{3(DF+1)}{4 \times f_{xtal}}$ ; $t_{LOW} = \frac{DF-3}{4 \times f_{xtal}}$.
3. Determined by I²C-bus capacitance ($C_b$) and external pull-up resistor.
4. At maximum allowed I²C-bus capacitance $C_b$ = 400 pF.
5. Determined by program.
6. If $t_{LOW} < \frac{24}{f_{xtal}}$, $t_{SU:DAT} \geq \frac{t_{LOW} - 9}{f_{xtal}}$, independent of ASC.

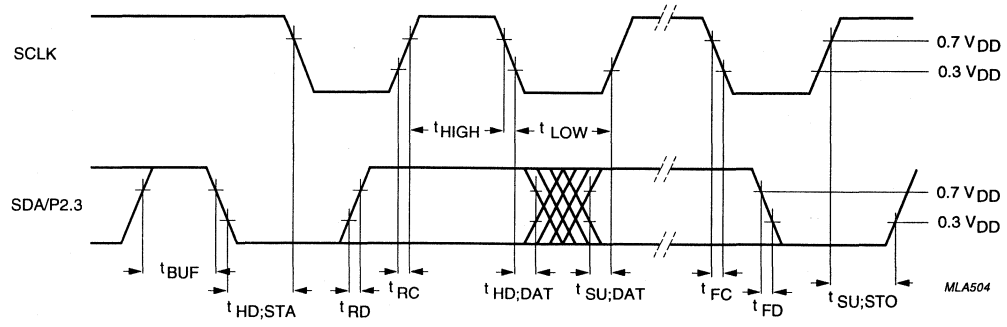



Fig.14 Slave SCLK and receiver SDA/P2.3 timing (SCLK and SDA/P2.3 are inputs).

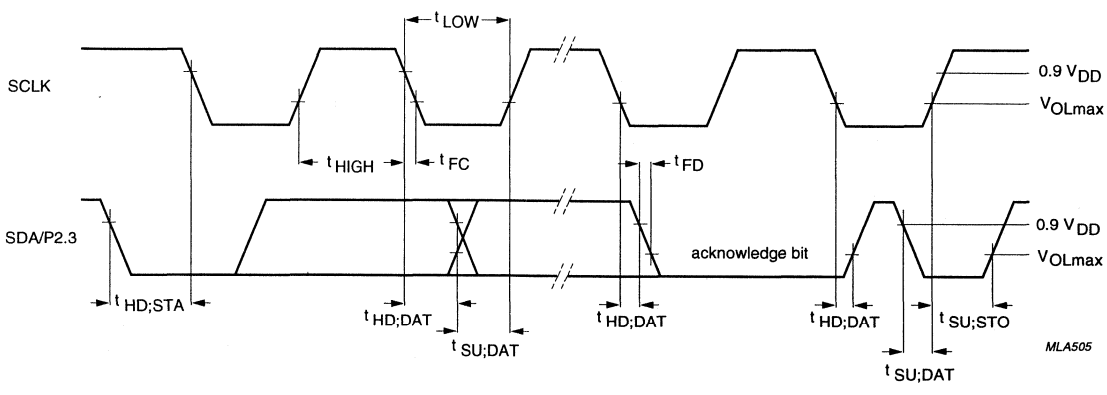
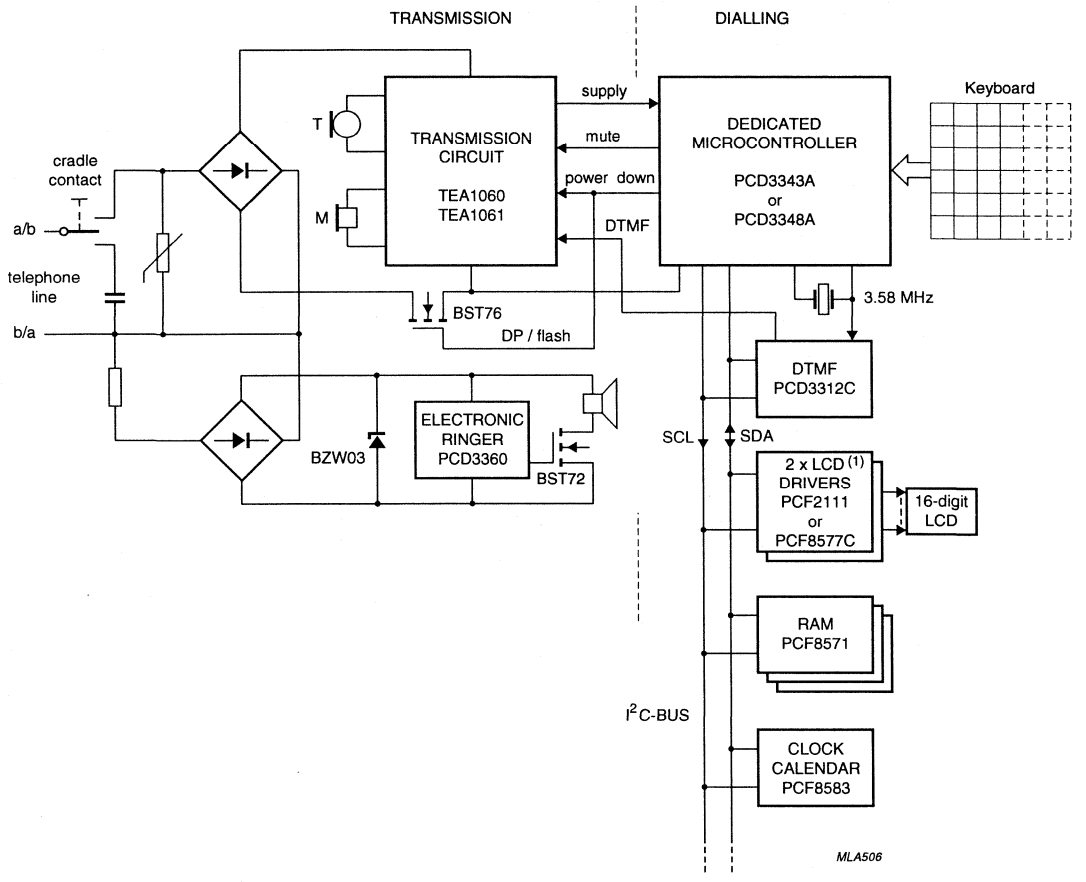



Fig.15 Master SCLK and transmitter SDA/P2.3 timing (SCLK and SDA/P2.3 are outputs).

## 12 APPLICATION INFORMATION

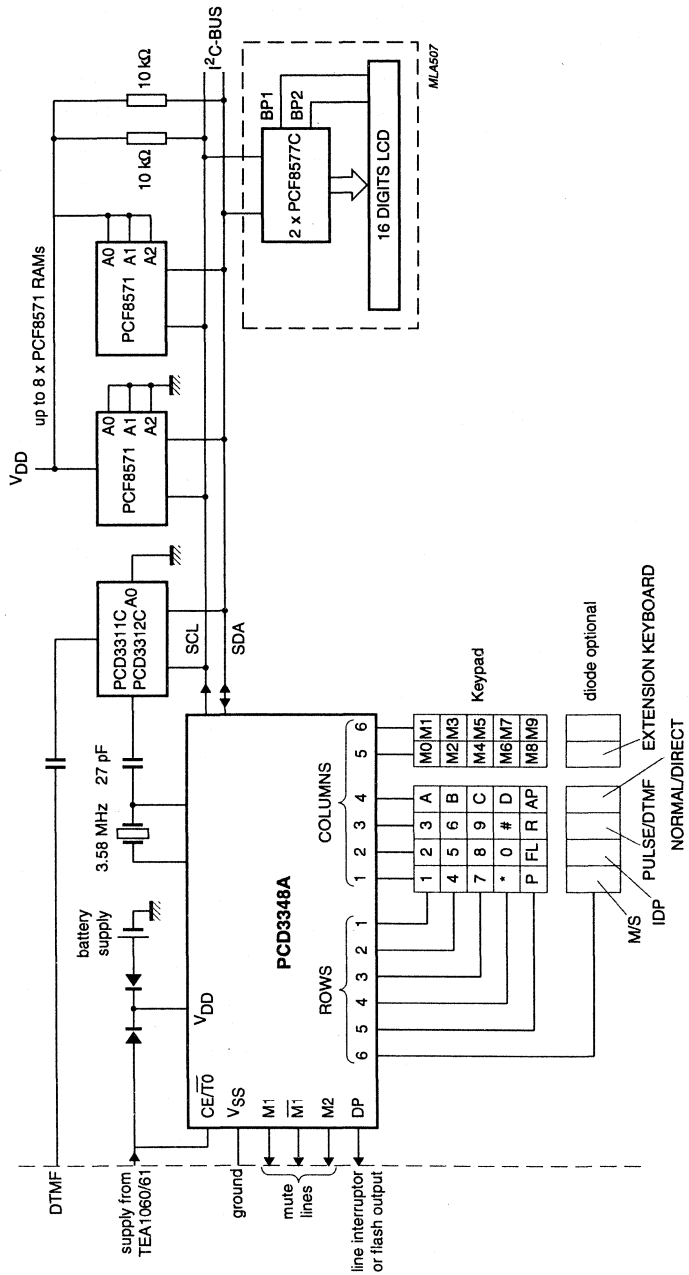
A block diagram of an electronic feature-phone built around the PCD3343A/48A is shown in Fig.16. It comprises the following dedicated telecom ICs:

- TEA1060/1061: transmission circuit for telephony
- PCD3312C: DTMF generator with I²C-bus interface
- PCF2111 or PCF8577C: 2 LCD drivers in LCD module MB7020160
- PCF8571: 1 kbyte RAMs with I²C-bus interfaces
- PCD3360/3361: programmable multi-tone ringer.




(1) Maximum 8.

Fig.16 Block diagram of an electronic feature-phone with common line interface.




A more detailed schematic of the PCD3343A/48A with: PCD3312 (DTMF), two PCF8571 (RAMs) and two PCF2111 (LCD drivers).
Row 5 of the keyboard contains the following special keys:
P: program and autodial
FL: flash or register recall
R: redial or extended redial
AP: access pause
Row 6 contains the different diode options. Columns 5 and 6 contain the keys M0 to M9 used as single name keys for repertory telephone numbers.

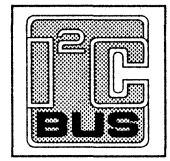
Fig.17 PCD3343A/PCD3348A application for an electronic feature-phone with associated keyboard.

# Single-chip 8-bit Telecom Microcontroller

# PCD3344A PCD3349A

**FOR DETAILED INFORMATION SEE THE LATEST ISSUE OF HANDBOOK IC03 OR DATASHEET**

## FEATURES




- 8-bit CPU, ROM, RAM, I/O in a single 28-lead package
- 2 k ROM bytes (PCD3344A)
- 4 k ROM bytes (PCD3349A)
- 224 bytes RAM
- Over 100 instructions (based on MAB8048) all of 1 or 2 cycles
- 20 quasi-bidirectional I/O port lines
- 8-bit programmable timer/event counter 1
- Two single-level vectored interrupts: external, 8-bit programmable timer/event counter 1
- Two test inputs, one of which also serves as the external interrupt input
- DTMF tone generator
- Reference for supply and temperature-independent tone output
- Filtering for low output distortion (CEPT CS203 compatible)
- Power-on reset
- Stop and idle modes
- Logic supply from 1.8 V to 6 V (DTMF tone output from 2.5 V)
- Low standby voltage of 1 V
- Low standby current of 2 μA (typ.)
- Clock frequency from 1 MHz to 16 MHz (3.58 MHz for DTMF suggested)
- Manufactured in silicon gate CMOS process.

## GENERAL DESCRIPTION

This data sheet details the specific properties of the PCD3344A and PCD3349A. The shared characteristics of the PCD33XXA family of microcontrollers are described in the PCD33XXA family data sheet, which should be read in conjunction with this publication.
The PCD3344A and PCD3349A are microcontrollers which have been designed primarily for Telecom applications. They include an on-chip dual tone multi-frequency (DTMF) generator. The PCD3344A and the PCD3349A provide 2 k and 4 k bytes respectively of program memory, 224 bytes of RAM and 20 I/O lines. The instruction set is based on that of the MAB8048 and is software compatible with the PCD33XXA family.

## LIMITING VALUES

In accordance with the Absolute Maximum System (IEC 134).

| SYMBOL | PARAMETER | MIN. | MAX. | UNIT |
|---|---|---|---|---|
| $V_{DD}$ | positive supply voltage | −0.5 | +7.0 | V |
| $V_I$ | all input voltages | −0.5 | $V_{DD}$+0.5 | V |
| $I_I$,$I_O$ | DC input or output current | −10 | +10 | mA |
| $I_{SS}$ | ground supply current | −50 | +50 | mA |
| $P_{tot}$ | total power dissipation | − | 125 | mW |
| $P_O$ | power dissipation per output | − | 30 | mW |
| $T_{stg}$ | storage temperature range | −55 | +150 | °C |
| $T_j$ | operating junction temperature | − | 90 | °C |

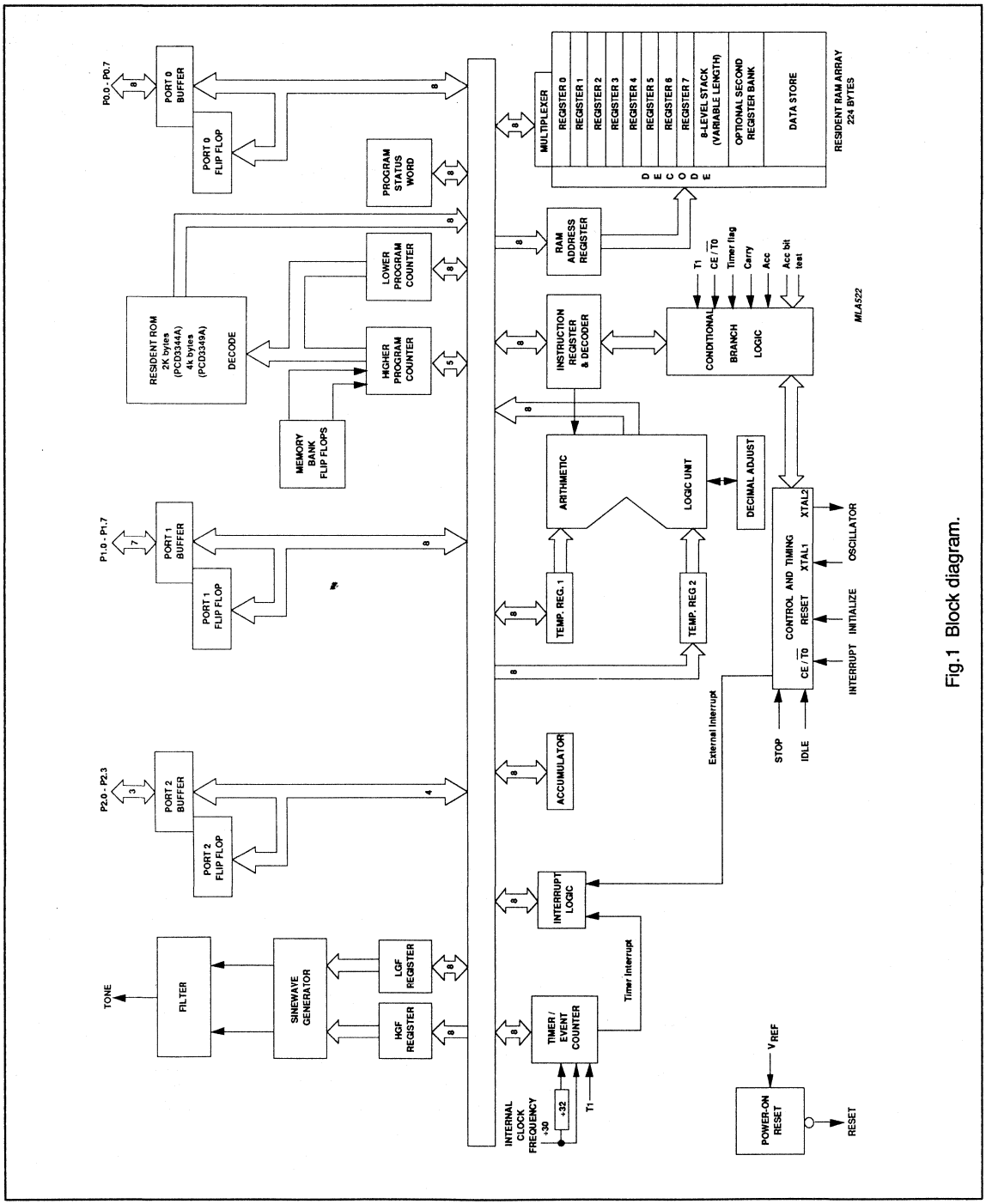



Fig.1 Block diagram.

## ORDERING INFORMATION

| EXTENDED TYPE NUMBER | PACKAGE | | | |
|---|---|---|---|---|
| | PINS | PIN POSITION | MATERIAL | CODE |
| PCD3344AP/49AP | 28 | DIL | plastic | SOT117 |
| PCD3344AT/49AT | 28 | mini-pack | plastic | SOT136A |

### Note to the Ordering Information

Full and up-to-date data for this device is available upon request via your Philips local sales office.

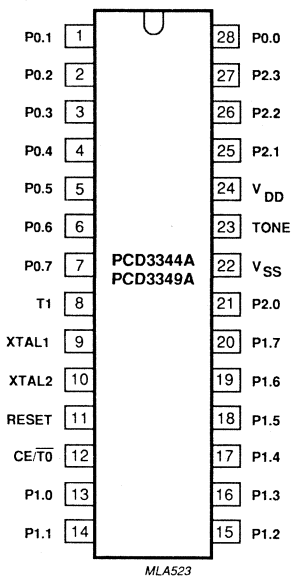



Fig.2 Pin configuration.

### PINNING

| SYMBOL | PIN | I/O | DESCRIPTION |
|---|---|---|---|
| P0.1-P0.7 | 1-7 | I/O | Port 0: quasi-bidirectional I/O lines |
| T1 | 8 | I | test 1/count input of 8-bit timer/event counter 1 |
| XTAL1 | 9 | I | crystal oscillator/ external clock input |
| XTAL2 | 10 | O | crystal oscillator output |
| RESET | 11 | I | reset input |
| CE/$\overline{T0}$ | 12 | I | chip enable / test 0 |
| P1.0-P1.7 | 13-20 | I/O | Port 1: quasi-bidirectional I/O line |
| P2.0 | 21 | I/O | Port 2: quasi-bidirectional I/O line |
| $V_{SS}$ | 22 | P | ground |
| TONE | 23 | O | DTMF output |
| $V_{DD}$ | 24 | P | positive supply voltage |
| P2.1-P2.3 | 25-27 | I/O | Port 2: quasi-bidirectional I/O lines |
| P0.0 | 28 | I/O | Port 0: quasi-bidirectional I/O line |

## PURCHASE OF PHILIPS I²C COMPONENTS

Purchase of Philips I²C components conveys a license under the Philips' I²C patent to use the components in the I²C system provided the system conforms to the I²C specification defined by Philips. This specification can be ordered using the code 9398 393 40011.

# SINGLE-CHIP 8-BIT MICROCONTROLLER

## DESCRIPTION

The PCD3346 is a single-chip 8-bit microcontroller manufactured in CMOS technology. The PCD3346 is a member of the PCD33XX family and as such has special on-chip features for telephony applications.

The PCD3346 has 20 quasi-bidirectional I/O port lines, a serial I/O interface, a single-level vectored interrupt circuit, two 8-bit timer event counters and on-board clock oscillator and clock circuits. The PCD3346 also incorporates 256 bytes of EEPROM permitting intermediate data storage without the need for battery back-up.

The instruction set is based on that of the MAB8048 and is instruction set compatible with the MAB8400 family. The PCD3346 has bit handling abilities for both binary and BCD arithmetic.

### Features

- 8-bit CPU, ROM, EEPROM, RAM, I/O in a single 28-lead DIL or SO package
- 4 K ROM bytes
- 128 RAM bytes
- 256 bytes EEPROM
- 20 quasi-bidirectional I/O port lines
- 2 x 8-bit programmable timers
- Two test inputs: one of which is also the external interrupt input (CE/$\overline{T0}$)
- Single-level vectored interrupts: external, timer/event counter, serial I/O and derivative port
- $I^2C$ hardware interface for serial data transfer on two lines (serial I/O data via an existing port line and clock via a dedicated line)
- Clock frequency 450 kHz to 10 MHz
- Over 80 instructions (based on MAB8048) all of 1 or 2 cycles
- Single supply voltage from 2.5 V to 6.0 V
- STOP and IDLE mode
- On-chip oscillator with output drive capability for peripherals (e.g. PCD3312 DTMF generator)
- Individual mask configuration of all port lines for: pull-up, push-pull or open drain
- Power-on-reset circuit and low supply voltage detection
- Individual mask selection of reset state for all ports
- Operating temperature range: −25 to + 70 °C

## PACKAGE OUTLINES

PCD3346P: 28-lead DIL; plastic (SOT117).
PCD3346T: 28-lead mini-pack; plastic (SO28; SOT136A).

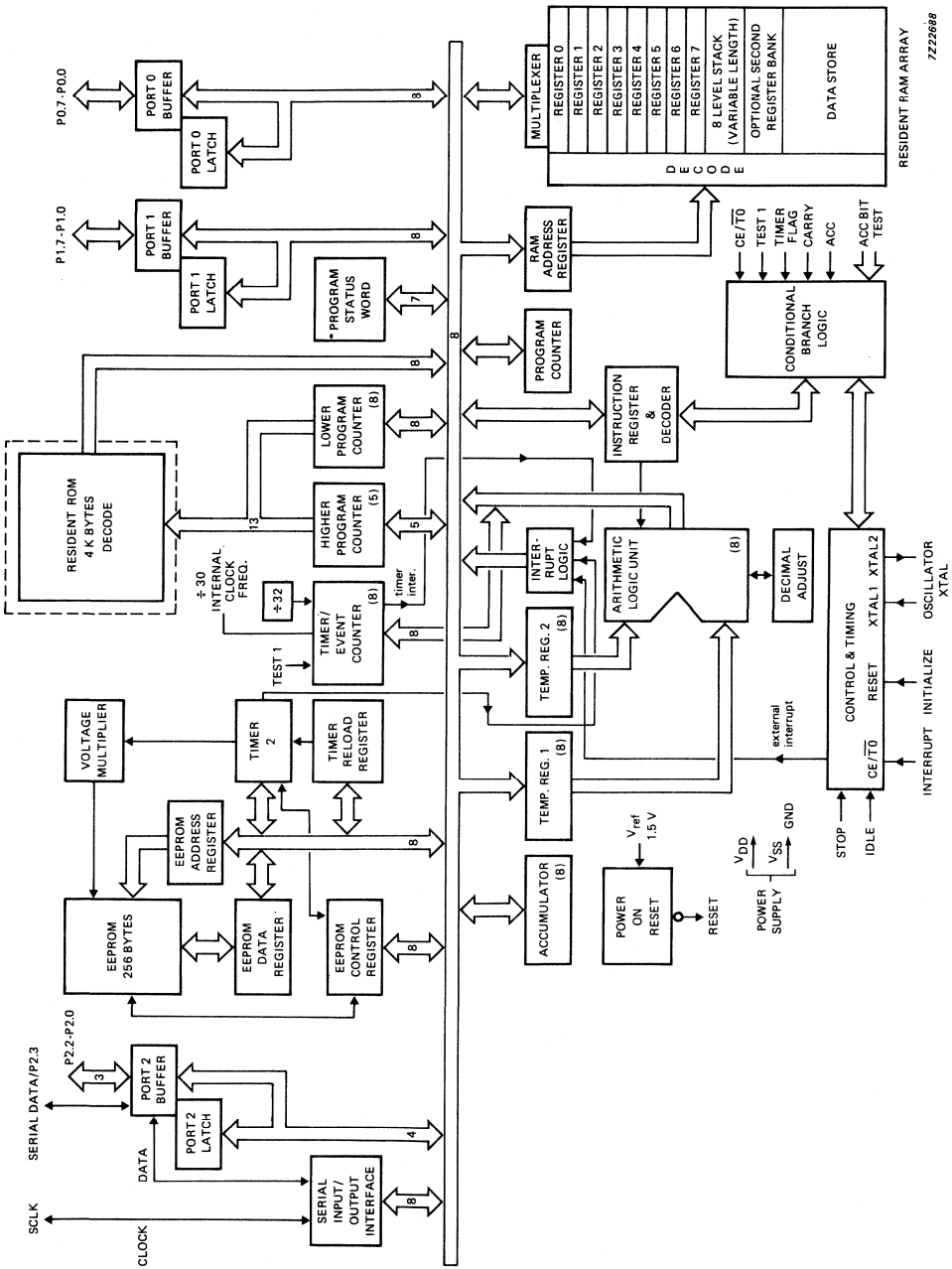



Fig. 1 Block diagram.

## PINNING

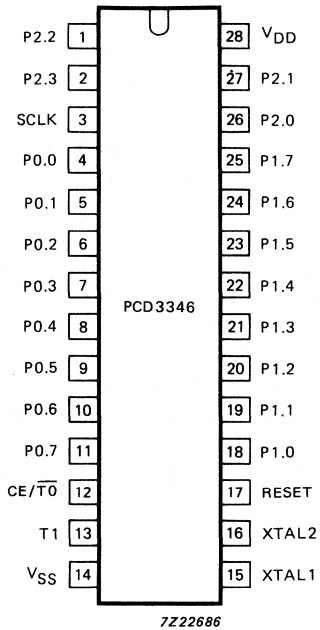



Fig. 2 Pinning diagram.

## PIN DESIGNATION

| Pin | symbol | type | function |
|---|---|---|---|
| 3 | SCLK | I/O | Clock: bidirectional clock for serial I/O. |
| 4-11 | P00-P07 | I/O | Port 0: 8-bit quasi-bidirectional I/O port. |
| 12 | CE/$\overline{T0}$ | I | Interrupt/Test 0: external interrupt input (sensitive to positive-going edge)/test input pin; when used as a test input directly tested by conditional branch instruction JT0 and JNT0. |
| 13 | T1 | I | Test 1: test input pin, directly tested by conditional branch instructions JT1 and JNT1. T1 also functions as an input to the 8-bit timer/event counter, using the STRT CNT instruction. |
| 14 | $V_{SS}$ | I | Ground: circuit earth potential. |
| 15 | XTAL1 | I | Crystal input: connection to timing component (crystal) which determines the internal oscillator frequency; also the input for an external clock source. |
| 16 | XTAL2 | I/O | Connection to other side of timing component. |
| 17 | RESET | I/O | Reset input: used to initialize the processor (active HIGH), or output of power-on-reset circuit. |
| 18-25 | P10-P17 | I/O | Port 1: 8-bit quasi-bidirectional I/O port. |
| 26,27, 1,2 | P20-P23 | I/O | Port 2: 4-bit quasi-bidirectional I/O port. P23 is the serial data input/output in serial I/O mode. |
| 28 | $V_{DD}$ | I | Power supply: 2.5 V to 6.0 V. |

## FUNCTIONAL DESCRIPTION

### Emulation of PCD3346

Emulation of the PCD3346 can be achieved using the piggyback version PCA3346B.

### Program memory

The program memory consists of 4 K bytes, in a read-only memory (ROM). Each location is directly addressable by the program counter. The memory is mask-programmed at the factory. Figure 3 shows the program memory map.

Four program memory locations are of special importance:

- Location 0; contains the first instruction to be executed after the processor is initialized (RESET),
- Location 3; contains the first byte of an external interrupt service subroutine,
- Location 5; contains the first byte of a serial I/O and derivative interrupt service subroutine,
- Location 7; contains the first byte of a timer/event counter interrupt service subroutine.

Program memory is arranged in banks of 2 K bytes, which are selected by SEL MB instructions. The program memory is further divided into location 'pages', each of 256 bytes. This latter division applies only for conditional branches. Memory bank boundaries can be crossed only by using the unconditional branch instructions after the appropriate memory bank has been selected. A CALL instruction can transfer control to a subroutine on any 'page'; RET and RETR instructions can transfer control from a subroutine back to the main program.

### Data memory

Data memory consists of 128 bytes, random-access data memory (RAM). Allocations are indirectly addressable using RAM pointer registers; up to 16 designated locations are directly addressable. Memory also includes an 8-level program counter stack addressed by a 3-bit stack pointer. Figure 4 shows the data memory map.

*Working registers*

Locations 0 to 7 are designated as working registers, directly addressable by the direct register instructions. Ease of addressing, and a minimum requirement of instruction bytes to manipulate their contents, makes these locations suitable for storing frequently addressed intermediate results. This bank of registers can be selected by the SEL RB0 instruction.

Executing the select register bank instruction SEL RB1, designates locations 24 to 31 as working registers, instead of locations 0 to 7, and these are then directly addressable. This second bank of working registers may be used as an extension of the first or reserved for use during interrupt service subroutines saving the first bank for use in the main program. If the second bank is not used, locations 24 to 31 may serve as general purpose RAM.

The first locations of each bank contain the RAM pointer registers R0, R1, R0' and R1', which indirectly address all RAM locations.

All RAM locations make efficient program loop counters when used with the decrement register and test instruction DJNZ.

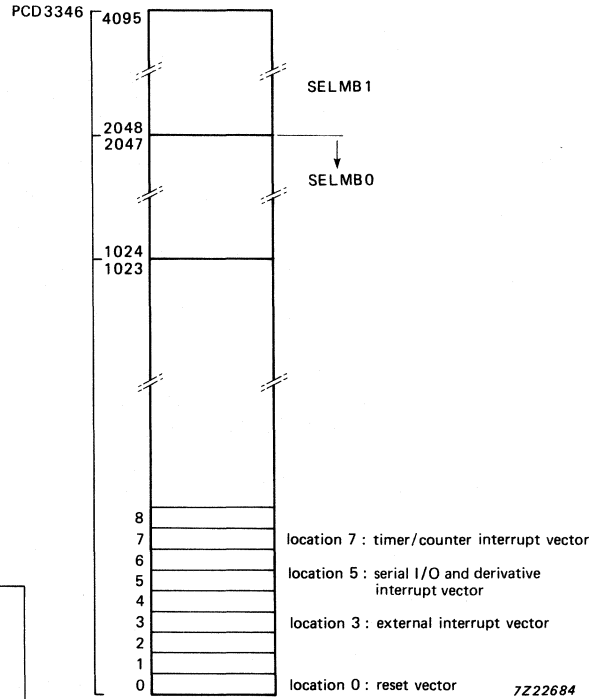



Fig. 3 Program memory map.

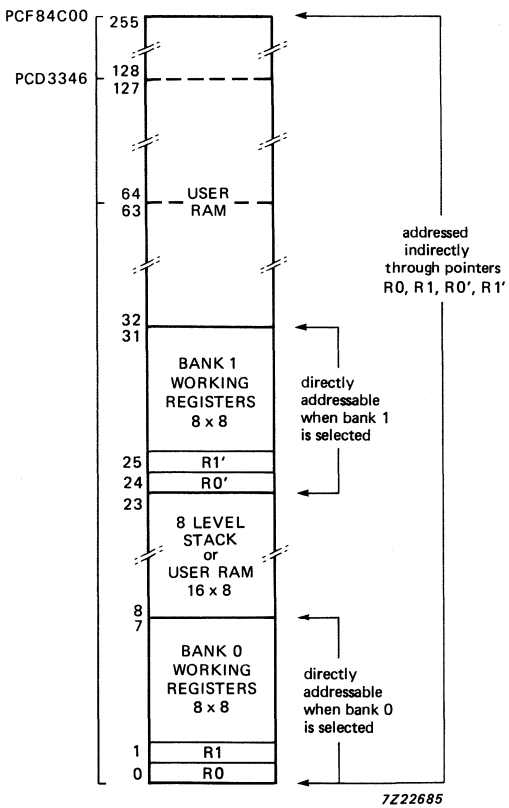



Fig. 4 Data memory map.

**FUNCTIONAL DESCRIPTION** (continued)

*Program counter stack*

Locations 8 to 23 may be designated as an 8-level program counter stack (2 locations per level), or as general purpose RAM. The program counter stack (Fig. 5) enables the microcontroller to keep track of the return addresses and status prior to interrupts or CALL instructions by storing the contents of the program counter before servicing the subroutine. A 3-bit stack pointer determines which of the eight register pairs of the program counter stack will be loaded with the next return address and status.

The stack pointer, when initialized to 000 by RESET, points to RAM locations 8 and 9. On the first subroutine CALL or interrupt, the contents of the program counter and bits 4, 6 and 7 of the program status word (PSW) are transferred to locations 8 and 9. The stack pointer increments by one and points to locations 10 and 11, ready for another CALL. Because an address may be up to 13 bits long, two bytes must be used to store each address.

At the end of a subroutine, which is signalled by a return instruction (RET or RETR), the stack pointer decrements by one and the contents of the register pair on top of the stack are transferred to the program counter. The saved PSW bits are transferred to the PSW only by the RETR instruction.

If all 8 levels of subroutine and interrupt nesting are not used, the unused portion of the stack may be used as any other indirectly addressable RAM locations.

Nesting of subroutines within subroutines can continue up to 8 times without overflowing the stack. If overflow does occur the deepest address stored (locations 8 and 9) will be overwritten and lost since the stack pointer overflows from 111 to 000. It also underflows from 000 to 111.

The value of the saved contents of the program counter during an interrupt CALL is not the same as the value saved during a normal CALL to subroutine. During an interrupt CALL, the program counter return address is saved; during a subroutine CALL, the saved program counter value is one less than the program counter return address.

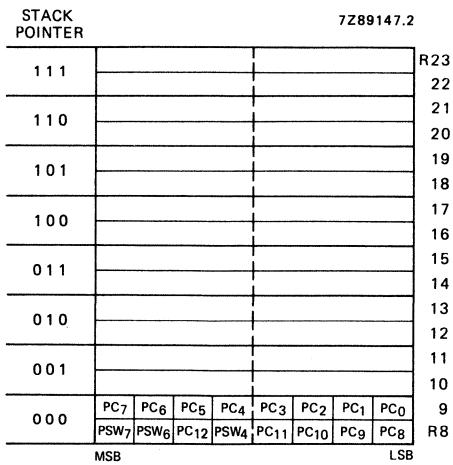



Fig. 5 Program counter stack.

### EEPROM memory

The PCD3346 has 256 bytes of Electrically Eraseable Programmable Read Only Memory on-chip. On-chip EEPROM makes register retention possible without battery back-up. The 256 bytes of EEPROM are arranged as 64 pages. Each page is a single row of 32 bits, i.e. 4 bytes. This organisation makes it possible to read and erase/write in single page and byte mode. Addressing is performed on-chip using an 8-bit address decoder; the upper six bits select the page and the lowest two a byte on that page. The EEPROM address register is auto-incremented within the page which simplifies the read/write operation. Auto-increment is active during read and write functions, but is inactive during erase mode. Overflow during auto-increment leaves the register at 00000000.

The EEPROM is interfaced with the CPU via the derivative registers. Table 1 shows the instructions that control the derivative registers.

**Table 1** Derivative register instruction set.

The PCD3346 has an additional 4 instructions to handle the derivative registers, these are:

| Mnemonic | description | opcode (hex) | function<br>x = 1 to 6 |
|---|---|---|---|
| MOV A,Dx | move derivative register contents to accumulator | 8C Dx | (A) <– (Dx) |
| MOV Dx,A | move accumulator contents to derivative register | 8D Dx | (Dx) <– (A) |
| ANL Dx,A | AND derivative register with accumulator | 8E Dx | (Dx) <– (Dx) AND (A) |
| ORL Dx,A | OR derivative register with accumulator | 8F Dx | (Dx) <– (Dx) OR (A) |

Each instruction is 2 cycles, 2 bytes. The second byte selects the derivative register to be manipulated (1 to 6) e.g. data transport from accumulator to derivative register D01 is performed by the two byte opcode 8 D 01.

For the PCD3346 six derivative registers are required and these are given in Table 2.

**Table 2** PCD3346 derivative registers

| name | derivative address | function |
|---|---|---|
| ADD1 | 01H* | contains the EEPROM array address |
| DATR | 03H | contains the read or write data |
| EPCR | 04H | EEPROM control register |
| RELR | 05H | timer T2 reload |
| T2 | 06H | contents of timer T2 |

* For larger EEPROMs a second address register (ADD2) is necessary to accommodate the higher part of the EEPROM address. To ensure uniformity of derivative registers, a derivative address (02H) should be reserved for the high address part.

**FUNCTIONAL DESCRIPTION** (continued)

**Table 3** EEPROM control register

| EPCR (04H) | STT2 | EIT2 | TF2 | EWP | MC3 | MC2 | MC1 | MC0 |
|---|---|---|---|---|---|---|---|---|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |

EPCR is a derivative R/W register containing the status of the EEPROM interface and timer T2.

**Abbreviations used**

| Bit | | Function |
|---|---|---|
| STT2 | start timer T2 | Setting this bit clears the pre-scaler and starts timer T2 in the general purpose mode. Timer T2 in the general purpose mode does not affect the EEPROM interface. T2 is loaded with the value in the RELR register at both start up and underflow. Setting STT2 to zero, stops Timer T2 and the pre-scaler, clears the pre-scaler and then reloads Timer T2. |
| EIT2 | enable interrupt | When this bit is set and the SIO enabled, a derivative interrupt request is generated when TF2 is set. No interrupt request will be generated if EIT2 is set to zero. |
| TF2 | timer flag T2 | This bit is set by hardware when Timer T2 underflows and a derivative interrupt is requested if the SI is enabled (Fig. 6). TF2 must be cleared by software. |
| EWP | erase/write cycle in progress | This bit is set by software at the beginning of an erase or write cycle in the byte mode. It must be set by software in the page mode. It is cleared by hardware when the erase or write cycle has expired. |
| MCn | mode control bits | Four bits which control the working mode of the EEPROM. The truth table is given in Table 4. |

**Table 4** Truth table for EEPROM interface control bits

| EWP | MC3 | MC2 | MC1 | MC0 | function |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | read byte |
| 1 | 0 | 1 | 1 | 0 | write byte |
| 1 | 1 | 0 | 0 | 1 | erase/write byte |
| 1 | 1 | 1 | 0 | 0 | erase page |
| 0/1* | 0 | 1 | 0 | 1 | write page |
| 1 | 1 | 0 | 1 | 0 | erase bulk |

Other bit patterns have no effect upon the interface. Bits MC3 to MC0 can be read and written by software, but are also cleared by hardware after each erase or write cycle has expired.

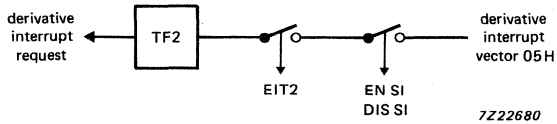



Fig. 6 Organization of a derivative interrupt.

**Timer T2**

Timer T2 determines the period of the erase and write timing cycles. Both times are the same, and are required to be of a fixed duration independent of the oscillator frequency. This requirement is fulfilled by T2 which is programmable timer with reload register RELR. Timer T2 is clocked via a 4-bit pre-scaler, which in turn is serviced by the processor clock ( $f_{osc}/30$).

This means that T2 is decremented every 480 oscillator periods.
Erase and write times are both of **10 ms duration. The value for the RELR is to be calculated according** to the following formula. Table 5 gives examples of the RELR values for different XTAL frequencies.

$$(RELR) = \frac{f_{XTAL}}{480} \cdot 10\ ms$$

RELR is a R/W derivative register. At the beginning of a read/write cycle, Timer 2 is loaded with the contents of RELR and the count started. Underflow of the timer signals that the erase cycle has finished and that Timer T2 is reloaded once more with the contents of RELR. The next underflow will signify the end of the write cycle and the counter will be switched off. Erase and write times are both of **10 ms duration.**

* EWP = 0 selects write page mode, EWP = 1 executes write page.

**FUNCTIONAL DESCRIPTION** (continued)

**Table 5** RELR values for various XTAL frequencies

| $f_{xtal}$ | timer 2 clocking (after 4-bit pre-scaler) | RELR value for 10 ms duration |
|---|---|---|
| 450 kHz | 1.1 ms | 10 |
| 1 MHz | 0.48 ms | 21 |
| 3.58 MHz | 0.134 ms | 75 |
| 10 MHz | 0.048 ms | 209 |

Timer T2 may also be used as a general purpose timer. It is started by setting the STT2 bit in register EPCR, which loads the contents of RELR into the timer. At overflow, the interrupt flag T2 is set and the timer auto-reloaded with RELR. Timer T2 can be read "on-the-fly" and is addressed as derivative register T2. The following instructions demonstrate control of the timer in the general purpose mode.

```
                        ; TF2 must be cleared
MOV A, #_timevalue      ; time to be down-counted
MOV RELR, A             ; load derivative register RELR
EN SI                   ; enable SI interrupt
MOV A, #STT2+EIT2       ; start counter and enable derivative interrupt
ORL EPCR, A             ; load status in derivative control register
```

When the timer underflows, an interrupt call to ROM address 5 will be generated (SI interrupt) and appropriate actions will be decided by software. The content of timer can only be read "on-the-fly" using the instruction MOV A, D6.

**Read EEPROM**

Reading data from the EEPROM is possible only when no erase/write action is in progress. The read operation of a data byte is performed as follows:

- load EEPROM address
- load EPCR (not necessary when the MCn bits are already set to read mode)
- read data register

A program example could be:

```
MOV A, #_address        ; load EEPROM read address
MOV ADD1, A             ; send address to ADD1 register
MOV A, DATR             ; load data from EEPROM to accumulator
MOV Rm, A               ; store data
```

After executing the third instruction, the ADD1 address register is auto-incremented and the next byte can be read by repeating the last two instructions.

### The erase/write operation

There are five erase/write modes controlled by the MCn bits in register EPCR:

- write byte
- erase/write byte
- erase page
- write page
- erase bulk

Each write or erase action takes 10 ms, this value must be loaded into the RELR register before starting any write/erase action, e.g.:

```
MOV A, #_RELR value     ; load appropriate 10 ms value according to fXTAL
                        ; frequency into accumulator
MOV RELR, A             ; load RELR register
```

A new erase or write operation can be started if no previous erase or write operation is in progress.

### Write byte

A write operation may be started only if the addressed byte has been erased. The write operation is as follows:

- load EEPROM address
- load data
- load EPCR

The last instruction sets the operation mode and starts Timer T2, the address register is incremented and an interrupt is generated when the 10 ms count is reached.

### Erase/write byte

The erase/write operation clears the addressed byte and writes the new. The erase/write operation is as follows:

- load EEPROM address
- load data
- load EPCR

The last instruction sets the operation mode and starts Timer T2, the address register is incremented and an interrupt is generated when the 2 x 10 ms count is reached.

### Erase page

In this mode a complete four byte page will be erased. This operation requires only that the first byte of the page be addressed, after which the entire page will be erased. The erase page operation is as follows:

- load EEPROM with address of first byte of any page
- load EPCR

The last instruction starts Timer T2 and an interrupt is generated after the 10 ms period has elapsed.

**FUNCTIONAL DESCRIPTION** (continued)

* Piggybacks and SDS probes only (it is recommended to do it always, also for ROM code).

If the EPCR-register is checked before any erase/write operation, a dummy read of the EEPROM data register is neccessary for technical reasons. Please note that the EEPROM address register ADD1 will be autoincremented by one due to the dummy read.

Example.

```
LABEL1      MOV A, D4                    ; EPCR check
            test EWP- bit, jump to LABEL1 ; busy wait
            MOV A, D3                    ; dummy read
            .
            .
            MOV A, value                 ; choose e/w operation
            MOV D4, A                    ; start e/w operation
```

**Write page**

A write page operation can only be started if the addressed page is empty (erased). If the page is not empty an erase page must be executed beforehand. The write page operation is as follows:

- load EPCR (select write page mode)
- load ADD1 with address of first byte of any page
- load four data bytes (EEPROM address register auto-increments after each load except after fourth byte*)
- load EPCR (start write page)

The last instruction starts Timer T2 and an interrupt is generated after the 10 ms period has elapsed.

Given that an erase page cycle has been executed, a write page program is as follows:

```
MOV A, #MC2+MC0          ;
MOV EPCR, A              ; select write page mode
MOV A, #_address         ; EEPROM start address to accumulator
MOV ADD1, A              ; address to derivative address register

MOV A, R1                ; byte to be written to accumulator
MOV DATR, A              ; write 1st byte, address register auto-incremented
MOV A, R2                ; byte to be written to accumulator
MOV DATR, A              ; write 2nd byte, address register auto-incremented
MOV A, R3                ; byte to be written to accumulator
MOV DATR, A              ; write 3rd byte, address register auto-incremented
MOV A, R4                ; byte to be written to accumulator
MOV DATR, A              ; write 4th byte

MOV A, #EWP+MC2+MC0      ; set write page mode and execute bit in accumulator
MOV EPCR,A               ; start cycle
```

The last instruction sets the operation mode and starts Timer T2, the address register is incremented and an interrupt is generated at Timer T2 underflow.

* Note that data cannot be addressed (read or write) over a page limit.

**Bulk erase**

In the bulk erase mode the complete EEPROM is erased. The bulk erase operation is as follows:

– Load EPCR.

After start of bulk erase Timer T2 is started and an interrupt is generated at underflow. Below is an example of a program for a bulk erase:

```
MOV A,#EWP+MC3+MC1      ; set the bulk erase mode in accumulator
MOV EPCR,A              ; load EPCR and start erase cycle
```

Timer T2 is started and an interrupt is generated at Timer T2 underflow.

**FUNCTIONAL DESCRIPTION** (continued)

**IDLE and STOP modes**

*IDLE mode*

When the microcontroller enters the IDLE mode via the IDLE instruction (01H), the oscillator, timer/counter and serial I/O are kept running. The microcontroller exits from the IDLE mode via a RESET or one of three interrupts if they are enabled. If the interrupt is not enabled, the microcontroller will remain in the IDLE mode. An active signal on the RESET pin restarts the microcontroller and a normal RESET sequence is executed (see Fig. 7).

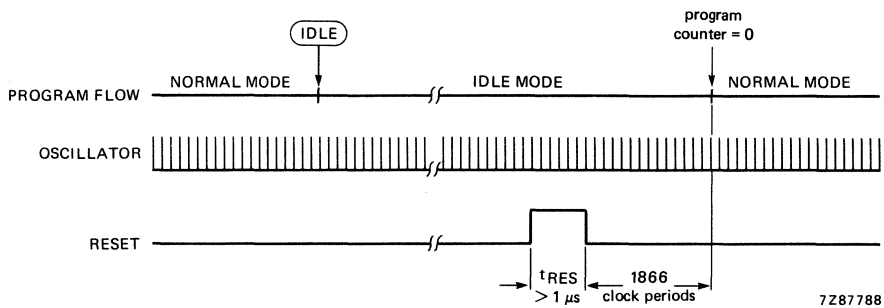



Fig. 7 Exit from IDLE mode via a RESET.

Any erase/write cycle in progress during IDLE mode will be completed. The interrupt request generated at the end of the erase/write cycle, if enabled, wakes up the microcontroller returning it to the normal mode.

An active signal coming from an enabled interrupt causes the execution of the normal interrupt routine since normal interrupt scanning is still being carried out. A LOW to HIGH transition on the external interrupt pin (CE/$\overline{TO}$) reactivates the microcontroller. A HIGH level applied to CE/$\overline{TO}$ will reactivate the microcontroller only in the STOP mode. Thus if CE/$\overline{TO}$ was HIGH before the microcontroller entered the IDLE mode, it must go LOW before the microcontroller can be reactivated (see fig. 8).

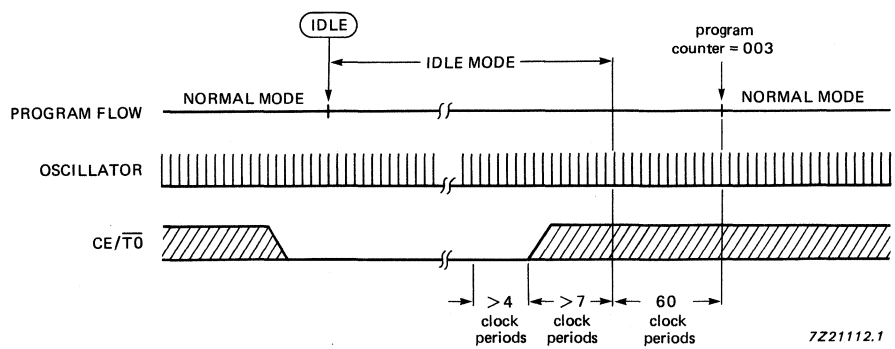



Fig. 8 Exit from IDLE mode via an interrupt.

Wake-up from the IDLE mode is ensured when CE/$\overline{T0}$ is LOW for at least 4 CP (clock periods) and then HIGH for 7 CP. After the initial forced CALL 003H operation (60 CP), the program continues with the external interrupt service routine. During the STOP mode, the address of the instruction immediately following the instruction that caused the microcontroller to enter the IDLE mode is present on the address bus.

*STOP mode*

**Before entering the STOP mode bit EWP of the EPCR register must be set to logic 0.**

The microcontroller enters the STOP mode via the STOP instruction (22H). The oscillator is switched off and the internal status of the CPU, RAM contents and the state of I/O ports are not affected. The microcontroller can be brought out of the STOP mode by an active signal at the external interrupt input or by an external RESET signal. When one of these two signals is applied, an internal delay of 1866 CP is provided to ensure that all internal clocks are operating correctly before restart (see Fig. 9).

Note: the start-up time of a crystal oscillator is measured in milliseconds, and the 1866 CP count begins after this start-up time.

If the microcontroller exits from the STOP mode by activating RESET, a normal RESET sequence is executed.

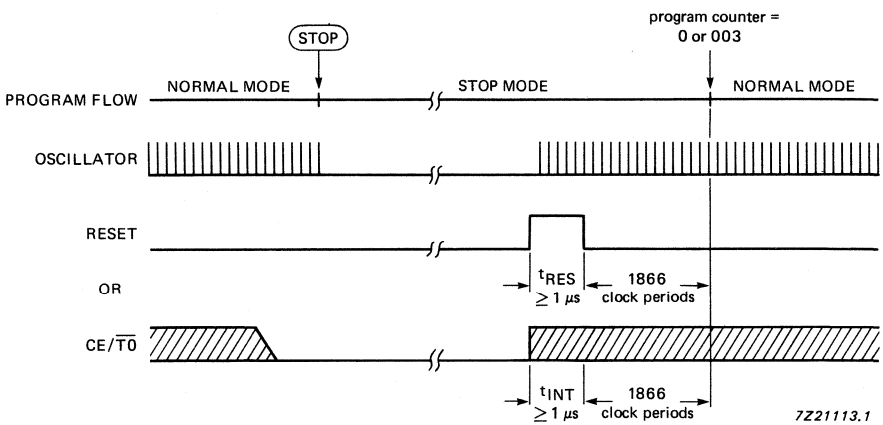



Fig. 9 Entering and exiting the STOP mode.

If the microcontroller exits the STOP mode by pulling the external interrupt input pin HIGH, an interrupt sequence is executed only if the external interrupt is enabled. In this event the microcontroller resumes the normal program sequence after returning from the interrupt routine, as in the normal mode. If the interrupt is not enabled, it continues the normal program sequence, executing the instruction following the STOP instruction.

The microcontroller is restarted by a HIGH level applied at the CE/$\overline{T0}$ pin.
Note: When exiting the STOP mode via an interrupt, a further instruction in the main program series is executed prior to entering the interrupt routine.

If the CE/$\overline{T0}$ level is active during the STOP instruction then no STOP is executed.

A HIGH level on the external interrupt input of at least 1 µs will cause the microcontroller to exit the STOP mode. During the STOP mode, the address of the instruction immediately following the last STOP instruction is present on the address bus.

**FUNCTIONAL DESCRIPTION** (continued)

**I/O facilities**

The PCD3346 has 23 I/O lines arranged as:

- Port 0 8-bit parallel port (P0.0 to P0.7)
- Port 1 8-bit parallel port (P1.0 to P1.7)
- Port 2 4-bit parallel port (P2.0 to P2.3)
- SCLK serial I/O clock line
- CE/$\overline{T0}$ external interrupt and test input. When used as a test input can be directly tested by conditional branch instructions JT0 and JNT0
- T1 test input which can alter program sequences when tested by conditional jump instructions JT1 and JNT1. T1 also functions as an input to the 8-bit timer/event counter.

*Parallel ports*

Parallel port lines can be individually configured as outputs or inputs; their structure is quasi-bidirectional. Output data written to a port is latched and remains unchanged until rewritten. Input data is not latched and must be present until read by an input instruction.

Input lines are fully CMOS compatible; output lines can drive one TTL or CMOS load.

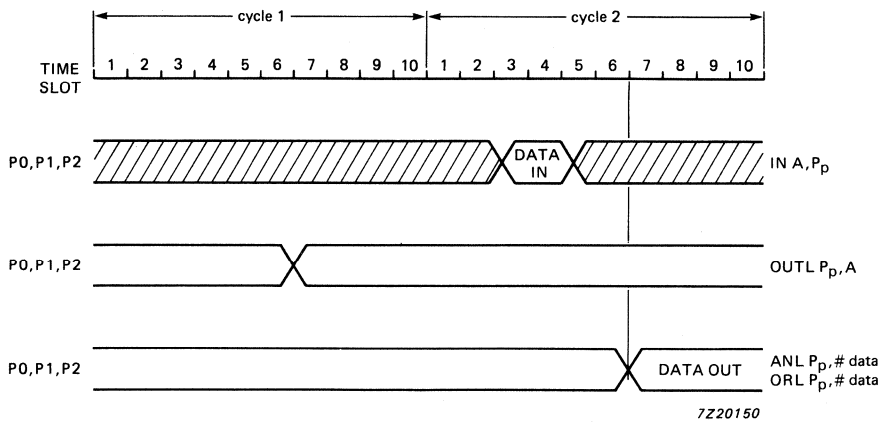



Fig. 10 Timing diagram for all ports using IN, OUTL, ANL and ORL instructions.

Figure 10 shows the timing diagram for all ports using IN, OUTL, ANL and ORL instructions. For the OUTL instruction data changes on time slot 7 of cycle 1. For the ANL and ORL instructions, the ports change on time slot 7 of cycle 2.

Figure 11 shows the quasi-bidirectional I/O interface with push-pull output and switched pull-up current source.

Each line is pulled up to $V_{DD}$ via a constant current source (TR4), which is enabled via TR3 whenever one of the two output latches contains a logic 1. This current source is sufficient for a TTL HIGH level, yet can be pulled LOW by an external CMOS device, thus allowing the same pin to be used for both input and output.

When a logic 1 is written to a port line for the first time (MQ = 1, SQ = 0), TR2 is switched on for the duration of the internal write pulse (one oscillator period), to provide a fast transition from logic 0 to logic 1. Subsequent writing of a logic 1 to a port line will not switch TR2 on. This ensures that the port lines can be pulled LOW when configured as inputs.

When a logic 0 is written to a port line, TR3 switches off the current source. Current sinking capability is provided by TR1, which is now switched on. When used as an input, a logic 1 must first be written to the port line; otherwise TR1 will remain low impedance.

PCD3346 mask options make it possible for each individual port line to be configured as one of the following:

1. STANDARD PORT: quasi-bidirectional I/O with switched pull-up current source of 100 $\mu$A (typ.) and P-channel booster transistor TR2. TR2 is only active during 1 clock cycle (Fig. 11).
2. OPEN DRAIN: quasi-bidirectional I/O with only an N-channel open drain output. Application as an output requires connection of an external pull-up resistor (Fig. 12). When an open drain port is unused it must be connected to $V_{SS}$.
3. PUSH-PULL OUTPUT: the outputs can sink or source 1,6 mA (min.) at $V_{DD}$ = 5 V. These pins may not be used as inputs (Fig. 13). The contents of the port latch may be read using the IN A,pp instruction.

Individual mask selection of the pin's state following a RESET can be achieved by appending option S or R to options 1, 2 or 3.

Option S-SET; following RESET the pin will be set HIGH.

Option R-RESET; following RESET the pin will be set LOW.

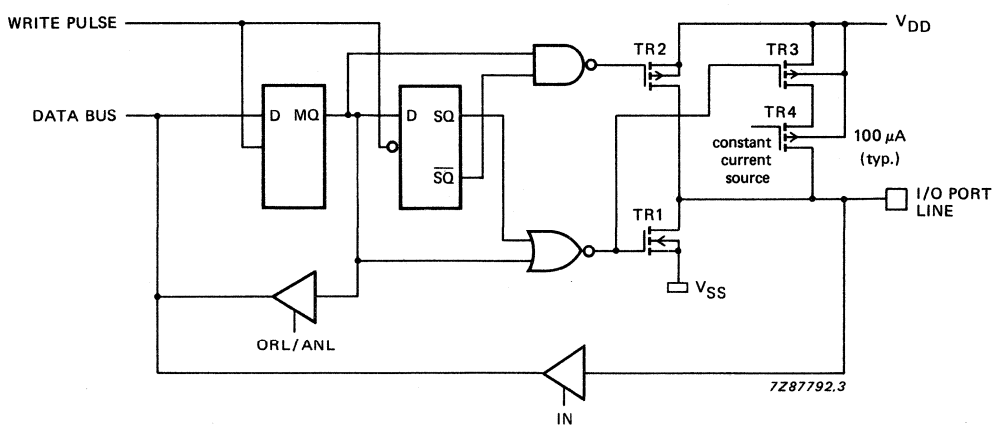



Fig. 11 Standard output with switched pull-up current source.

**FUNCTIONAL DESCRIPTION** (continued)

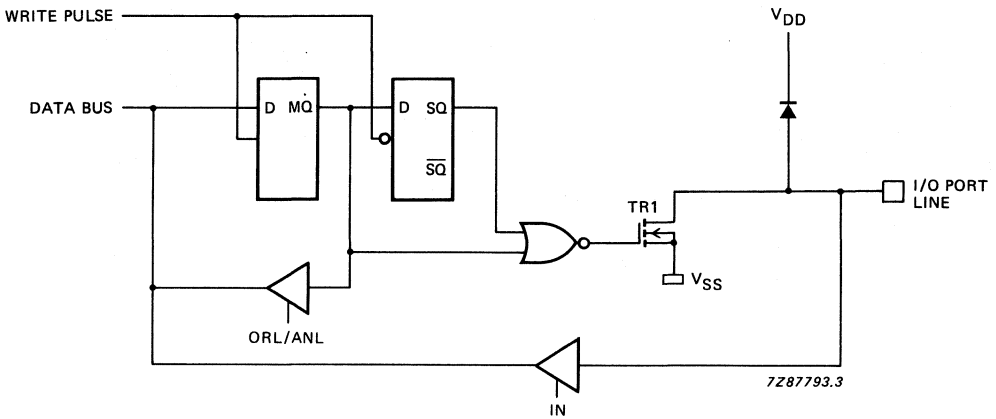



Fig. 12 Open drain output.

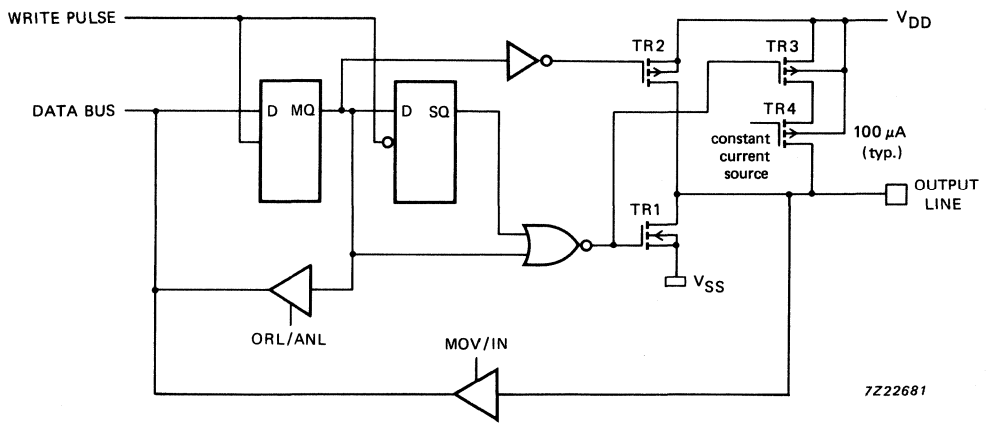



Fig. 13 Push-pull output.

*Serial I/O (SIO)*

The PCD3346 has an on-chip serial I/O interface tailored for $I^2C$-bus communications. The SIO interface is a versatile feature in intelligent telephone circuitry.

Where a normal microcontroller must regularly monitor the serial data bus for the presence of data, this serial I/O interface detects, receives and converts the serial data stream into parallel format without interrupting the execution of the current program. An interrupt is sent to the CPU only when a complete byte is received. It then reads the data byte in one instruction.

During transmission the serial I/O interface performs parallel to serial conversion and subsequent serial output of the data. The microcontroller is only interrupted in the execution of its programmed tasks when a complete byte has been transmitted.

The design of the PCD3346 serial I/O system allows any number of devices from the PCF8500 family (clips) to be connected via the two-line serial bus. The ability of any devices to communicate, without interrupting the operation of any other devices on the bus, is an outstanding attribute of the system. This is achieved by allocating a specific 7-bit address to each device and providing a system whereby a device reacts only to a message prefixed with its own address or the 'general CALL' address. Address recognition is performed by the interface hardware so that operation of the microcontroller need only be interrupted when a valid address has been received. This saves significant processing time and memory space compared with a conventional microcontroller employing a software serial interface. When the addressing facility is not required, for instance in a system with only two microcontrollers, direct data transfer without addressing can be performed. In multi-master systems, an automatically invoked arbitration procedure prevents two or more devices from continuing simultaneous transmission.

In NORMAL (running) and IDLE mode, the serial I/O logic remains active; its internal system clock will be switched off when there is no activity on the serial bus.

After execution of the STOP instruction, the oscillator of the PCD3346 is switched off. This means that the serial I/O logic will remain in the state it was in when the STOP mode was entered. To avoid "bus block" problems and to assure correct start-up of the bus after exit from the STOP mode, the user should disable the serial logic (ESO = 0) prior to the execution of the STOP instruction.

After a reset, the first 30 clock pulses of the 1866 pulse initiation phase reset the serial I/O interface and set P2.3/SDA and SCLK HIGH.

*Serial I/O interface*

Figure 14 shows the serial I/O interface. The clock line of the serial bus has exclusive use of the SCLK pin while the data line shares the SERIAL DATA pin with I/O line P2.3. When the serial I/O is enabled, P2.3 is disabled as a parallel port line; (P2.3 and SCLK are open drain; when unused these lines must be connected to $V_{SS}$).

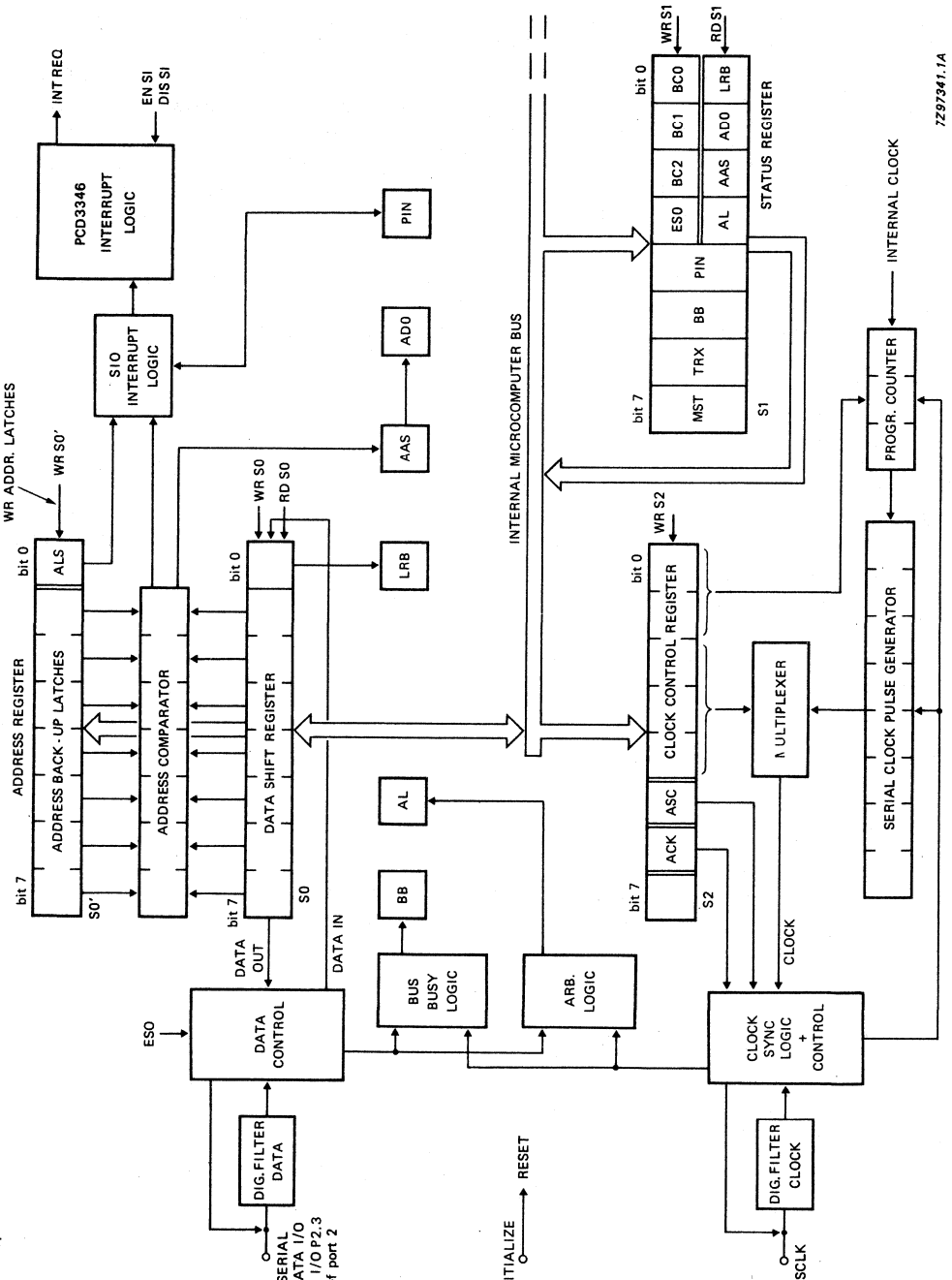



Fig. 14 Serial I/O interface.

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

- Data shift register (S0)
- Serial I/O interface status word (S1)
- Serial clock control word (S2)
- Address register (SO')

*Data shift register (S0)*

Register S0 converts serial data to parallel format and vice versa. An interrupt is generated only after a complete byte has been transmitted, or after a complete data byte, specific address or 'general CALL' address has been received. The most significant bit is transmitted first.

*Serial I/O interface status word (S1)*

Register S1 provides information concerning the state of the interface and stores information from the microcontroller. Bits 0 to 3 are duplicated: control bits in these positions can only be written by the microcontroller, while status bits can only be read.

MST and TRX (see Table 6)

These bits determine the operating mode of the serial I/O interface.

**Table 6** Operating modes of the serial I/O interface

| MST | TRX | operating mode |
|---|---|---|
| 0 | 0 | slave receiver |
| 1 | 0 | master receiver |
| 0 | 1 | slave transmitter |
| 1 | 1 | master transmitter |

BB: Bus Busy.

This flag indicates the status of the bus.

PIN: Pending Interrupt NOT

PIN = '0' indicates the presence of a pending interrupt, which will cause a Serial Interrupt Request when the serial interrupt mechanism is enabled.

ESO: Enable Serial output

The ESO flag enables/disables the serial I/O interface: ESO = '1' enables, ESO = '0' disables. ESO can only be written by software.

BC0, BC1 and BC2

Bits BC0, BC1 and BC2 indicate the number of bits received or transmitted in a data stream. These bits can only be written by software.

AL: Arbitration Lost

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

**FUNCTIONAL DESCRIPTION** (continued)

AAS: Addressed As Slave

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

AD0: Address Zero

This flag is set by hardware after detection of the 'general CALL' address when the interface is operating in the address recognition mode.

LRB: Last Received Bit

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

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

*Serial clock control word (S2)*

Bits 0 to 4 of the clock register S2 determine the frequency of the serial clock signal. When a 6 MHz crystal is used, the frequency of the serial clock can be varied between 1 kHz and 154 kHz (see Table 7). An asymmetrical clock with a HIGH-to-LOW ratio of 3 : 1 can be generated using bit 5. The asymmetrical clock allows a microcontroller more time per clock period for sampling the data line, making the timing of this action less critical. Bit 6 can be used to activate the acknowledge mode of the serial I/O. S2 is a write only register.

*Address register (S0)*

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

*Serial I/O interrupt logic*

An EN SI instruction enables and a DIS SI instruction disables the interrupt logic. When enabled, a pending interrupt results in a serial I/O interrupt to the processor, causing a CALL to location 5 in the ROM. When disabled, the presence of an interrupt request is still indicated by PIN in S1, but the interrupt will not be serviced.

**Table 7** SIO clock pulse frequency control when using a 3.58 MHz, 6 MHz or 10 MHz crystal

| hexadecimal S20-S24 code | divisor | $f_{XTAL}$ (3.58 MHz) $f_{SCLK}$ (kHz) ** | $f_{XTAL}$ (6 MHz) $f_{SCLK}$ (kHz) ** | $f_{XTAL}$ (10 MHz) $f_{SCLK}$ (kHz) ** |
|---|---|---|---|---|
| 0 | not allowed | | | |
| 1 | 39 | 92 | *154 | *256 |
| 2 | 45 | 80 | *133 | *222 |
| 3 | 51 | 70 | *118 | *196 |
| 4 | 63 | 57 | 95 | *159 |
| 5 | 75 | 48 | 80 | *133 |
| 6 | 87 | 41 | 69 | *115 |
| 7 | 99 | 36 | 61 | *101 |
| 8 | 123 | 29 | 49 | 81 |
| 9 | 147 | 24 | 41 | 68 |
| A | 171 | 21 | 35 | 58 |
| B | 195 | 18 | 31 | 51 |
| C | 243 | 15 | 25 | 41 |
| D | 291 | 12 | 21 | 34 |
| E | 339 | 11 | 18 | 29 |
| F | 387 | 9.2 | 16 | 26 |
| 10 | 483 | 7.4 | 12 | 21 |
| 11 | 579 | 6.2 | 10 | 17 |
| 12 | 675 | 5.3 | 8.9 | 15 |
| 13 | 771 | 4.6 | 7.8 | 13.4 |
| 14 | 963 | 3.7 | 6.2 | 10.4 |
| 15 | 1155 | 3.1 | 5.2 | 8.7 |
| 16 | 1347 | 2.7 | 4.5 | 7.4 |
| 17 | 1539 | 2.3 | 3.9 | 6.5 |
| 18 | 1923 | 1.9 | 3.1 | 5.2 |
| 19 | 2307 | 1.6 | 2.6 | 4.3 |
| 1A | 2691 | 1.3 | 2.2 | 3.7 |
| 1B | 3075 | 1.2 | 2.0 | 3.3 |
| 1C | 3843 | 0.93 | 1.6 | 2.6 |
| 1D | 4611 | 0.78 | 1.3 | 2.2 |
| 1E | 5379 | 0.67 | 1.1 | 1.9 |
| 1F | 6147 | 0.58 | 1.0 | 1.6 |

* Not permitted for $I^2C$ operation.

** Maximum clock frequency for the $I^2C$-bus is 100 kHz.

**FUNCTIONAL DESCRIPTION** (continued)

### Interrupts

When an interrupt routine is entered, the contents of the program counter and bits 4, 6 and 7 of the PSW are saved in the program counter stack. The contents of the accumulator can only be saved by user software. Interrupt acknowledgement may be carried out by software via I/O ports. All interrupt routines must reside in memory bank 0; the SEL MB instructions may not be used within an interrupt routine. An interrupt routine can only be terminated by the RETR (return and restore) instruction. During an interrupt routine, subroutine calls must be terminated by the RET instruction. Using the RETR instruction to terminate a subroutine called in an interrupt routine would terminate the interrupt routine prematurely and result in a wrong return address.

### External Interrupts

When the external interrupt is enabled and no interrupt routine is in progress, a LOW-to-HIGH transition on the CE/$\overline{T0}$ pin sets the External Interrupt Flag (EIF) and initiates the external interrupt routine by forcing a CALL to program memory location 3.* The program counter points to the external interrupt vector address (003 H) between 2,6 and 3,6 machine cycles after the transition occurs. Interrupt latency depends on the instruction that is being executed when the transition occurs. If an interrupt routine is already in progress, an external interrupt request is stored in the External Interrupt Flag (EIF). When the external interrupt has been disabled, the request is still latched into the digital filter. Execution of a DIS I instruction cancels stored interrupt requests by clearing both the digital filter latch and the external interrupt flag.

An additional external interrupt can be created by enabling the timer/counter interrupt and loading FFH into the counter (one less than overflow). If the event counter mode is enabled by executing the STRT CNT instruction, a LOW-to-HIGH transition on the T1 input will then initiate an interrupt routine by forcing a call to the timer/counter interrupt vector (location 7).

### SIO Interrupt

An interrupt request from the SIO hardware will set the PIN flag to its active LOW state. This action is independent of the Enable SIO interrupt flag. When the SIO interrupt is enabled, the active LOW PIN flag will invoke the SIO interrupt routine by forcing a CALL to program memory location 5. After the SIO interrupt is initiated, the PIN flag is not automatically reset HIGH. PIN must be cleared by software during the interrupt routine.

### Timer/Counter Interrupt

When no interrupt is in progress and the timer/counter is enabled, a timer/counter overflow sets the Timer Interrupt Flag (TIF). This initiates the timer interrupt routine by forcing a CALL to program memory location 7.** If an interrupt routine is in progress, the interrupt request is stored in the timer interrrupt flag only if the timer interrupt has been enabled. Execution of a DIS TCNTI instruction cancels a previously stored interrupt request. The timer flag (TF) is set every time the timer/counter overflows and is not automatically reset when the timer/counter interrupt routine is called. It can only be cleared by the JTF and JNTF instructions or by a hardware RESET.

* This CALL clears the EIF flag.
** This CALL clears the TIF flag.

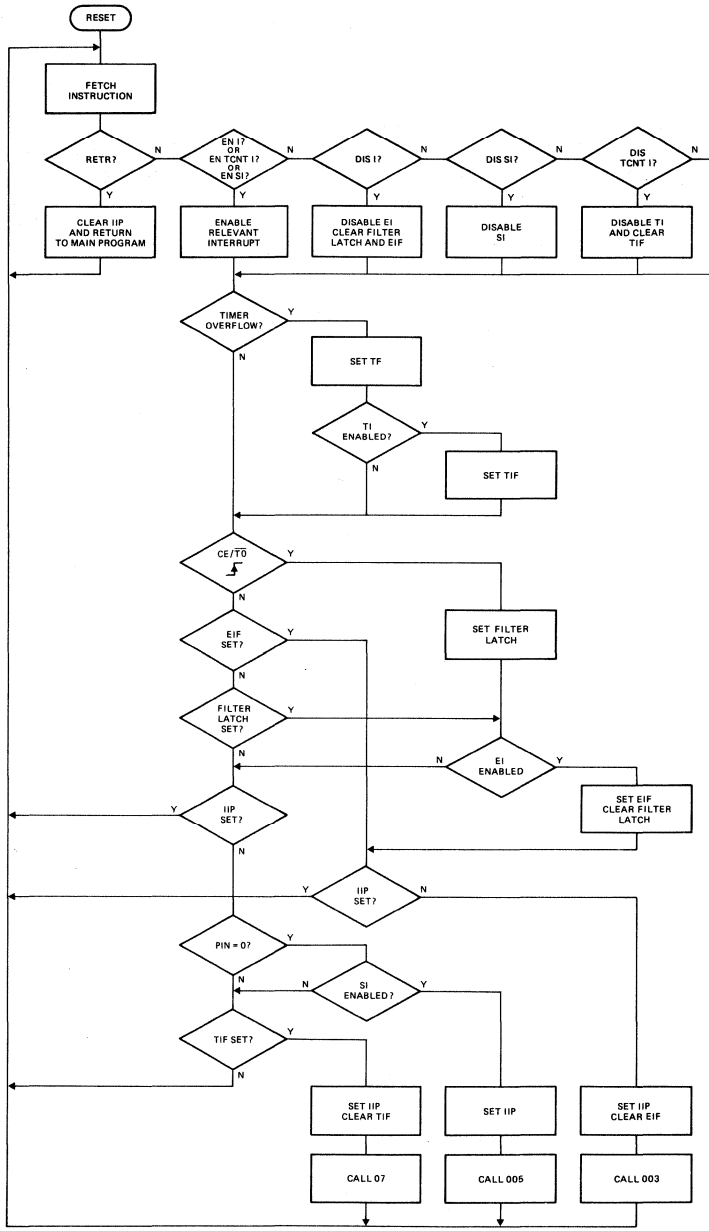
**Interrupt Priority**

If simultaneous interrupts occur, their priority is as follows:

External (highest)
SIO
Timer/Counter (lowest)

An interrupt routine can only be interrupted by a hardware RESET and cannot be interrupted by other interrupts (which will be latched). When the interrupt routine is terminated by the RETR instruction, at least one instruction of the main program will be executed before another interrupt routine is entered.

## FUNCTIONAL DESCRIPTION (continued)




| | | | |
|---|---|---|---|
| EI | External Interrupt | TF | Timer Flag |
| SI | Serial Interrupt | TIF | Timer Interrupt Flag |
| TI | Timer/counter Interrupt | PIN | Pending Interrupt Not (SIO) |
| EIF | External Interrupt Flag | IIP | Interrupt In Progress Flag |

Fig. 15 (a) Flow chart illustrating the interrupt handling sequence.

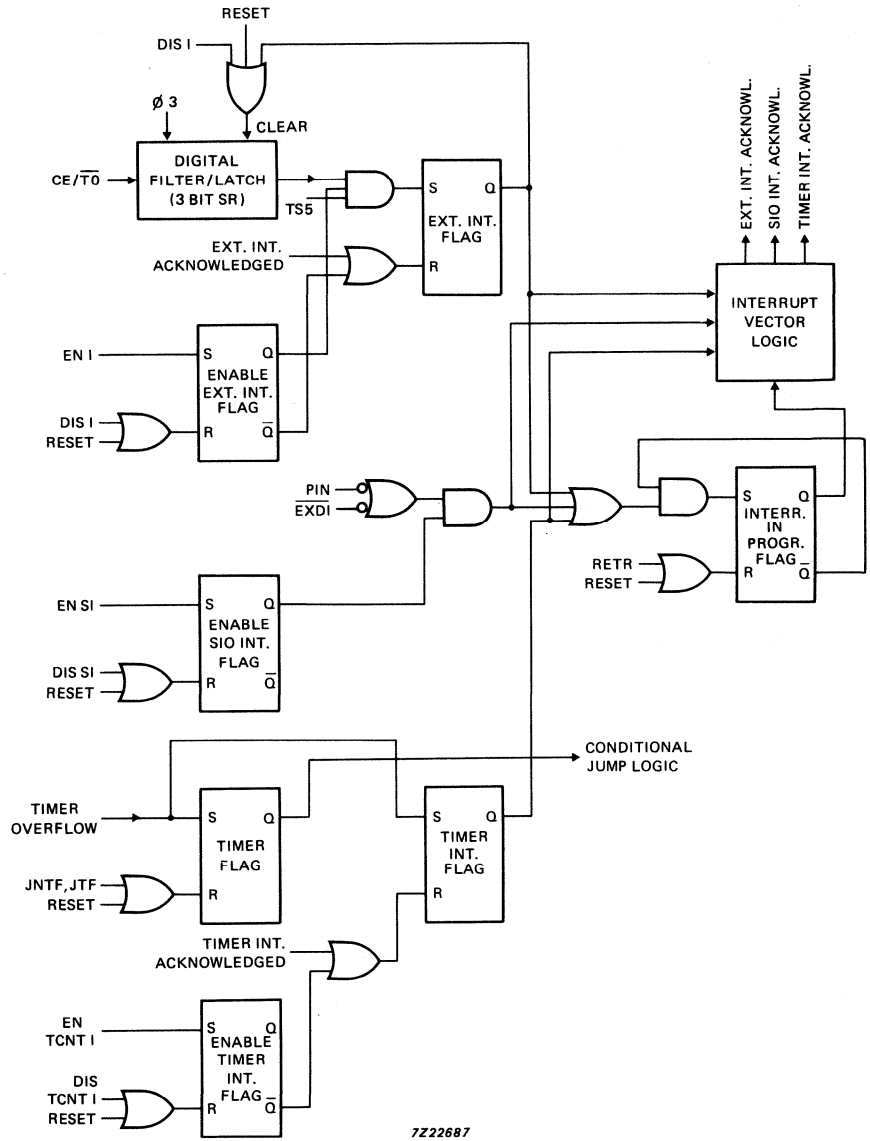



Fig. 15 (b) Simplified schematic of interrupt logic, to be used in conjunction with the functional description.

**Notes to Figure 15**

1. The positive edge of CE/$\overline{T0}$ is always latched in the digital filter/latch.
2. Correct interrupt timing is ensured when CE/$\overline{TO}$ is LOW for >4 CP followed by a HIGH for >7 CP.
3. When the interrupt in progress flag is set, further external and timer interrupts are latched but ignored, until RETR is executed.
4. A DIS I instruction always clears a pending external interrupt.
5. For all flip-flops, RESET overrules SET.

**FUNCTIONAL DESCRIPTION** (continued)

**Oscillator** (see Fig. 16)

The oscillator can be inhibited by the STOP instruction under software control. It is also inhibited when a low-supply voltage condition is present to prevent discharge of a weak back-up battery. Provided the supply voltage is within the operating range, the oscillator will be restarted after a STOP instruction by a HIGH level at the CE/$\overline{T0}$ pin or a HIGH level at the RESET pin.

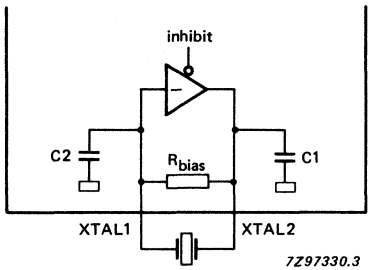



Fig. 16 Oscillator with integrated elements.

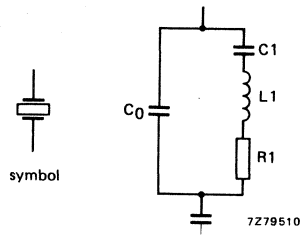



Fig. 17 Crystal unit equivalent circuit.

The values of crystal series resistance R1 and the crystal's total load capacitance $C_L$ ($C_0$ + wiring + external capacitors) must not be above the curve (Fig. 18) for the corresponding frequency.
Note: if external capacitors are connected to XTAL1 and XTAL2, they must be of equal value.

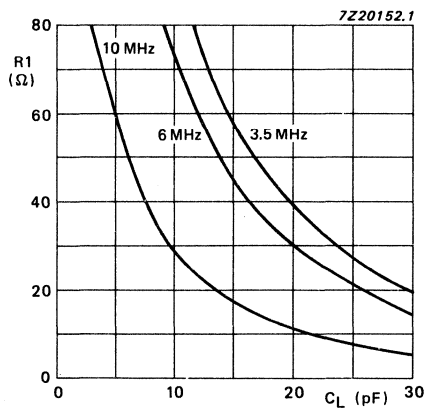



Fig. 18 Crystal circuit criteria.

XTAL2 is the output of the inverting amplifier. An external clock can be applied to XTAL1.
A machine cycle consists of 10 time slots; each time slot is 3 oscillator periods.

In telephony applications the 3.58 MHz crystal provides an 8.4 $\mu$s machine cycle.

**Timer/event counter** (see Fig. 19)

An internal 8-bit up-counter is provided. This can count external events, modulo-32 machine cycles, or machine cycles directly. Table 8 shows the instructions that control the counter and the prescaler, and the functions performed.

When used as a timer, the input to the counter is either the overflow or input of a 5-bit prescaler. When used as an event counter, LOW-to-HIGH transitions on pin T1 are counted. The counter is incremented during a machine cycle only if the falling edge occurs during the first 7 time slots; otherwise it is incremented during the next cycle. The maximum rate at which the counter may be incremented is once every machine cycle. When the counter overflows, the Timer flag is set. The flag can be tested and reset using the JTF (jump if Timer flag = logic 1) or JNTF (jump if Timer flag = logic 0) instruction. Overflow also generates an interrupt request to the microcontroller by setting the Timer Interrupt Flag when the timer/event counter interrupt is enabled.

**Table 8** Timer/event counter control

| function | timer mode modulo-1, modulo-32* | counter mode |
|---|---|---|
| CLEAR | MOV T,A (A) = 0 or RESET | MOV T,A (A) = 0 or RESET |
| PRESET | MOV T,A | MOV T,A |
| START | STRT T | STRT CNT |
| STOP | STOP TCNT or RESET | STOP TCNT or RESET |
| TEST | JTF/JNTF | JTF/JNTF |
| READ** | MOV A,T | MOV A,T |

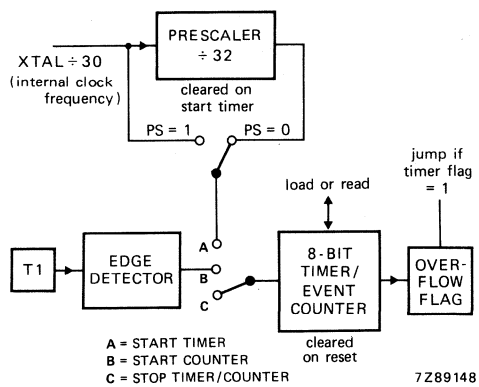



Fig. 19 Timer/event counter.

* With prescaler select (PS) = logic 0, the timer is incremented every 32 machine cycles; with PS = logic 1 the timer is incremented every machine cycle (prescaler not used), the prescaler is cleared by the STRT T instruction and is not readable.

** READ does not disturb the counting process.

**FUNCTIONAL DESCRIPTION** (continued)

**Program status word** (see Fig. 20)

The program status word (PSW) is an 8-bit word (1 byte) in the CPU which stores information about the current status of the microcontroller.
The PSW bits are:

- Bits 0 to 2 — stack pointer bits ($SP_0$, $SP_1$, $SP_2$)
- Bit 3 — prescaler select (PS);
  0 = modulo-32; 1 = modulo-1 (no prescaling)
- Bit 4 — working register bank select (RBS);
  0 = register bank 0; 1 = register bank 1
- Bit 5 — not used (1)
- Bit 6 — auxiliary carry (AC); half-carry bit generated by an ADD instruction and used by the decimal adjust instruction DA A
- Bit 7 — carry (CY); the carry flag indicates that the previous operation resulted in an overflow of the accumulator

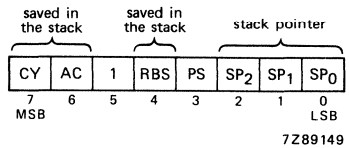



Fig. 20 Program status word.

All bits can be read using the MOV A, PSW instruction. Bits 7 and 6 are set and cleared by CPU operation. Bit 4 can be changed by the SEL RB instructions, bit 3 by the MOV PSW, A instruction, and bits 0, 1 and 2 by the CALL, RET or RETR instructions and in the event of an interrupt. Bits 7, 6 and 4 are stored in the program counter stack during subroutine and interrupt calls. These bits are restored in the PSW with a RETR (return and restore) instruction which must be used at the end of an interrupt service routine and can be used at the end of a normal subroutine. The RET instruction has no restore feature and cannot be used at the end of an interrupt service routine.

**Program counter** (see Fig. 21)

The 13-bit program counter is able to address 8 K bytes of ROM. The arrangement of the bits is shown in Fig. 21. During an interrupt subroutine PC 11 and PC 12 are forced to logic 0. All 13 bits are saved in the stack during CALL and interrupt routines.

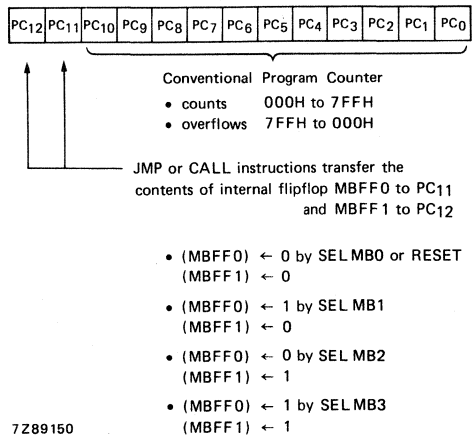



Fig. 21 Program counter.

### Central processing unit

The PCD3346 has arithmetic, logical and branching capabilities. The DA A, SWAP A and XCHD instructions simplify BCD arithmetic and the handling of nibbles. The MOVP A,@A instruction permits efficient table look-up from the current ROM page.

### Conditional branch logic

The conditional branch logic within the microcontroller enables several conditions, internal and external to the microcontroller, to be tested by the user's program. Table 9 lists the conditional jump instructions used to change the program sequence. The DJNZ instruction decrements a designated register or data memory location and branches if the contents are not zero. This instruction is useful for looping control. The JMPP@A instruction allows multiway branches to destinations determined by the contents of the accumulator.

**Table 9** Conditional branches

| test | jump condition | jump instruction |
|---|---|---|
| accumulator | all bits zero | JZ |
| | any bit non-zero | JNZ |
| accumulator bit test | 1 | JB0 to JB7 |
| carry flag | 1 | JC |
| | 0 | JNC |
| timer overflow flag | 1 | JTF |
| | 0 | JNTF |
| test input T0 | 1 | JNT0 |
| | 0 | JT0 * |
| test input T1 | 1 | JT1 |
| | 0 | JNT1 |
| register | non-zero | DJNZ |

* Because of the inverted interrupt input CE/$\overline{T0}$ the conditional jump JT0 is also inverted.

### Test input T1

The T1 input line can be used as:

- A test input for branch instructions JT1 and JNT1
- An external input to the event counter

When used as a test input:

- JT1 instruction tests for logic 1 level
- JNT1 instruction tests for logic 0 level

When used as an input to the event counter, T1 must be LOW for $>$ 4 CP, and then HIGH for $>$ 4 CP. A transition can be recognized every 30 oscillator clock periods (1 machine cycle).

There is no internal pull-up or pull-down resistor connected to the T1 input. When T1 is not used, it must be tied to $V_{DD}$ or $V_{SS}$.

### Power-on-reset

The internal power-on reset circuit monitors the supply voltage $V_{DD}$. As long as the supply voltage remains below the internal reference level $V_{ref}$ (typically 1.5 V), the oscillator is inhibited and RESET has an undefined level. When $V_{DD}$ rises above the internal reference level, the oscillator is released and RESET is pulled high to $V_{DD}$ by TR1 for a period $t_D$ (typically 50 $\mu$s).

N.B. Because of the narrow bandwidth of the crystal, the start-up time of the oscillator is typically 10 ms.

Three applications of power-on-reset are possible:

1. If $V_{DD}$ can be switched with a fast rise time i.e. $V_{DD}$ reaches its minimum operating value (corresponding to the selected oscillator frequency) before the RESET signal has finished ($t_D$), then no extra components are required (see Figs 22 and 23). Note that the first instruction is executed after the oscillator start-up time plus 1866 clock periods have elapsed.
2. If $V_{DD}$ has a slow rise time then the RESET signal should be stretched by an external RC circuit (see Figs 24 and 25). In the event of a short drop in the supply voltage, the diode path rapidly discharges the capacitor to ensure a reliable power-on-reset. To ensure a correct reset, the RESET signal should reach at least 70% of the final value of $V_{DD}$. Given that the RESET voltage and $V_{DD}$ rise exponentially, the above requirement is satisfied when the time constant $\tau$ of the RESET pulse is $>$ 8 times the time constant of $V_{DD}$. If $V_{DD}$ rises linearly, then a RESET time constant $>$ 2 times the rise time of $V_{DD}$ is required.

   When a reset is completed (RESET goes LOW) before the oscillator has started up, program execution begins after the oscillator start-up time plus 1866 clock periods have elapsed (see Fig. 25). If the oscillator is started up prior to the completion of RESET, then program execution begins 1866 clock periods after RESET goes LOW.
3. Fig. 26 shows an external reset during power-on. The external reset signal must remain HIGH until $V_{DD}$ has reached its minimum operating value corresponding to the selected oscillator frequency. When a reset is completed (RESET goes LOW) before the oscillator has started up, program execution begins after the oscillator start-up time plus 1866 clock periods have elapsed (see Fig. 27). If the oscillator is started-up prior to the completion of RESET, then program execution begins 1866 clock periods after RESET goes LOW.

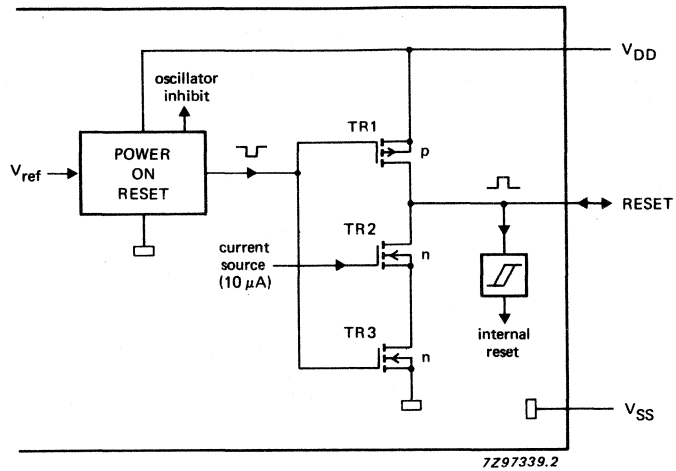



Fig. 22 Power-on-reset configuration.

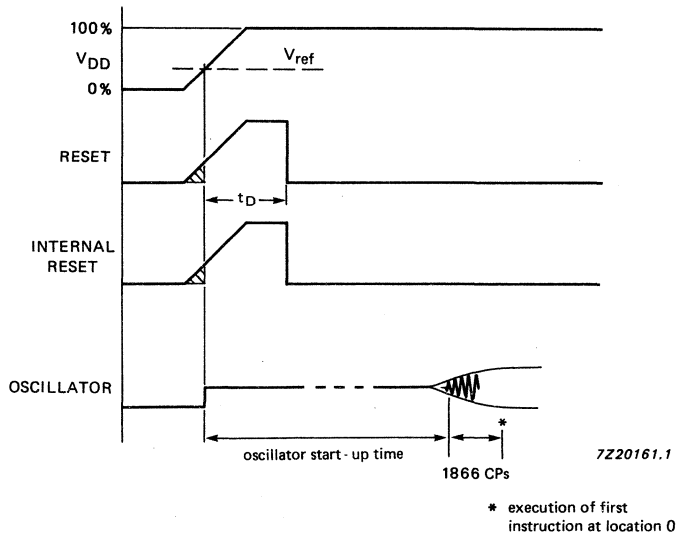



Fig. 23 Timing of power-on-reset with fast rise time.

## FUNCTIONAL DESCRIPTION (continued)

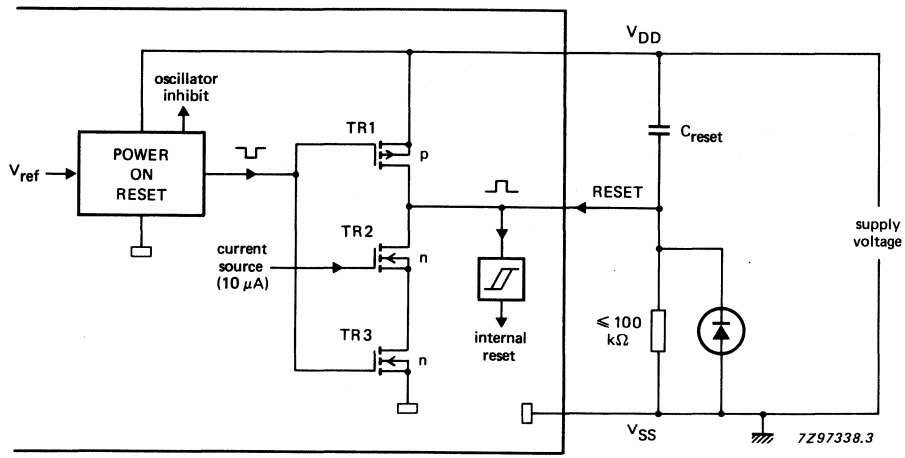



Fig. 24 Stretched power-on-reset with external components.

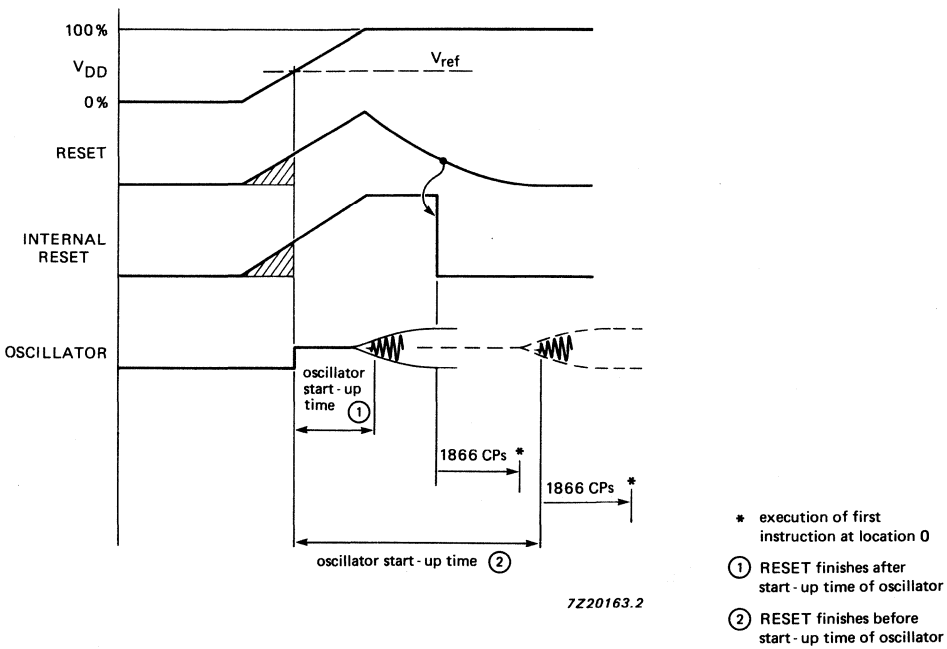



Fig. 25 Timing of power-on-reset with a slowly rising $V_{DD}$ and a stretched RESET pulse.

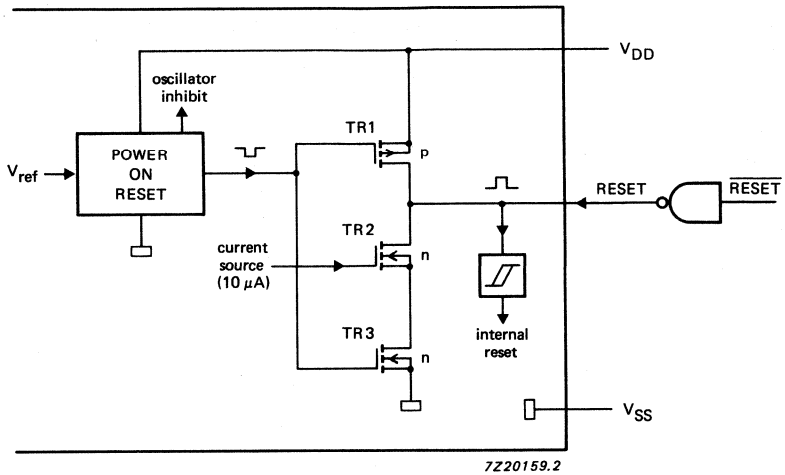



Fig. 26 External power-on-reset configuration.

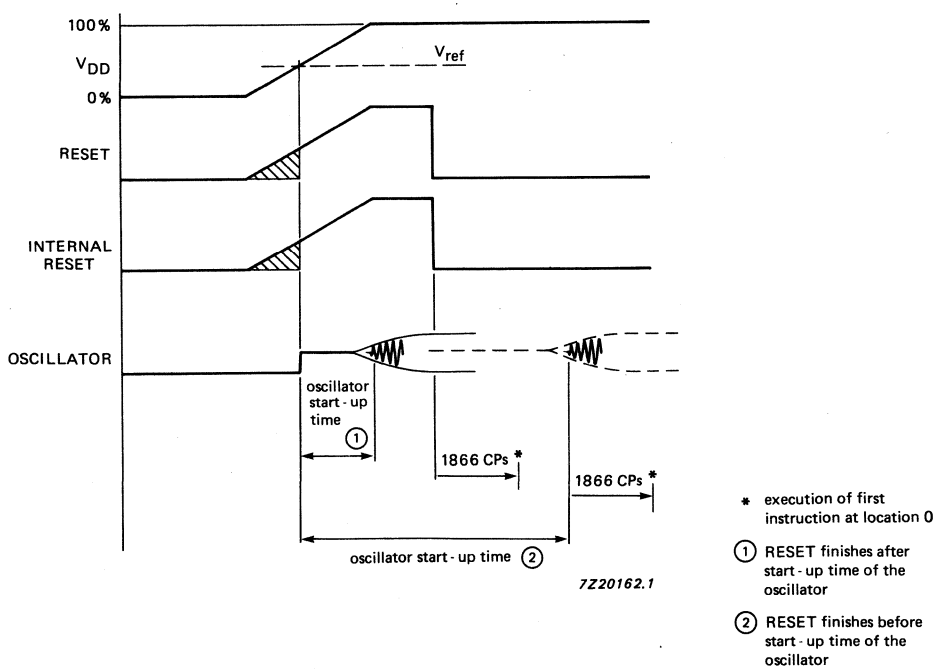



Fig. 27 Timing of external power-on-reset.

## RATINGS

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

| parameter | symbol | min. | max. | unit |
|---|---|---|---|---|
| Supply voltages | $V_{DD}$ | −0.8 | + 8 | V |
| All input voltages | $V_I$ | 0.5 | $V_{DD}$ + 0.5 | V |
| DC current into any input or output | $\pm I_I \pm I_O$ | – | 10 | mA |
| Total power dissipation (see note) | $P_{tot}$ | – | 500 | mW |
| Power dissipation per output | $P_O$ | – | 50 | mW |
| Storage temperature range | $T_{stg}$ | −65 | + 150 | °C |
| Operating ambient temperature range | $T_{amb}$ | −25 | + 70 | °C |
| Operating junction temperature | $T_j$ | – | 125 | °C |

## HANDLING

Inputs and outputs are protected against electrostatic discharge in normal handling. However, to be totally safe it is desirable to take normal precautions appropriate to handling MOS devices (see 'Handling MOS devices').

## THERMAL RESISTANCE

From junction to ambient

| | | | |
|---|---|---|---|
| SOT117 | $R_{th\ j\text{-}a}$ | = | 120 K/W |
| SOT136A | $R_{th\ j\text{-}a}$ | = | 150 K/W |

## DC CHARACTERISTICS

$V_{DD}$ = 2.5 to 6.0 V; $V_{SS}$ = 0 V; $T_{amb}$ = −25 to + 70 °C; all voltages with respect to $V_{SS}$; unless otherwise specified

| parameter | symbol | min. | typ. | max. | unit |
|---|---|---|---|---|---|
| Supply voltage operating (see Fig. 28) | $V_{DD}$ | 2.5 | – | 6 | V |
| Supply current operating (see Fig. 29 and note 2) | | | | | |
| at $V_{DD}$ = 5 V; $f_{XTAL}$ = 10 MHz | $I_{DD}$ | – | 2.0 | 3.5 | mA |
| at $V_{DD}$ = 5 V; $f_{XTAL}$ = 6 MHz | $I_{DD}$ | – | 1.2 | 2.4 | mA |
| at $V_{DD}$ = 3 V; $f_{XTAL}$ = 3.58 MHz | $I_{DD}$ | – | 0.4 | 0.8 | mA |
| IDLE mode (see Fig. 30 and note 2) | | | | | |
| at $V_{DD}$ = 5 V; $f_{XTAL}$ = 10 MHz | $I_{DD}$ | – | 1.0 | 2.0 | mA |
| at $V_{DD}$ = 5 V; $f_{XTAL}$ = 6 MHz | $I_{DD}$ | – | 0.7 | 1.2 | mA |
| at $V_{DD}$ = 3 V; $f_{XTAL}$ = 3.58 MHz | $I_{DD}$ | – | 0.25 | 0.4 | mA |
| STOP mode (see Fig. 36 and notes 1 and 2) | | | | | |
| at $V_{DD}$ = 2.5 V; $T_{amb}$ = 70 °C | $I_{DD}$ | – | – | 10 | μA |
| EEPROM Erase/Write cycle limitation* | | $10^4$ | | | |
| EEPROM data retention time | $T_{RET}$ | 10 | – | – | years |
| RESET I/O | | | | | |
| Switching level | $V_{RESET}$ | – | 1.5 | – | V |
| Sink current at $V_{DD} > V_{RESET}$ | $I_{OL}$ | – | 7 | – | μA |
| **Inputs** | | | | | |
| Input voltage LOW | $V_{IL}$ | 0 | – | 0.3 $V_{DD}$ | V |
| Input voltage HIGH | $V_{IH}$ | 0.7 $V_{DD}$ | – | $V_{DD}$ | V |
| Input leakage current at $V_{SS} < V_I < V_{DD}$ | $\pm I_{IL}$ | – | – | 1 | μA |
| **Outputs** | | | | | |
| Output voltage LOW at $V_I = V_{SS}$ or $V_{DD}$; $\lvert I_O \rvert < 1$ μA | $V_{OL}$ | – | – | 0.05 | V |
| Output sink current LOW at $V_{DD}$ = 3 V; $V_O$ = 0.4 V | | | | | |
| except P2.3/SDA, SCLK (see Fig. 32) | $I_{OL}$ | 0.7 | 1.5 | – | mA |
| P2.3/SDA, SCLK (see Fig. 33) | $I_{OL}$ | 1.5 | – | – | mA |
| Pull-up output source current HIGH (see Fig. 34) | | | | | |
| at $V_{DD}$ = 3 V; $V_O$ = 0.9 $V_{DD}$ | $-I_{OH}$ | 10 | – | – | μA |
| at $V_{DD}$ = 3 V; $V_O$ = $V_{SS}$ | $-I_{OH}$ | – | – | 300 | μA |
| Push-pull output source current HIGH at $V_{DD}$ = 3 V; $V_O$ = $V_{DD}$ − 0.4 V | $-I_{OH}$ | 0.7 | 1.5 | – | mA |

* Verified on sampling bases.

**Notes to the DC characteristics**

1. Crystal connected between XTAL1 and XTAL2; T1 and CE both tied to $V_{SS}$.
2. $V_{IL} = V_{SS}$; $V_{IH} = V_{DD}$; all outputs disconnected, all open-drain outputs connected to $V_{SS}$.

## AC CHARACTERISTICS

$V_{DD}$ = 2.5 to 5.5 V; $V_{SS}$ = 0 V; $T_{amb}$ = −40 to + 85 °C. All voltages with respect to $V_{SS}$ unless otherwise specified

| parameter | symbol | min. | typ. | max. | unit |
|---|---|---|---|---|---|
| Rise time all outputs (note 1) | $t_R$ | – | 30 | – | ns |
| Fall time all outputs (note 1) | $t_F$ | – | 30 | – | ns |
| Cycle time (= 30 CP; note 2) | $t_{CY}$ | 3 | – | 67 | μs |

**Notes to the AC characteristics**

1. At $V_{DD}$ = 5 V; $T_{amb}$ = + 25 °C; $C_L$ = 50 pF.
2. 1 Time slot (TS) = 3 CP, 1 clock pulse (CP) = $1/f_{XTAL}$.

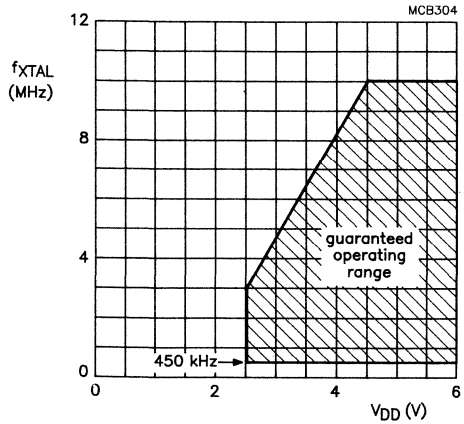
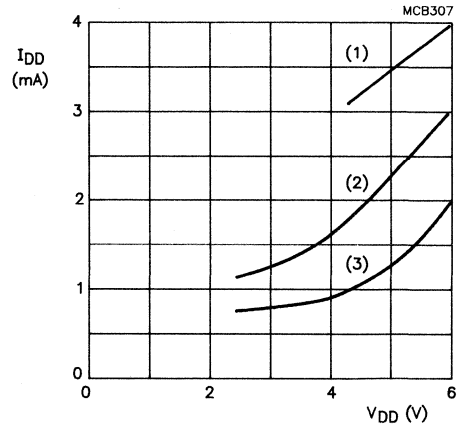
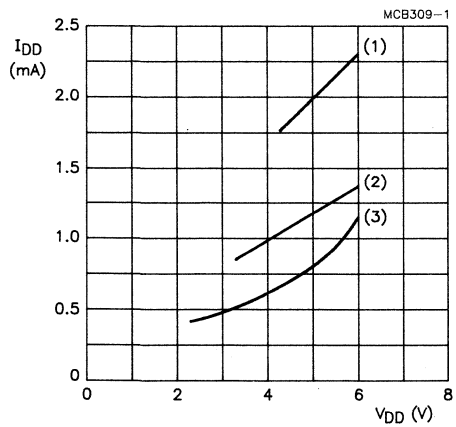



Fig. 28 Maximum clock frequency ($F_{XTAL}$) as a function of the supply voltage ($V_{DD}$).




(1) clock frequency = 10 MHz
(2) clock frequency = 6 MHz
(3) clock frequency = 3.58 MHz

Fig. 29 Maximum supply current ($I_{DD}$) in operation mode as a function of the supply voltage ($V_{DD}$).




(1) clock frequency = 10 MHz
(2) clock frequency = 6 MHz
(3) clock frequency = 3.58 MHz

Fig. 30 Maximum supply current ($I_{DD}$) in IDLE mode as a function of the supply voltage ($V_{DD}$).

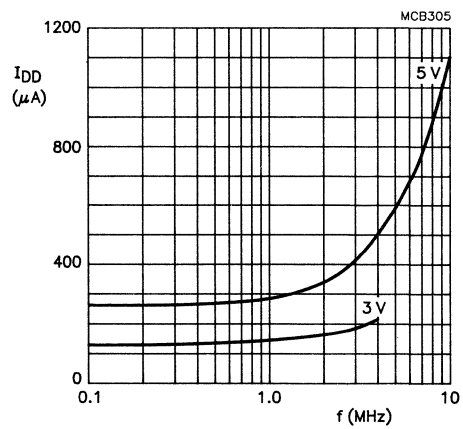



Fig. 31 Typical supply current during IDLE mode as a function of frequency at $V_{DD}$ = 3 V and $V_{DD}$ = 5 V.

**AC CHARACTERISTICS** (continued)

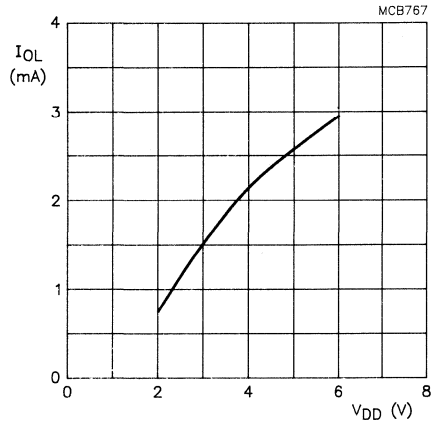



Fig. 32 Typical output sink current ($I_{OL}$), outputs P0.0 to P0.7 and P2.0 to P2.2, as a function of the supply voltage ($V_{DD}$); $V_O$ = 0.4 V.

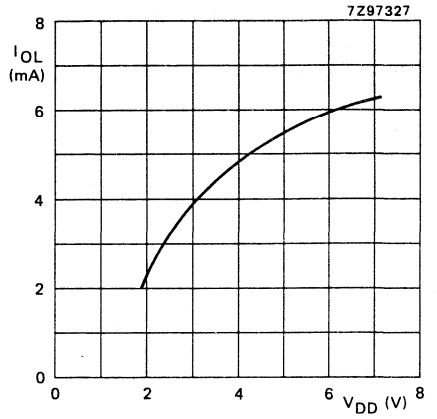
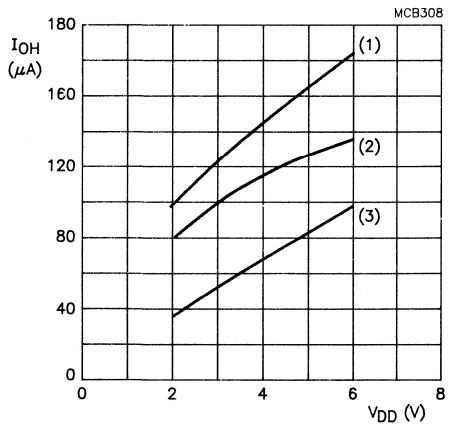



Fig. 33 Typical output sink current ($I_{OL}$), outputs P2.3/SDA and SCLK, as a function of the supply voltage ($V_{DD}$); $V_O$ = 0.4 V.




(1) $V_O = V_{SS}$
(2) $V_O = 0.7\ V_{DD}$
(3) $V_O = 0.9\ V_{DD}$

Fig. 34 Typical output source current ($-I_{OH}$) as a function of the supply voltage ($V_{DD}$).

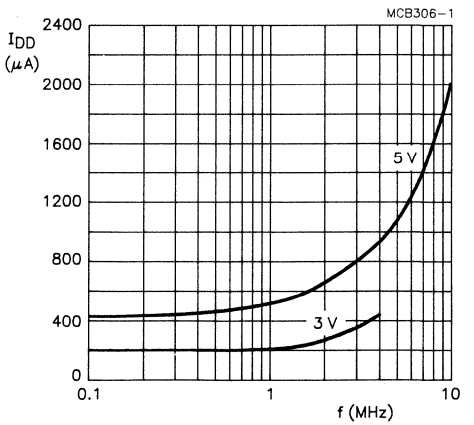
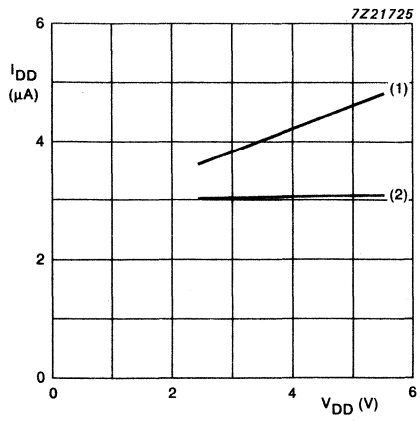



Fig. 35 Typical supply current during operating mode as a function of frequency at $V_{DD}$ = 3 V and $V_{DD}$ = 5 V.




(1) $T_{amb}$ = 85 °C
(2) $T_{amb}$ = 25 °C

Fig. 36 Typical supply current ($I_{DD}$) in STOP modes as a function of the supply voltage ($V_{DD}$).

**Table 13** Input timing shown in Fig. 37

| symbol | timing |
|---|---|
| $t_{BUF}$ | $\geqslant 14t_{XTAL}$ |
| $t_{HD;STA}$ | $\geqslant 14t_{XTAL}$ |
| $t_{HIGH}$ | $\geqslant 17t_{XTAL}$ |
| $t_{LOW}$ | $\geqslant 17t_{XTAL}$ |
| $t_{SU:STO}$ | $\geqslant 14t_{XTAL}$ |
| $t_{HD;DAT}$ | $> 0$ |
| $t_{SU;DAT}$ | $\geqslant 250$ ns |
| $t_{RD}$ | $\leqslant 1\ \mu s$ |
| $t_{RC}$ | $\leqslant 1\ \mu s$ |
| $t_{FD}$ | $\leqslant 1\ \mu s$ |
| $t_{FC}$ | $\leqslant 0.3\ \mu s$ |

**Notes to Table 13**

$t_{XTAL}$ = one period of the XTAL input frequency ($f_{XTAL}$)
= 167 ns for $f_{XTAL}$ = 6 MHz
These figures apply to all modes.

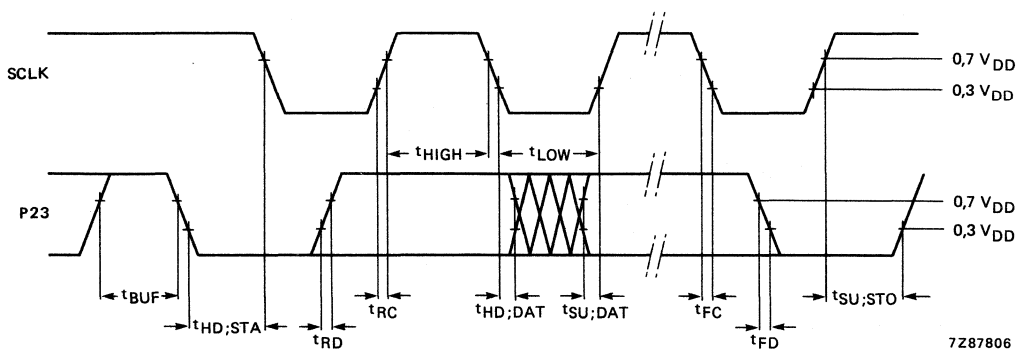



Fig. 37 Timing requirements for the P2.3 and SCLK *input* signals.

**AC CHARACTERISTICS** (continued)

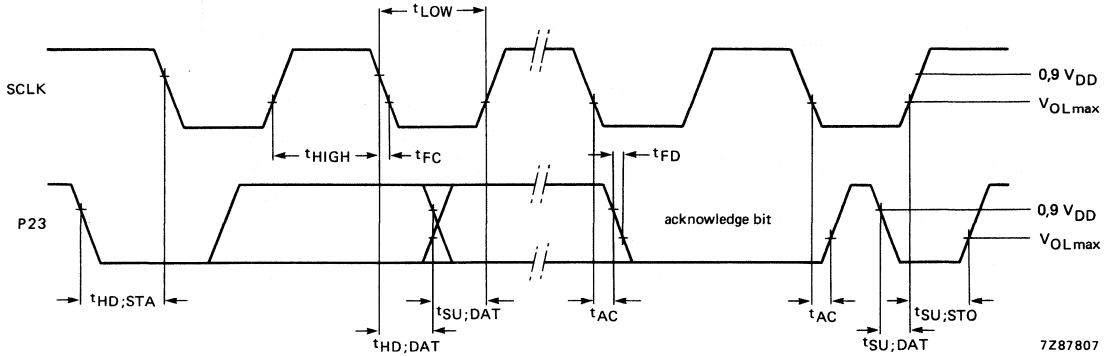



Fig. 38 Timing requirements for the P2.3 and SCLK *output* signals.

**Table 14** Output timing shown in Fig. 38

| symbol | timing | |
|---|---|---|
| | normal mode (ASC in S2 = 0) | low-speed mode (ASC in S2 = 1) |
| $t_{HD;STA}$ | ½ (DF + 9) $t_{XTAL}$ | ¾ (DF + 9) $t_{XTAL}$ |
| $t_{HIGH}$ | ½ (DF) $t_{XTAL}$ | ¾ (DF) $t_{XTAL}$ |
| $t_{LOW}$ | ½ (DF) $t_{XTAL}$ | ¼ (DF) $t_{XTAL}$ |
| $t_{SU;STO}$ | ½ (DF − 3) $t_{XTAL}$ | ¼ (DF − 3) $t_{XTAL}$ |
| $t_{HD;DAT}$ (slave transmitter) any DF | ⩾ $9t_{XTAL}$<br>⩽ $12t_{XTAL}$ | ⩾ $9t_{XTAL}$<br>⩽ $12t_{XTAL}$ |
| $t_{HD;DAT}$ (master transmitter) for DF ⩽ 51) | ⩾ $9t_{XTAL}$<br>⩽ $12t_{XTAL}$ | –<br>– |
| for DF ⩽ 99 | –<br>– | ⩾ $9t_{XTAL}$<br>⩽ $12t_{XTAL}$ |
| $t_{SU;DAT}$ (master transmitter) for DF > 51 | ⩾ $15t_{XTAL}$<br>⩽ $24t_{XTAL}$ | –<br>– |
| for DF > 99 | –<br>– | ⩾ $15t_{XTAL}$<br>⩽ $24t_{XTAL}$ |
| $t_{AC}$ | ⩾ $9t_{XTAL}$<br>⩽ $12t_{XTAL}$ | ⩾ $9t_{XTAL}$<br>⩽ $12t_{XTAL}$ |
| $t_{FD}$, $t_{FC}$ | ⩽ 100 ns at $C_b$ = 400 pF | ⩽ 100 ns at $C_b$ = 400 pF |

**Notes to Table 14**

$t_{XTAL}$ = one period of the XTAL input frequency ($f_{XTAL}$) = 167 ns for $f_{XTAL}$ = 6 MHz.
DF = divisor (see Table 7 Serial I/O section).
$C_b$ = the maximum bus capacitance for each line.

# CMOS MICROCONTROLLER WITH ON-CHIP DTMF GENERATOR

## GENERAL DESCRIPTION

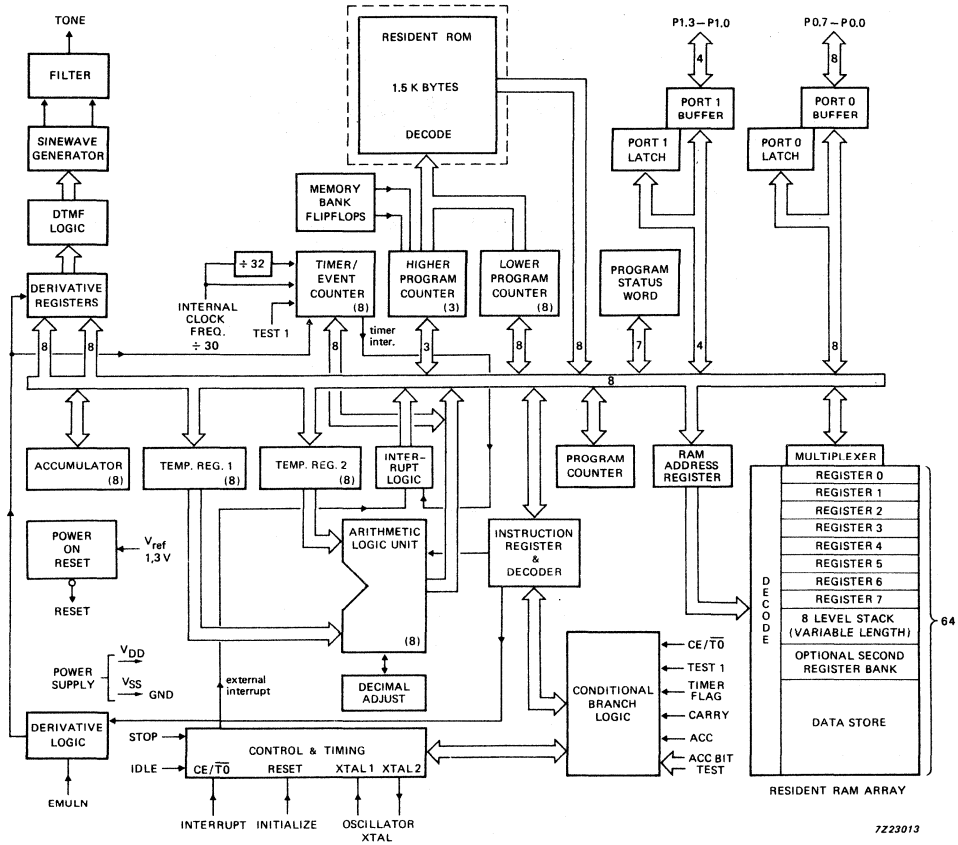
The PCD3347 is a single-chip 8-bit microcontroller fabricated in CMOS and is a member of the PCD33XX family. It has an on-chip dual tone multi-frequency (DTMF) generator and other features for application in telephone sets. For further detailed information, see PCD33XX family specification.

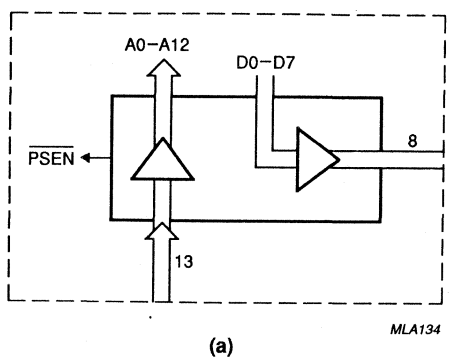
### Features

- 8-bit CPU, ROM, RAM, I/O in a single 20-lead DIL or SO package
- 1536 ROM bytes
- 64 RAM bytes
- On-chip DTMF tone generator
- On-chip voltage reference for supply and temperature-independent tone output
- On-chip filtering for low output distortion (CEPT CS203 compatible)
- 12 quasi-bidirectional I/O port lines
- Two test inputs: one of which is also the external interrupt input (CE/$\overline{T0}$)
- Single-level vectored interrupts: external, timer/event counter
- 8-bit programmable timer/event counter
- Over 80 instructions (based on MAB8048)
- All instructions 1 or 2 cycles
- Clock frequency 3,58 MHz
- Single supply voltage from 2,5 V to 6 V
- Low standby voltage and current
- STOP and IDLE mode
- On-chip oscillator
- Configuration of all I/O port lines individually selected by mask: pull-up, open drain or push-pull
- Power-on-reset circuit and low supply voltage detection
- Reset state of all ports individually selected by mask
- Operating temperature range: −25 to + 70 °C

## PACKAGE OUTLINES

PCD3347P: 20-lead DIL; plastic (SOT146).
PCD3347T: 20-lead mini-pack; plastic (SO20; SOT163A).




(a)

Fig. 1 PCD3347 block diagram: the function in the dotted outline is replaced as shown in (a) for the PCD3344B 'piggy-back' version.

**PINNING** (for normal operation)

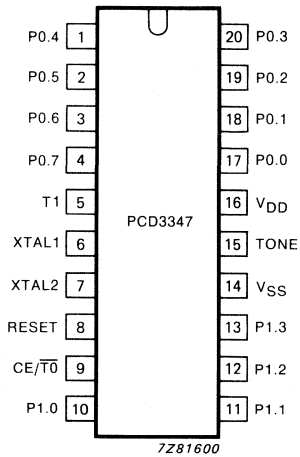



Fig. 2 Pinning diagram.

**PIN DESIGNATION**

| | | |
|---|---|---|
| 17-20, 1-4 | P0.0-P0.7 | Port 0: 8-bit quasi-bidirectional I/O port. |
| 5 | T1 | Test 1. test input, directly tested by conditional branch instructions JT1 and JNT1. T1 also functions as an input to the 8-bit timer/event counter using the STRT CNT instruction. |
| 6 | XTAL1 | Crystal input: connection to the timing component (crystal) which determines the frequency of the internal oscillator; is also the input for an external clock source. |
| 7 | XTAL2 | Connection to other side of timing component. |
| 8 | RESET | Reset input (active HIGH): used to initialize the processor or output of the power-on-reset circuit. |
| 9 | CE/$\overline{T0}$ | Interrupt/Test 0: external interrupt input (sensitive to positive-going edge)/test input pin. When used as a test input is directly tested by conditional branch instructions JT0 and JNT0. |
| 10-13 | P1.0-P1.3 | Port 1: 4-bit quasi-bidirectional I/O port. |
| 14 | $V_{SS}$ | Ground: circuit earth potentional. |
| 15 | TONE | Tone output: single or dual tone frequency output with on-chip filtering for low output distortion (CEPT CS203 compatible). This generator is controlled via the internal processor bus. |
| 16 | $V_{DD}$ | Power supply: 2,5 to 6 V. |

## FUNCTIONAL DESCRIPTION

### Program memory PCD3347

The program memory comprises 1536 bytes (8-bit words) in a read-only memory (ROM). Each location is directly addressable by the program counter. The memory is mask-programmed at the factory. Figure 3 shows the program memory map.

Three program memory locations are of special importance:

- Location 0; contains the first instruction to be executed after the processor is initialized (RESET),
- Location 3; contains the first byte of an external interrupt service subroutine,
- Location 7; contains the first byte of a timer/event counter interrupt service subroutine.

The program memory is divided into location 'pages', each of 256 bytes. This division applies only for conditional branches. A CALL instruction can transfer control to a subroutine on any 'page'; RET and RETR instructions can transfer control from a subroutine back to the main program.

### Data memory PCD3347

Data memory consists of 64 bytes (8-bit words) of random-access data memory (RAM). All locations are indirectly addressable using RAM pointer registers; up to 16 designated locations are directly addressable. Memory also includes an 8-level program counter stack addressed by a 3-bit stack pointer. Figure 4 shows the data memory map.

*Working registers*

Locations 0 to 7 are designated as working registers, directly addressable by the direct register instructions. Ease of addressing, and a minimum requirement of instruction bytes to manipulate their contents, makes these locations suitable for storing frequently-addressed intermediate results. This bank of registers can be selected by the SEL RB0 instruction.

Executing the select register bank instruction SEL RB1, designates locations 24 to 31 as working registers, instead of locations 0 to 7, and these are then directly addressable. This second bank of working registers may be used as an extension of the first or reserved for use during interrupt service subroutines saving the first bank for use in the main program. If the second bank is not used, locations 24 to 31 may serve as general purpose RAM.

The first locations of each bank contain the RAM pointer registers R0, R1, R0' and R1', which indirectly address all RAM locations.

All RAM locations make efficient program loop counters when used with the decrement register and test instruction DJNZ.

*Program counter stack*

Locations 8 to 23 may be designated as an 8-level program counter stack (2 locations per level), or as general purpose RAM. The program counter stack (Fig. 5) enables the processor to keep track of the return addresses and status generated by interrupts or CALL instructions by storing the contents of the program counter prior to servicing the subroutine. A 3-bit stack pointer determines which of the eight register pairs of the program counter stack will be loaded with next generated return address.

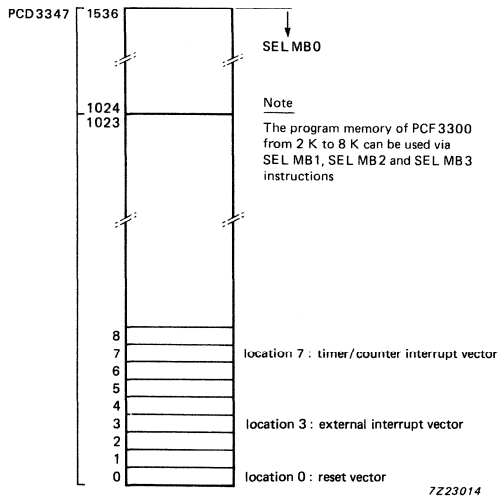



Fig. 3 Program memory map.

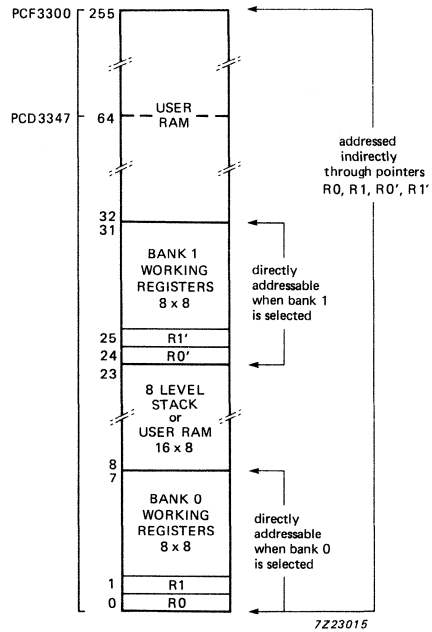



Fig. 4 Data memory map.

The stack pointer, when initialized to 000 by RESET, points to RAM locations 8 and 9. On the first subroutine CALL or interrupt, the contents of the program counter and bits 4, 6 and 7 of the program status word (PSW) are transferred to locations 8 and 9. The stack pointer increments by one and points to locations 10 and 11 ready for another CALL. Because an address may be up to 13 bits long, two bytes must be used to store each address.

At the end of a subroutine, which is signalled by a return instruction (RET or RETR), the stack pointer decrements by one and the contents of the register pair on top of the stack are transferred to the program counter. The saved PSW bits are transferred to the PSW only by the RETR instruction.

If not all 8 levels of subroutine and interrupt nesting are used, the unused portion of the stack may be used as any other indirectly addressable RAM locations. Possible locations from 32 to 64 may be used for storage of program variables or data.

Nesting of subroutines within subroutines can continue up to 8 times without overflowing the stack. If overflow does occur the deepest address stored (locations 8 and 9) will be overwritten and lost since the stack pointer overflows from 111 to 000. It also underflows from 000 to 111.

The value of the saved contents of the program counter is different for an interrupt CALL compared to a normal CALL to subroutine. With an interrupt CALL, the program counter return address is saved; with a subroutine CALL, the saved program counter value is one less than the program counter return address.

**FUNCTIONAL DESCRIPTION** (continued)

*Program counter stack (continued)*

| stack pointer | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 1 1 1 | ---------- | | | | | | | | R23/22 |
| 1 1 0 | ---------- | | | | | | | | R21/20 |
| 1 0 1 | ---------- | | | | | | | | R19/18 |
| 1 0 0 | ---------- | | | | | | | | R17/16 |
| 0 1 1 | ---------- | | | | | | | | R15/14 |
| 0 1 0 | ---------- | | | | | | | | R13/12 |
| 0 0 1 | ---------- | | | | | | | | R11/10 |
| 0 0 0 | PSW7 | PSW6 | PC12 | PSW4 | PC11 | PC10 | PC9 | PC8 | R9 |
| | PC7 | PC6 | PC5 | PC4 | PC3 | PC2 | PC1 | PC0 | R8 |

Fig. 5 Program counter stack.

**IDLE and STOP modes**

*IDLE mode*

When the microcontroller enters the IDLE mode via the IDLE instruction (H'01') the oscillator and timer/counter are kept running. The microcontroller exits from the IDLE mode by one of two interrupts if they are enabled or by activating a RESET. If the interrupt is not enabled the processor will remain in the IDLE mode. An active signal on the RESET pin restarts the microcontroller and a normal RESET sequence is executed (see Fig. 6).

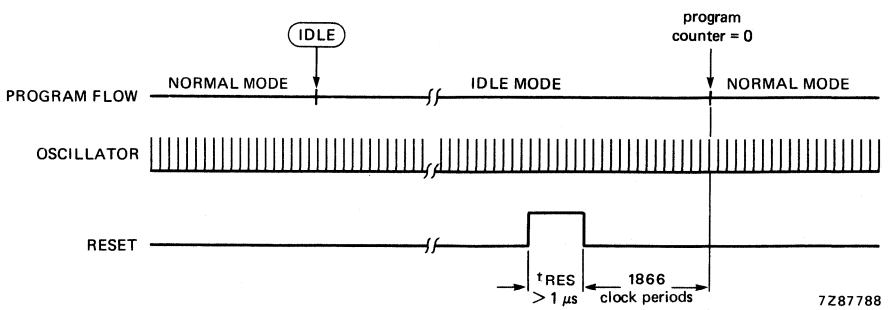



Fig. 6 Exit from IDLE mode via a RESET.

An active signal coming from an enabled interrupt causes the execution of the normal interrupt routine since normal interrupt scanning is still being carried out. A LOW-to-HIGH transition on the external interrupt pin (CE/$\overline{T0}$) reactivates the microcontroller. A HIGH level applied to CE/$\overline{T0}$ will reactivate the microcontroller only in the STOP mode. Thus, if CE/$\overline{T0}$ was HIGH before the microcontroller entered the IDLE mode, it must go LOW before the microcontroller can be reactivated (see Fig. 7).

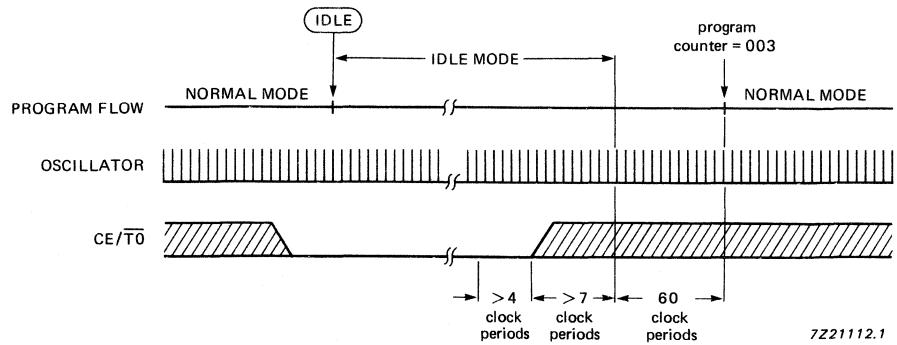



Fig. 7 Exit from IDLE mode via an interrupt.

Wake-up from the IDLE mode is ensured when CE/$\overline{T0}$ is LOW for 4 CP (clock periods) followed by a HIGH for 7 CP. After the initial forced CALL H'003' operation (60 CP) the program continues with the external interrupt service routine.

*STOP mode*

The microcontroller enters the STOP mode by the STOP instruction (H'22'). The oscillator is switched off. The internal status of the CPU, RAM contents and the state of I/O ports are not affected. The microcontroller can be brought-out of the STOP mode by an active signal at the external interrupt input or by an external RESET signal. When one of these two signals is applied an internal delay of 1866 CP is provided to ensure that all internal clocks are operating correctly before restart (see Fig. 8).

If the microcontroller exits from the STOP mode by activating RESET, a normal RESET sequence is executed.

If the microcontroller exits the STOP mode by pulling the external interrupt input pin HIGH, an interrupt sequence is executed only if the external interrupt is enabled. In this event the microcontroller resumes the normal program sequence after returning from the interrupt routine, as in the normal mode. If the interrupt is not enabled, it continues the normal program sequence, executing the instruction following the STOP instruction.

The microcontroller is restarted by a HIGH level applied at the CE/$\overline{T0}$ pin, and not by a LOW-to-HIGH transition as in a normal interrupt mechanism.

When the CE/$\overline{T0}$ level is active during the STOP instruction then no STOP is executed.

A HIGH level on the external interrupt input of at least 1 $\mu$s will cause the microcontroller to exit the STOP mode.

**FUNCTIONAL DESCRIPTION** (continued)

*STOP mode (continued)*

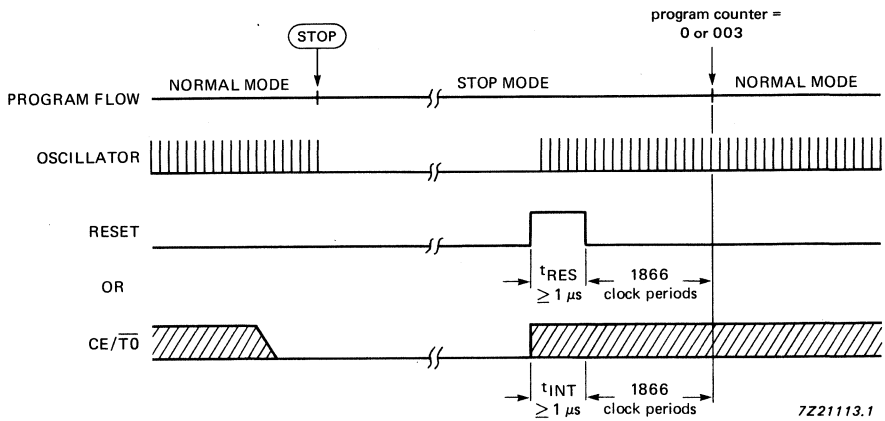



Fig. 8 Entering and exiting the STOP mode.

**Tone output (DTMF mode)**

*Control of the sinewave generator*

The on-chip sinewave oscillator is controlled by the 'derivative' registers Dx (x = H'O' to 'FF'). The instruction that controls the derivative registers is shown in Table 1.

**Table 1** Derivative register control

| mnemonic | opcode | description | function |
|---|---|---|---|
| MOV Dx,A | 8D Dx | move accumulator contents to derivative register | (Dx) ← (A) |

The instruction is 2 cycles/2 bytes. The second byte selects the derivative register to be addressed (H'O' to 'FF'). Register H'01' is for control of HIGH group frequencies, and register H'02' for control of LOW group frequencies. Thus data transport from accumulator to derivative register D01 is done by the 2-byte opcode 8D,01.

*Generation of frequencies*

The single and dual tones at the tone output are filtered by an on-chip switched-capacitor filter followed by an on-chip active RC low-pass filter. These ensure that the total harmonic distortion of the DTMF tones fulfil the CEPT CS 203 recommendations. An on-chip reference voltage provides output tone levels that are independent of the supply voltage.

The output frequency can be calculated as follows:

$$f_{out} = \frac{f_{XTAL}}{23\,(x + 2)} \quad \text{Hz}$$

x = 60 to 255 and is the decimal value of the appropriate ROM-code (see Table 2)

**Table 2** ROM-codes for DTMF applications

| telephone keyboard symbol | contents of low register (hex) | contents of high register (hex) |
|---|---|---|
| 0 | A3 | 72 |
| 1 | DD | 7F |
| 2 | DD | 72 |
| 3 | DD | 67 |
| 4 | C8 | 7F |
| 5 | C8 | 72 |
| 6 | C8 | 67 |
| 7 | B5 | 7F |
| 8 | B5 | 72 |
| 9 | B5 | 67 |
| A | DD | 5D |
| B | C8 | 5D |
| C | B5 | 5D |
| D | A3 | 5D |
| * | A3 | 7F |
| # | A3 | 67 |

DTMF generation is stopped by loading H'00' into both derivative registers.

**I/O facilities**

The PCD3344 family has 14 I/O lines arranged as:

- Port 0 parallel port of 8 lines (P0.0 to P0.7)
- Port 1 parallel port of 4 lines (P1.0 to P1.3)
- CE/$\overline{T0}$ external interrupt and test input. When used as a test input it can be directly tested by conditional branch instructions JT0 and JNT0.
- T1 test input which can alter program sequences when tested by conditional jump instructions JT1 and JNT1. T1 also functions as an input to the 8-bit timer/event counter.

**FUNCTIONAL DESCRIPTION** (continued)

*Parallel ports*

All parallel ports can be used as outputs or inputs, their structure is quasi-bidirectional. Output data written to a port is latched and remains unchanged until rewritten. Input data is not latched and so must be present until read by an input instruction.

Input lines are fully CMOS compatible, output lines can drive one LS-TTL or CMOS load.

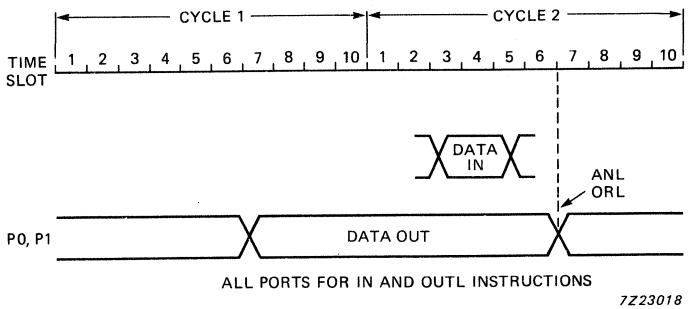



Fig. 9 Timing diagram of all ports on IN and OUTL instructions; for ANL and ORL instructions, the ports change on the time slot 7 of cycle 2.

Fig. 10 shows the quasi-bidirectional I/O interface with push-pull output and switched pull-up current source. Each line is pulled up to $V_{DD}$ via a constant current source (TR4), which is enabled via TR3 whenever one of the two output latches contains a logic 1. This current source provides sufficient current for a TTL HIGH level, yet can be pulled LOW by an external CMOS device, thus allowing the same pin to be used for both input and output.

When a logic 1 is written to the line for the first time ($\overline{MQ}$ = 1, $\overline{SQ}$ = 0), TR2 is switched on for the duration of the internal write pulse (one oscillator period) to provide a fast transition from logic 0 to logic 1. Subsequent writing of a logic 1 to the port lines will not switch TR2 on. This prevents unnecessary current through external components connected to the port lines of the same port which might be in the input mode and also connected to ground.

When a logic 0 is written to the line, TR3 switches off the current source. Current sinking capability is provided by TR1, which is now switched on. When used as an input, a logic 1 must first be written to the line, otherwise TR1 will remain low impedance.

In telephone applications this switched pull-up source may not be sufficient. Therefore the PCD3347 offers the possibility to select individually the 12 parallel port pins by the following mask options:

Option 1 –STANDARD PORT; quasi-bidirectional I/O with switched pull-up current source of 100 $\mu$A (typ.) and P-channel booster transistor TR2 (2,5 mA). TR2 is active only during 1 clock cycle (0,28 $\mu$s at 3,58 MHz).

Option 2 –OPEN DRAIN; quasi-bidirectional I/O with only an N-channel open drain output. Application as an output requires connection of an external pull-up resistor (Fig. 11).

Option 3 –PUSH-PULL OUTPUT; drive capability of the output will be 2,5 mA (typ.) at $V_{DD}$ = 3 V in both polarities. To avoid a large current flowing through the output transistors during the input mode, these push-pull pins must be used only as outputs (Fig. 12).

Also, individual mask selection of the RESET state of these port pins can be achieved by appending the following options S and R to options 1, 2 or 3.

Option S-SET; after RESET this pin will be initialized to HIGH

Option R-RESET; after RESET this pin will be initialized to LOW.

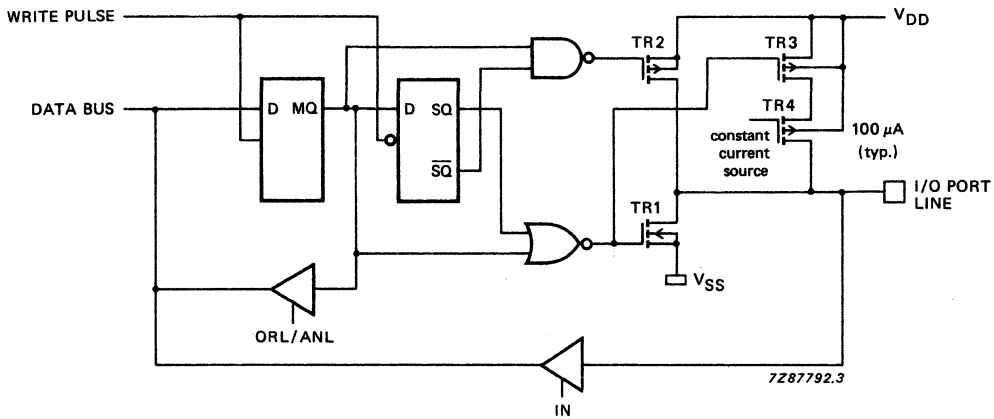



Fig. 10 Standard output with switched pull-up current source.

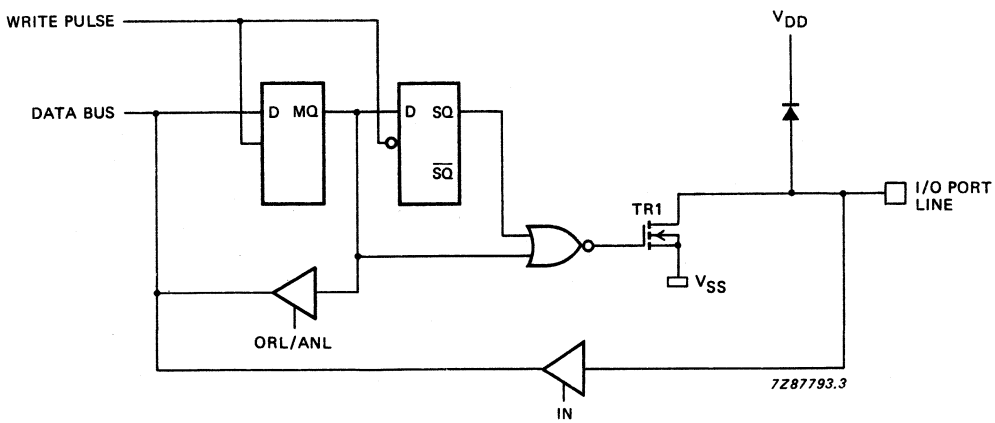



Fig. 11 Open drain output.

**FUNCTIONAL DESCRIPTION** (continued)

*Parallel ports (continued)*

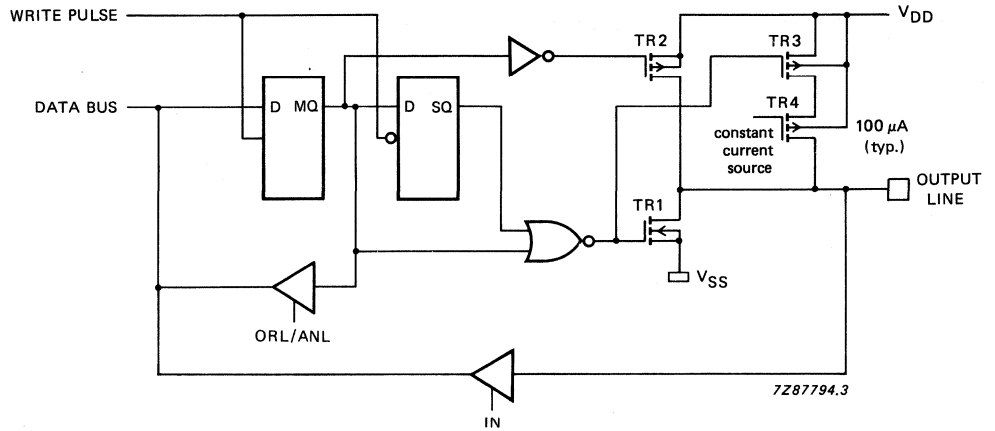



Fig. 12 Push-pull output.

**Interrupts** (see Fig. 13(a) and Fig. 13(b))

When an interrupt routine is entered, the contents of the program counter and bits 4, 6 and 7 of the PSW are saved in the program counter stack. The contents of the accumulator can only be saved by user software. Interrupt acknowledgement can be carried out by software via I/O ports. All interrupt routines must reside in memory bank 0; the SEL MB1, SEL MB2 and SEL MB3 instructions may not be used in an interrupt routine. An interrupt routine can only be terminated by the RETR (return and restore) instruction. During an interrupt routine, subroutine calls must be terminated by the RET instruction. Using the RETR instruction to terminate a subroutine called in an interrupt routine would terminate the interrupt routine prematurely and result in a wrong return address.

1. **External interrupt**

When the external interrupt is enabled, a HIGH-to-LOW transition on the CE/$\overline{TO}$ input initiates an external interrupt routine which forces a call to program memory location 3. The program counter points to the external interrupt vector address (003 H) between 2,6 and 3,6 machine cycles after the transition occurs. Interrupt latency depends on the instruction that is being executed when the transition occurs. External interrupts are latched in the External Interrupt Flag (EIF) even when they are not enabled. Execution of a DIS I instruction clears previously latched interrupts, the digital filter latch and the external interrupt flag.

2. **Timer/counter interrupt**

When the timer interrupt is enabled, a timer/counter overflow sets the Timer Interrupt Flag (TIF) and forces a CALL to location 7. The timer interrupts are only latched when they are enabled. The timer flag is set every time the timer/counter overflows and is not automatically reset when the timer/counter interrupt routine is called. It can only be cleared by the JTF and JNTF instructions or by a hardware RESET.

### 3. Simultaneous interrupts

If simultaneous interrupts occur their priority is as follows:

external (highest);
timer/counter (lowest).

An interrupt routine can only be interrupted by a hardware RESET and cannot be interrupted by other interrupts (which will be latched if enabled). When the interrupt routine is terminated by the RETR instruction, at least one instruction of the main program will be executed before another interrupt routine is entered.

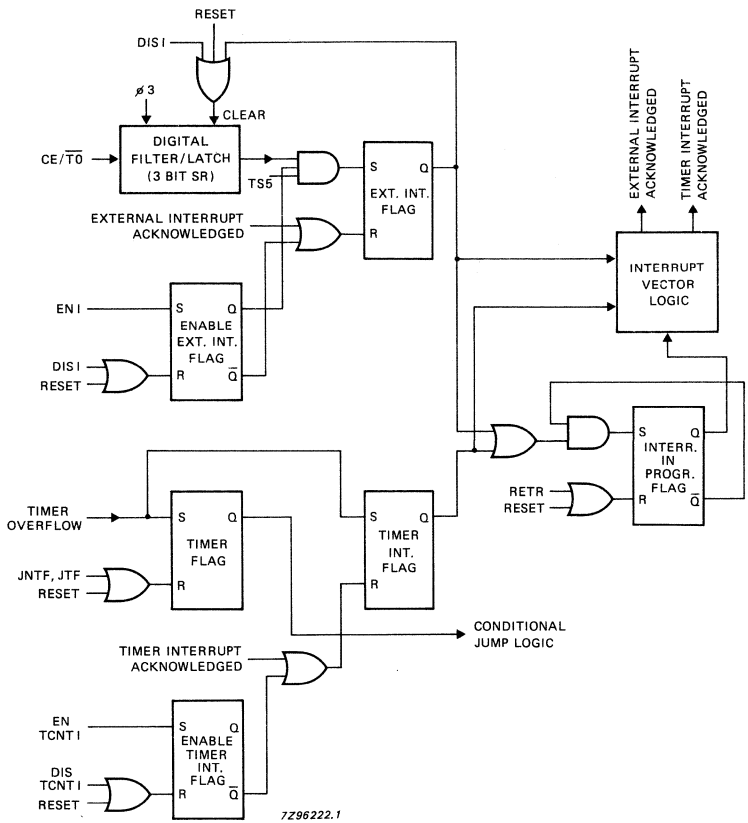


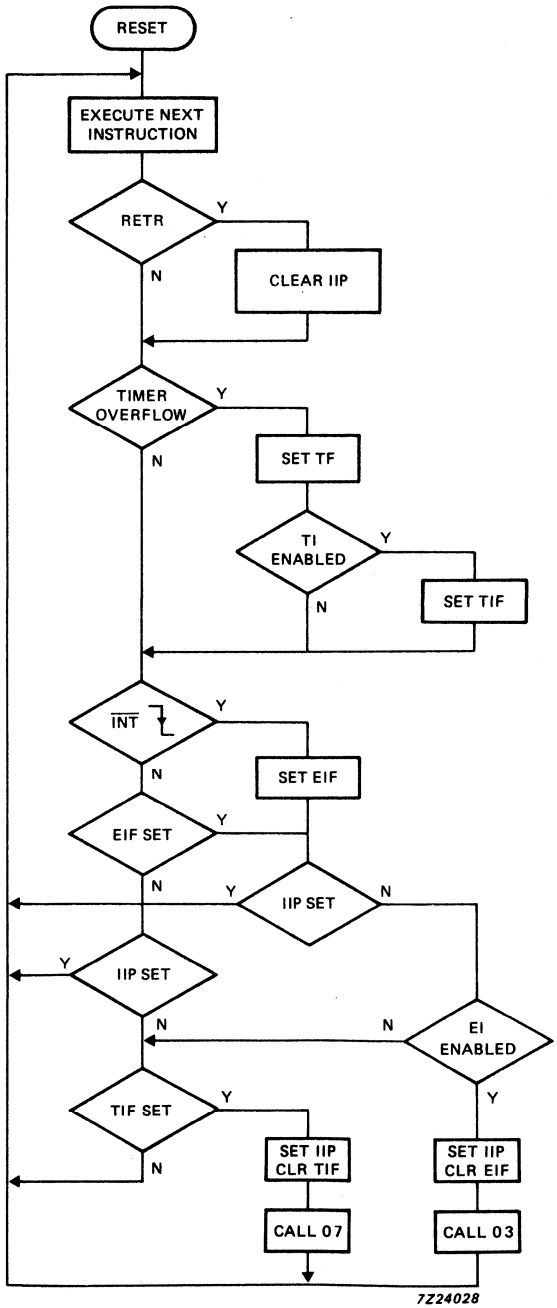

Fig. 13(a) Interrupt logic.

**Notes to figure 13(a)**

1. CE/$\overline{T0}$ positive edge is always latched in the digital filter/latch.
2. Correct interrupt timing is ensured when CE/$\overline{T0}$ is LOW for $>$ 4 CP followed by a HIGH for $>$ 7 CP.
3. When the interrupt in progress flag is set, further interrupts are latched but ignored, until RETR is executed.
4. A DIS I instruction always clears a pending external interrupt.
5. For all flip-flops, RESET overrules SET.

**FUNCTIONAL DESCRIPTION** (continued)

**Interrupts** (continued)




EI External Interrupt
TI Timer/counter Interrupt
EIF External Interrupt Flag
TF Timer Flag
IIP Interrupt In Progress flag

Fig. 13(b) Interrupt flowchart.

**Oscillator** (see Fig. 14)

The 3,58 MHz oscillator can be inhibited by the STOP instruction under software control. It is also inhibited when a low-voltage condition is present to prevent discharge of a weak back-up battery.

Provided the supply voltage is within the operating range, the oscillator will be restarted after a STOP instruction by a HIGH level at either the CE/$\overline{T0}$ or RESET pin.

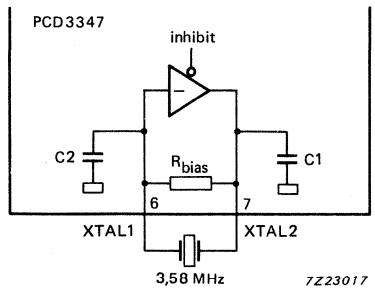



Fig. 14 Oscillator with integrated elements.

The oscillator has an output drive capability from pin 7 (XTAL2). An external clock can be applied to pin 6 (XTAL1). A machine cycle comprises 10 time slots, each time slot being 3 oscillator periods.

In telephony applications the 3,58 MHz crystal provides an 8,4 $\mu$s machine cycle. The range of the clock frequency is from 100 kHz up to a maximum which is a function of the supply voltage.

**Timer/event counter** (see Fig. 15)

An internal 8-bit up-counter is provided. This can count external events, modulo-32 machine cycles, or machine cycles directly. Table 3 gives the instructions that control the counter and the prescaler, and the functions performed.

When used as a timer, the input to the counter is either the overflow or input of a 5-bit prescaler. When used as an event counter, LOW-to-HIGH transitions on pin 8 (T1) are counted. The maximum rate at which the counter may be incremented is once every machine cycle (182,6 kHz for an 8,4 $\mu$s machine cycle). When the counter overflows, the timer flag is set. The flag can be tested and reset using the JTF (jump if timer flag = 1) or JNTF instruction. Overflow also generates an interrupt to the processor via setting of the Timer Interrupt Flag when the timer/event counter interrupt is enabled.

**FUNCTIONAL DESCRIPTION** (continued)

**Timer/event counter** (continued)

**Table 3** Timer/event counter control

| function | timer mode modulo-1, modulo-32* | counter mode |
|---|---|---|
| CLEAR | MOV T,A (A) = 0 or RESET | MOV T,A (A) = 0 or RESET |
| PRESET | MOV T,A | MOV T,A |
| START | STRT T | STRT CNT |
| STOP | STOP TCNT or RESET | STOP TCNT or RESET |
| TEST | JTF/JNTF | JTF/JNTF |
| READ** | MOV A,T | MOV A,T |

* With prescaler select, PS = 0, the timer counts modulo-32 machine cycles, with PS = 1 it counts modulo-1 cycles (prescaler not used); prescaler cleared with STRT T, prescaler not readable.

** READ does not disturb the counting process.

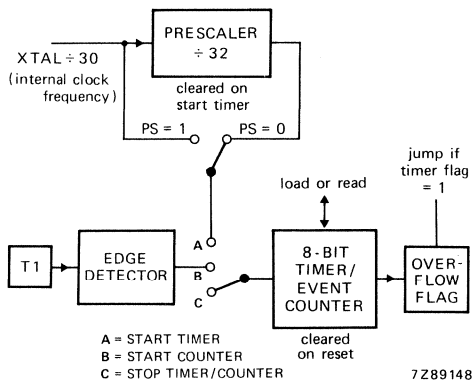



Fig. 15 Timer/event counter.

**Program status word** (see Fig. 16)

The program status word (PSW) is an 8-bit word (1 byte) in the CPU which stores information about the current status of the microcontroller.
The PSW bits are:

- Bits 0 to 2 — stack pointer bits ($SP_0$, $SP_1$, $SP_2$)
- Bit 3 — prescaler select (PS); 0 = modulo-32; 1 = modulo-1 (no prescaling)
- Bit 4 — working register bank select (RBS); 0 = register bank 0; 1 = register bank 1
- Bit 5 — not used (1)
- Bit 6 — auxiliary carry (AC); half-carry bit generated by an ADD instruction and used by the decimal adjust instruction DA A
- Bit 7 — carry (CY); the carry flag indicates that previous operation has resulted in an overflow of the accumulator.

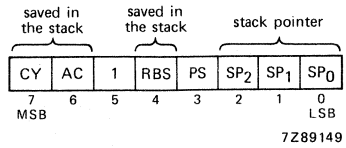



Fig. 16 Program status word.

All bits can be read using the MOV A, PSW instruction. Bits 7 and 6 are set and cleared by CPU operation. Bit 4 can be changed by a SEL RB instruction, bit 3 by the MOV PSW, A instruction, and bits 0, 1 and 2 by the CALL, RET or RETR instructions and in the event of an interrupt. Bits 7, 6 and 4 are stored in the program counter stack during subroutine and interrupt calls. These bits are restored in the PSW with a RETR (return and restore) instruction which must be used at the end of an interrupt and can be used at the end of a normal subroutine. The RET instruction has no restore feature and cannot be used at the end of an interrupt.

**Program counter** (see Fig. 17).

A 12-bit program counter is used to facilitate 8 K bytes of ROM being addressed. The arrangement of the bits is shown in Figure 17. During an interrupt subroutine $PC_{11}$ and $PC_{12}$ are forced to logic 0. All 12 bits are saved in the stack during CALL and interrupt routines.

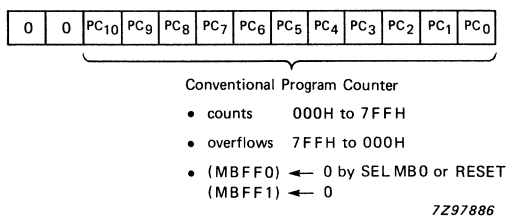



Fig. 17 Program counter.

**Central processing unit**

The PCD3347 has arithmetic, logical and branching capabilities. The DA A, SWAP A and XCHD instructions simplify BCD arithmetic and the handling of nibbles. The MOVP A,@A instruction permits efficient table look-up from the CURRENT ROM page.

## FUNCTIONAL DESCRIPTION (continued)

### Conditional branch logic

The conditional branch logic within the processor enables several conditions, internal and external to the processor, to be tested by the user's program.
Table 4 lists the conditional jump instructions used to change the program sequence. The DJNZ instruction decrements a designated register or data memory location and branches if the contents are not zero. This instruction is useful for looping control. The JMPP@A instruction allows multiway branches to destinations determined by the contents of the accumulator.

**Table 4** Conditional branches

| test | jump condition | jump instruction |
|---|---|---|
| accumulator | all bits zero | JZ |
| | any bit non-zero | JNZ |
| accumulator bit test | 1 | JB0 to JB7 |
| carry flag | 1 | JC |
| | 0 | JNC |
| timer overflow flag | 1 | JTF |
| | 0 | JNTF |
| test input T0 | 1 | JNT0 |
| | 0 | JT0* |
| test input T1 | 1 | JT1 |
| | 0 | JNT1 |
| register | non-zero | DJNZ |

* Because of the inverted interrupt input CE/$\overline{T0}$ the conditional jump JT0 is also inverted.

**Test input T1** (pin 8)

The T1 input line can be used as:

- A test input for branch instructions JT1 and JNT1
- An external input to the event counter

When used as a test input:

- JT1 instruction tests for logic 1 level
- JNT1 instruction tests for logic 0 level

When used as an input to the event counter, T1 must be LOW for $>$ 4 CP, followed by a HIGH for $>$ 4 CP. The transition can be recognized with a repetition rate of once per 30 oscillator clock periods (1 machine cycle).

There is no internal pull-up or pull-down resistor connected to the T1 input. If required it must be externally connected to a resistor ($R = \leqslant 100\ k\Omega$).
When T1 is not used pin 8 must be connected to $V_{DD}$ or $V_{SS}$.

**Reset** (pin 11)

A positive-going signal on the RESET input/output:

- Sets the program counter to zero
- Selects location 0 of memory bank 0 and register bank 0
- Sets the stack pointer to zero (000); pointing to RAM address 8
- Disables the interrupts (external and timer)
- Stops the timer/event counter, then sets it to zero
- Sets the timer prescaler to modulo-32
- Resets the timer flag
- Sets all ports according to reset states
- Cancels IDLE and STOP mode

After the voltage is applied to RESET an internal delay of 1866 CP is introduced before the microcontroller commences operation.

**Power-on reset**

The internal power-on reset circuit monitors the supply voltage $V_{DD}$. As long as $V_{DD}$ remains below the internal reference level $V_{ref}$ (typically 1,3 V), the oscillator is inhibited and RESET (pin 8) has an undefined level. When $V_{DD}$ rises above $V_{ref}$, the oscillator is released and RESET is pulled HIGH to $V_{DD}$ by TR1 for a period $t_D$ (typically 50 $\mu$s). Note that the start-up time of the oscillator is typically 10 ms because of the narrow bandwidth of the crystal.

Three modes of power-on reset are possible:

1. If $V_{DD}$ has a fast rise time, i.e. $V_{DD}$ reaches its minimum value before the RESET signal finishes ($t_D$), then no additional circuit is required (see Figs 18 and 19). Note that the first instruction is executed after the oscillator start-up time plus 1866 clock periods.
2. If $V_{DD}$ has a slow rise time then the RESET signal should be stretched by an external CR circuit (see Figs 20 and 21). In the event of a short drop in $V_{DD}$, the diode path discharges the capacitor rapidly to ensure a reliable power-on reset. The RESET signal should reach at least 70% of the final value of $V_{DD}$ to ensure a correct reset. Given that the RESET voltage and $V_{DD}$ rise exponentially, the above requirement is satisfied when the time constant of the RESET pulse is $>$ 8 times the time constant of $V_{DD}$. If $V_{DD}$ rises linearly then a RESET time constant $>$ 2 times the rise time of $V_{DD}$ is required.

   When a reset is completed (RESET goes LOW) before the oscillator has started up, program execution begins after the oscillator start-up time plus 1866 clock periods (see Fig. 21). If the oscillator starts up prior to the completion of RESET then program execution begins 1866 clock periods after RESET goes LOW.
3. Fig. 22 shows an external reset applied during power-on. The external reset signal must remain HIGH until $V_{DD}$ has reached its minimum operating value corresponding to the selected oscillator frequency. When a reset is completed (RESET goes LOW) before the oscillator has started up, program execution begins after the oscillator start-up time plus 1866 clock periods (see Fig. 23). If the oscillator starts up prior to the completion of RESET then program execution begins 1866 clock periods after RESET goes LOW.

**FUNCTIONAL DESCRIPTION** (continued)

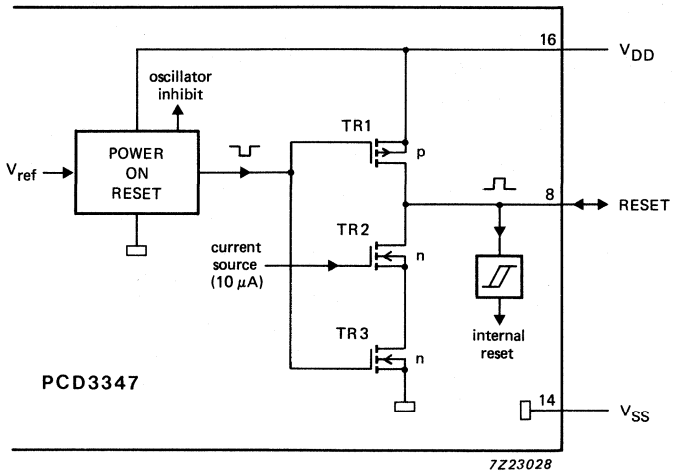



Fig. 18 Power-on reset configuration.

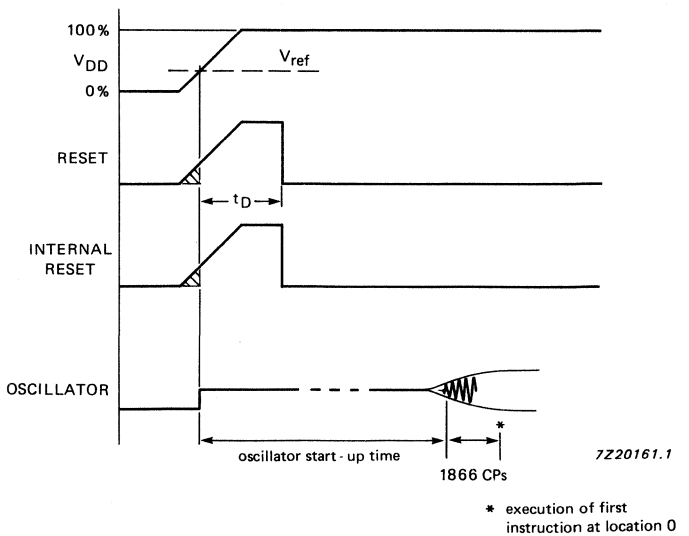



Fig. 19 Timing of power-on reset with fast rise time of $V_{DD}$.

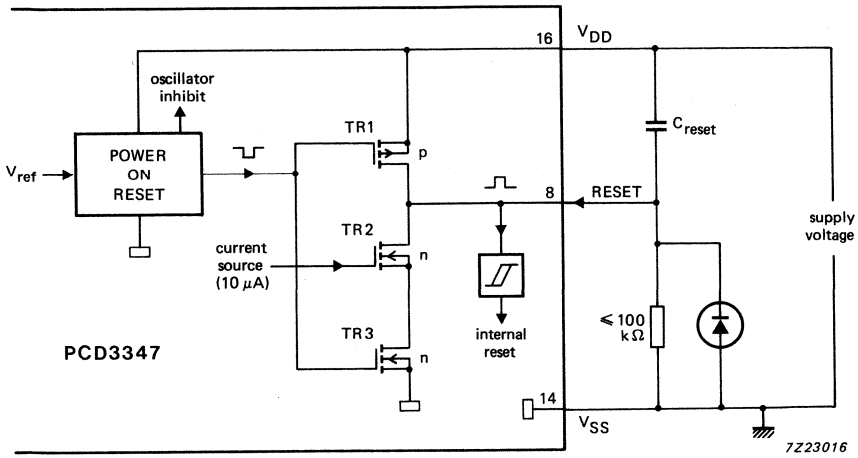



Fig. 20 Stretched power-on reset with external CR circuit.

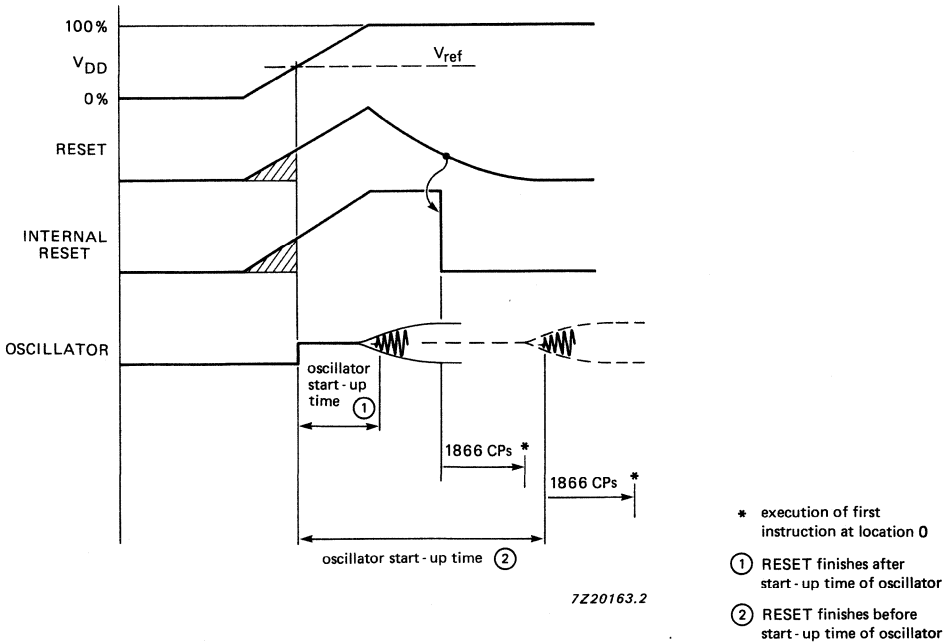



Fig. 21 Timing of power-on reset with a slowly rising $V_{DD}$ and a stretched RESET pulse.

## FUNCTIONAL DESCRIPTION (continued)

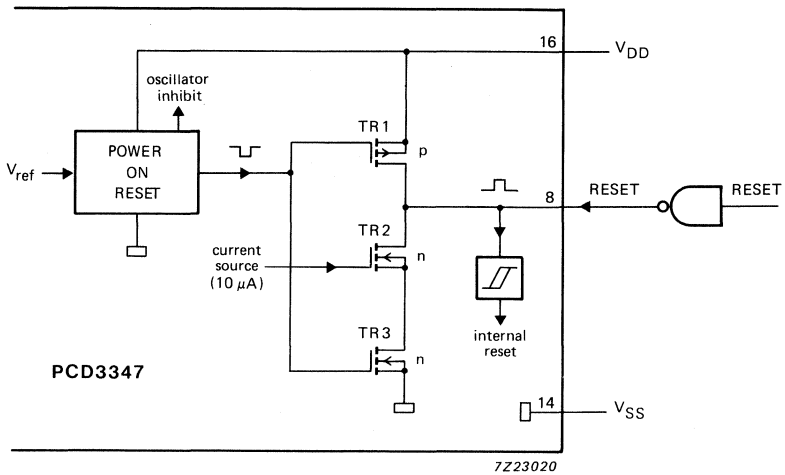



Fig. 22 External power-on reset configuration.

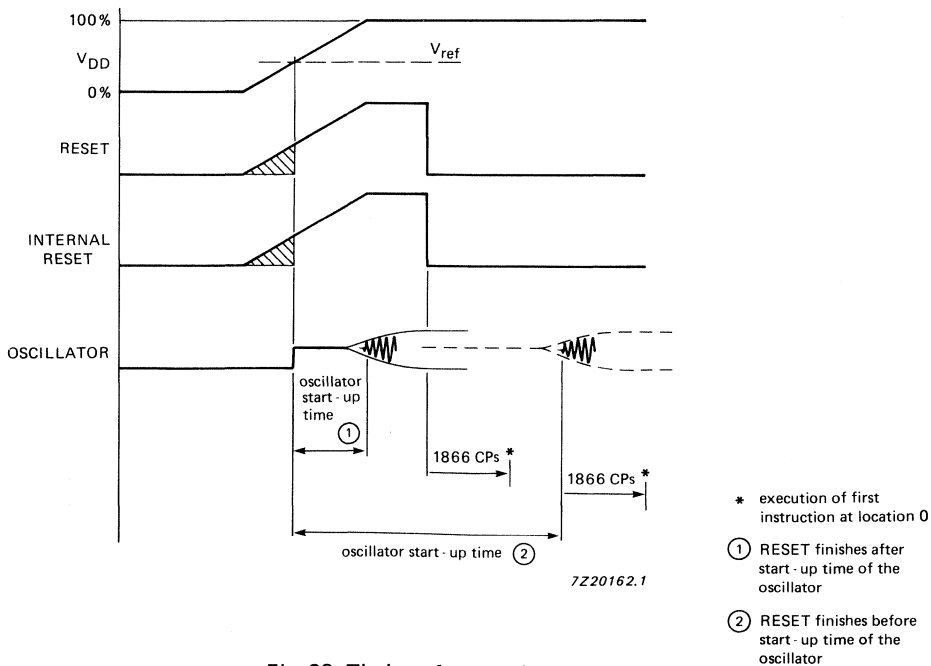



Fig. 23 Timing of external power-on reset.

## RATINGS

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

| parameter | conditions | symbol | min. | max. | unit |
|---|---|---|---|---|---|
| Supply voltage | pin 16 | $V_{DD}$ | −0,8 | +8 | V |
| Input voltage | any pin | $V_I$ | −0,8 | $V_{DD}$ + 0,8 | V |
| DC current | any input or output | $\pm I_I, \pm I_O$ | – | 10 | mA |
| Total power dissipation | derate according to thermal resistance | $P_{tot}$ | – | 500 | mW |
| Power dissipation | per output | $P_O$ | – | 50 | mW |
| Storage temperature range | | $T_{stg}$ | −65 | +150 | °C |
| Operating ambient temperature range | | $T_{amb}$ | −25 | +70 | °C |
| Operating junction temperature | | $T_j$ | – | +125 | °C |
| Thermal resistance junction to ambient | SOT146 | $R_{th\ j\text{-}a}$ | – | 120 | K/W |
| | SOT163A | $R_{th\ j\text{-}a}$ | – | 150 | K/W |

## DC CHARACTERISTICS

$V_{DD}$ = 2,5 to 6 V; $V_{SS}$ = 0 V; $T_{amb}$ = −25 to + 70 °C; all voltages with respect to $V_{SS}$; f = 3,58 MHz with $R_S$ = 100 Ω; unless otherwise specified

| parameter | conditions | symbol | min. | typ. | max. | unit |
|---|---|---|---|---|---|---|
| **Supplies** | | | | | | |
| Supply voltage | | | | | | |
| operating | | $V_{DD}$ | 2,5 | – | 6 | V |
| STOP mode for RAM data retention | | $V_{DD}$ | 1,0 | – | 6 | V |
| Supply current (Fig. 25) | | | | | | |
| operating with tone generator on | $V_{DD}$ = 3 V | $I_{DD}$ | – | 800 | – | μA |
| operating without tone generator | $V_{DD}$ = 3 V | $I_{DD}$ | – | 400 | – | μA |
| IDLE mode (Fig. 26) | | | | | | |
| with tone generator on | $V_{DD}$ = 3 V | $I_{DD}$ | – | 600 | – | μA |
| without tone generator | $V_{DD}$ = 3 V | $I_{DD}$ | – | 200 | – | μA |
| STOP mode (Fig. 27) | note 1; $V_{DD}$ = 1,8 V | | | | | |
| | $T_{amb}$ = 25 °C | $I_{DD}$ | – | 1,5 | 2,0 | μA |
| | $T_{amb}$ = 55 °C | $I_{DD}$ | – | – | 5 | μA |
| | $T_{amb}$ = 70 °C | $I_{DD}$ | – | – | 10 | μA |
| **RESET I/O** | | | | | | |
| Switching level | | $V_{RESET}$ | – | 1,3 | – | V |
| Sink current | $V_{DD} > V_{RESET}$ | $I_{OL}$ | – | 7 | – | μA |
| **Inputs** | | | | | | |
| Input voltage LOW | | $V_{IL}$ | 0 | – | 0,3 $V_{DD}$ | V |
| Input voltage HIGH | | $V_{IH}$ | 0,7 $V_{DD}$ | – | $V_{DD}$ | V |
| Input leakage current | $V_{SS} < V_I < V_{DD}$ | $\pm I_{IL}$ | – | – | 1 | μA |
| **Outputs** | | | | | | |
| Output voltage LOW | $V_I$ = $V_{SS}$ or $V_{DD}$; $\lvert I_O \rvert < 1$ μA | $V_{OL}$ | – | – | 0,05 | V |
| Output sink current LOW for all remaining ports | $V_{DD}$ = 3 V; $V_O$ = 0,4 V | $I_{OL}$ | 1,5 | 2,5 | – | mA |
| Pull-up output source current HIGH (Fig. 29) | $V_{DD}$ = 3 V; $V_O$ = 0,7 $V_{DD}$ | $-I_{OH}$ | 10 | – | – | μA |
| | $V_O$ = $V_{SS}$ | $-I_{OH}$ | – | – | 300 | μA |
| Push-pull output source current HIGH | $V_{DD}$ = 3 V; $V_O$ = $V_{DD}$ −0,4 V | $-I_{OH}$ | 0,6 | 1,5 | – | mA |

**Note 1**

Crystal connected between XTAL1 and XTAL2; CE and T1 at $V_{SS}$.

## TONE GENERATOR CHARACTERISTICS

$V_{DD}$ = 2,5 to 6 V; $V_{SS}$ = 0 V; $T_{amb}$ = −25 to + 70 °C; all voltages with respect to $V_{SS}$; f = 3,58 MHz with $R_S$ = 100 Ω; unless otherwise specified

| parameter | conditions | symbol | min. | typ. | max. | unit |
|---|---|---|---|---|---|---|
| **Tone output** (Fig. 24) | | | | | | |
| DTMF output voltage levels (r.m.s. values) | | | | | | |
| HIGH group | | $V_{HG(rms)}$ | 158 | 192 | 205 | mV |
| LOW group | | $V_{LG(rms)}$ | 125 | 150 | 160 | mV |
| Frequency deviation | | $\Delta f/f$ | −0,6 | – | + 0,6 | % |
| DC voltage level | | $V_{dc}$ | – | ½ $V_{DD}$ | – | V |
| Output impedance | | $\|Z_O\|$ | – | 0,1 | 0,5 | kΩ |
| Load resistance | | $R_L$ | 10 | – | – | kΩ |
| Pre-emphasis of group | | $\Delta V_G$ | 1,85 | 2,1 | 2,35 | dB |
| Total harmonic distortion | note 2; $T_{amb}$ = 25 °C | THD | – | −25 | – | dB |

**Note 2**

Related to the level of the LOW group frequency component (CEPT CS 203)

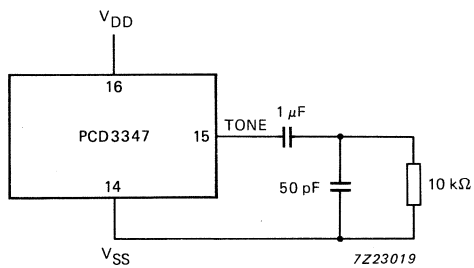



Fig. 24 Tone output test circuit.

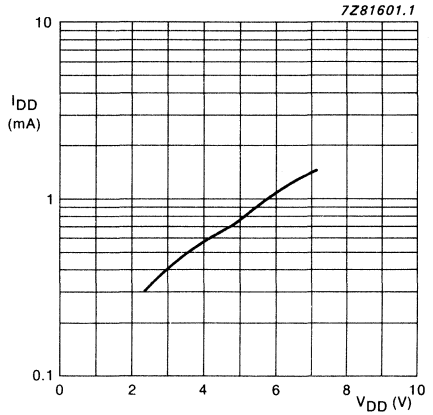



Fig. 25 Typical supply current ($I_{DD}$) in operating mode as a function of the supply voltage ($V_{DD}$); $T_{amb}$ = 25 °C; clock frequency = 3,58 MHz; $I_{DD}$ is increased by approximately 0,6 mA when the DTMF function is operating.

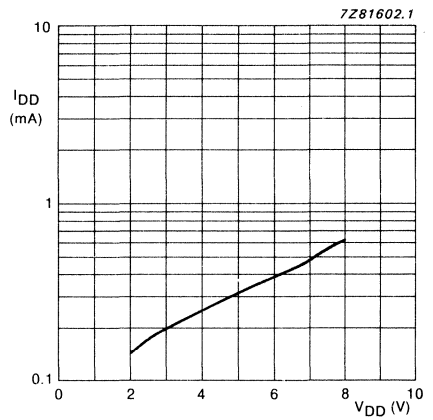
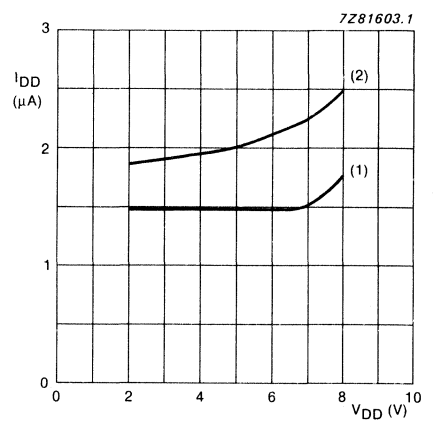
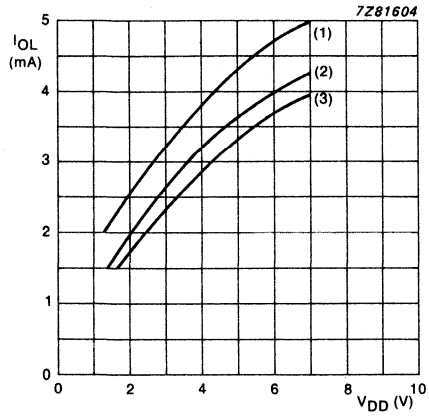



Fig. 26 Typical supply current ($I_{DD}$) in IDLE mode as a function of the supply voltage ($V_{DD}$); $T_{amb}$ = 25 °C; clock frequency = 3,58 MHz.



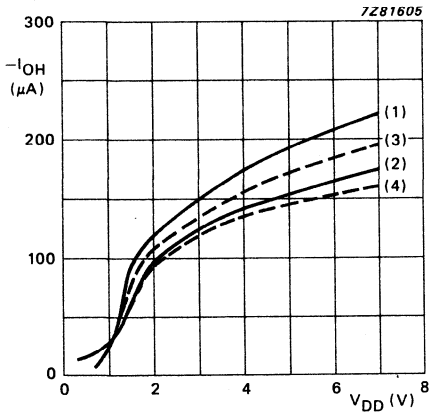

(1) $T_{amb}$ = 25 °C
(2) $T_{amb}$ = 70 °C

Fig. 27 Typical supply current ($I_{DD}$) in STOP mode as a function of the supply voltage ($V_{DD}$).




(1) $T_{amb} = -25\ ^{o}C$
(2) $T_{amb} = 25\ ^{o}C$
(3) $T_{amb} = 70\ ^{o}C$

Fig. 28 Output sink current LOW ($I_{OL}$), as a function of supply voltage ($V_{DD}$); $V_O = 0{,}4$ V.




(1) $T_{amb} = 25\ ^{o}C$; $V_O = V_{SS}$
(2) $T_{amb} = 25\ ^{o}C$; $V_O = 0{,}7\ V_{DD}$
(3) $T_{amb} = 70\ ^{o}C$; $V_O = V_{SS}$
(4) $T_{amb} = 70\ ^{o}C$; $V_O = 0{,}7\ V_{DD}$

Fig. 29 Output source current HIGH ($-I_{OH}$) as a function of supply voltage ($V_{DD}$).

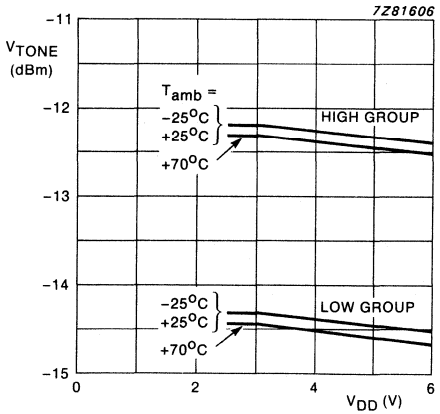



Fig. 30 DTMF output voltage levels as a function of operating supply voltage; $R_L$ = 1 MΩ.

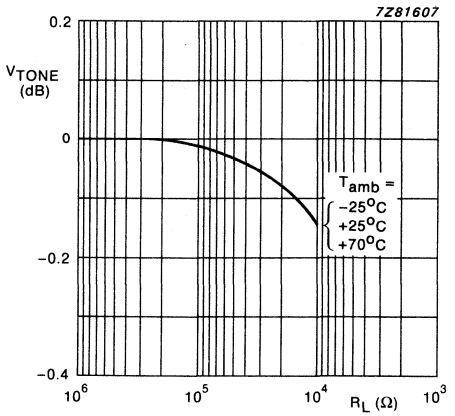



Fig. 31 Dual tone output voltage level as a function of output load resistance.

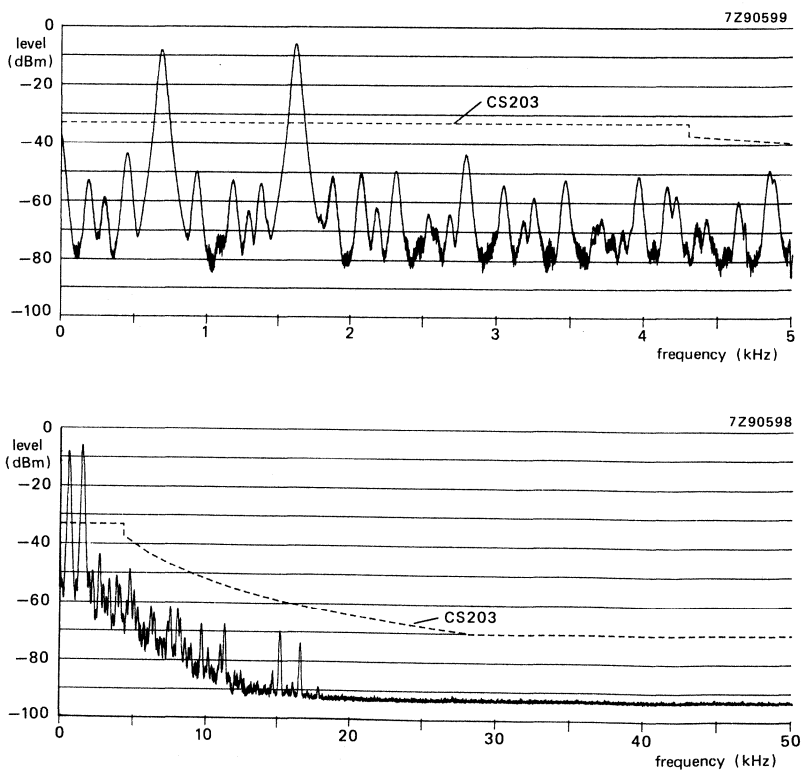



Fig. 32 Typical frequency spectrum of a dual tone signal after flat-band amplification of 6 dB.

## APPLICATION INFORMATION

A block diagram of an electronic featurephone built around the PCD3347 is shown in Figure 31. It comprises the following dedicated telephony ICs:

- TEA1060/1061/1067/1068 transmission circuit for telephony
- PCF8576 or PCF8577 2 LCD drivers in LCD module MB7020160
- PCF8571 1 K RAMs with Serial I/O; the number of RAMs depends on the required amount of stored telephone numbers
- PCD3360 programmable multi-tone ringer

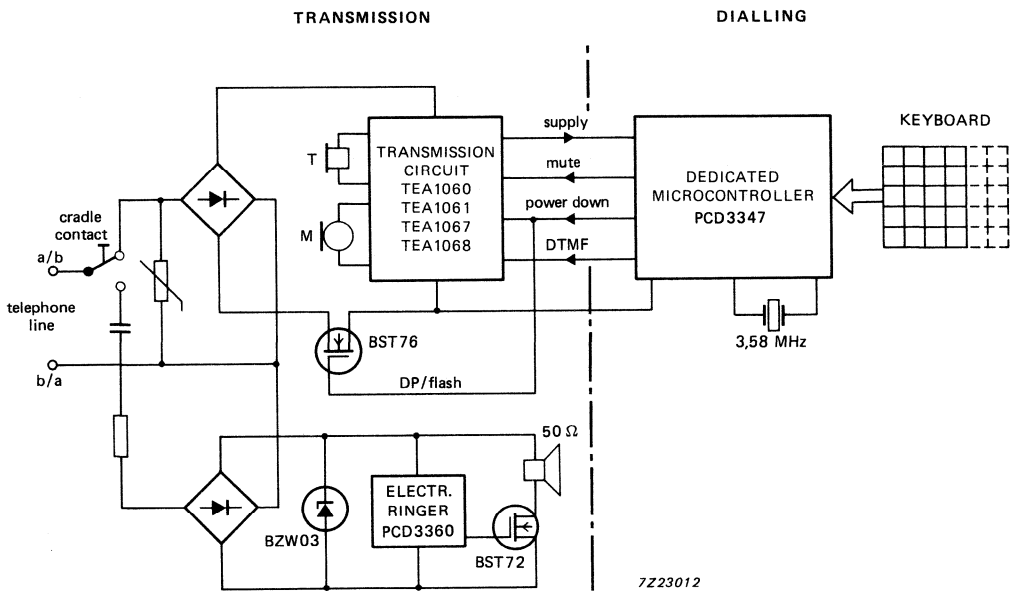


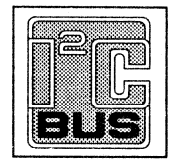

Fig. 33 Block diagram of electronic featurephone with common line interface.

# Single-chip 8-bit Telecom Microcontroller

# PCD3350A

**FOR DETAILED INFORMATION SEE THE LATEST ISSUE OF HANDBOOK IC03 OR DATASHEET**

## FEATURES



- 8-bit CPU, ROM, EEPROM, RAM, I/O in a 44-lead quad flat pack
- 8k ROM bytes; 256 RAM bytes
- 256 EEPROM bytes
- 32 kHz crystal oscillator for real-time clock
- EEPROM programmable real-time clock
- Over 100 instructions (based on MAB8048) all of 1 or 2 cycles
- 34 quasi-bidirectional I/O port lines
- 8-bit programmable timer/event counter 1
- 8-bit reloadable timer 2
- 3 single-level vectored interrupts: external, 8-bit programmable timer/event counter 1, derivative (triggered by reloadable timer 2)
- Two test inputs, one of which also serves as the external interrupt input
- DTMF tone generator
- Reference for supply and temperature-independent tone output
- Filtering for low output distortion (CEPT CS203 compatible)
- Melody output for ringer application
- Programmable DTMF clock divider
- Power-on reset
- Stop and idle modes
- Logic supply $V_{DD}$: 1.8 V to 6 V (DTMF tone output and EEPROM erase/write from 2.5 V)
- CPU Clock frequency: 1 MHz to 16 MHz (3.58 MHz or 10.74 MHz for DTMF)
- Operating temperature range: –25 °C to 70 °C
- Manufactured in silicon gate CMOS process

## GENERAL DESCRIPTION

This data sheet details the specific properties of the PCD3350A. The shared characteristics of the PCD33xxA family of microcontrollers are described in the PCD33xxA family data sheet, which should be read in conjunction with this publication.

The PCD3350A is a microcontroller oriented towards telephony applications. It includes 8k ROM bytes, 256 RAM bytes, 34 I/O lines, and an on-chip dual tone multi-frequency (DTMF) generator. In addition to dialling, the generated frequencies can be made available as square waves on port line P1.7/MDY for melody generation, providing ringer operation. The PCD3350A also incorporates 256 bytes of electrically erasable programmable read-only memory (EEPROM), permitting data storage without battery back-up. The EEPROM can be used for storing telephone numbers. Finally, the PCD3350A includes a low power 32 kHz crystal oscillator with an EEPROM programable real time clock (RTC) working in standby mode. The instruction set is based on that of the MAB8048 and is software compatible with the PCD33xxA family.

## ORDERING INFORMATION

| EXTENDED TYPE NUMBER | PACKAGE | | | |
|---|---|---|---|---|
| | PINS | PIN POSITION | MATERIAL | CODE |
| PCD3350AH | 44 | QFP | plastic | SOT205AG |

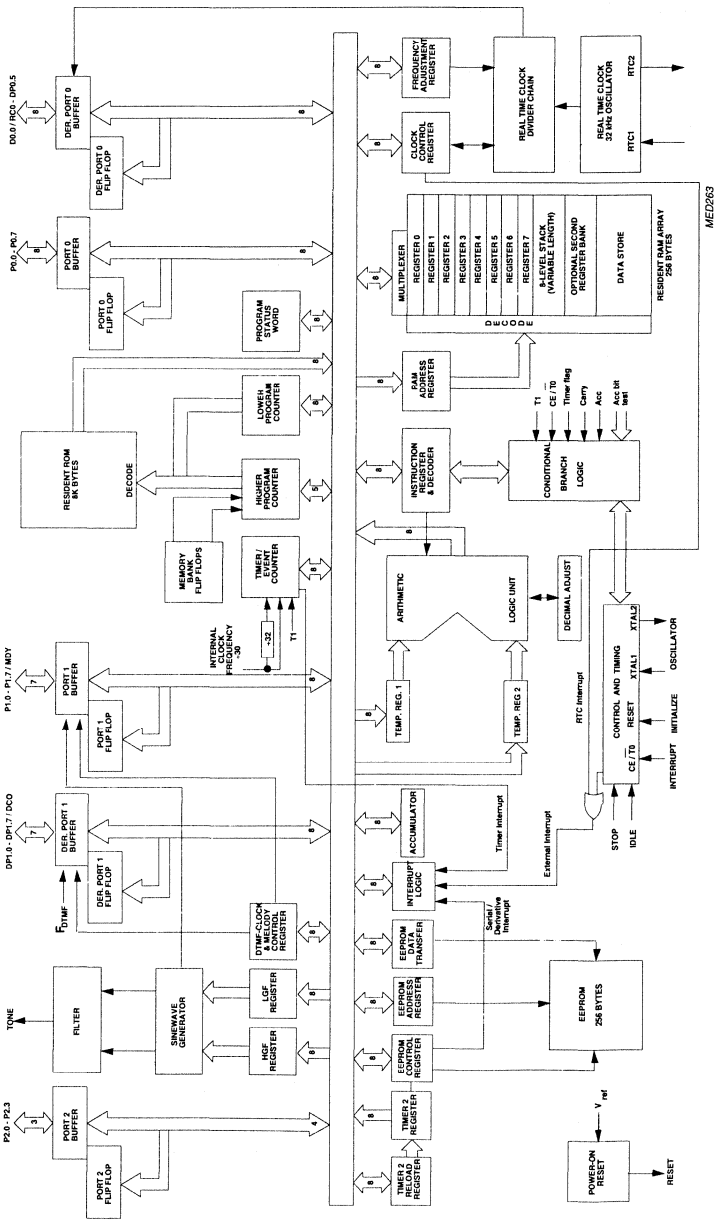



Fig.1 Block diagram.

## PINNING

| SYMBOL | PIN | DESCRIPTION |
|---|---|---|
| P2.1-P2.3 | 1-3 | **Port 2:** quasi-bidirectional I/O lines |
| DP0.0 / RCO | 4 | **Derivative port 0:** quasi bidirectional I/O line / Real Time Clock (RTC) |
| DP0.1 - DP0.5 | 5-9 | **Derivative port 0:** quasi bidirectional I/O lines |
| RTC1 | 10 | **RTC:** 32 kHz oscillator input |
| RTC2 | 11 | **RTC:** 32 kHz oscillator output |
| CE / T0N | 12 | **Enable:** Chip Enable / Test 0 |
| T1 | 13 | **Test 1**: count input of 8-bit timer / event counter 1 |
| RESET | 14 | **Reset:** reset input |
| DP1.0 - DP1.6 | 15-21 | **Derivative port 1:** quasi bidirectional I/O lines |
| DP1.7 / DCO | 22 | **Derivative port 1:** quasi bidirectional I/O lines / DTMF clock output |
| P0.0 - P0.3 | 23-26 | **Port 0:** quasi bidirectional I/O lines |
| XTAL1 | 27 | **Crystal oscillator:** external clock input |
| XTAL2 | 28 | **Crystal oscillator:** output |
| P0.4 - P0.7 | 29-32 | **Port 0:** quasi bidirectional I/O lines |
| P1.0 - P1.2 | 33-35 | **Port 1:** quasi bidirectional I/O lines |
| $V_{SS}$ | 36 | **Ground** |
| TONE | 37 | **DTMF:** output |
| $V_{DD}$ | 38 | **Supply voltage** |
| P1.3 - P1.6 | 39-42 | **Port 1:** quasi bidirectional I/O lines |
| P1.7 / MDY | 43 | **Port 1:** quasi bidirectional I/O lines / melody output |
| P2.0 | 44 | **Port 2:** quasi bidirectional I/O line |

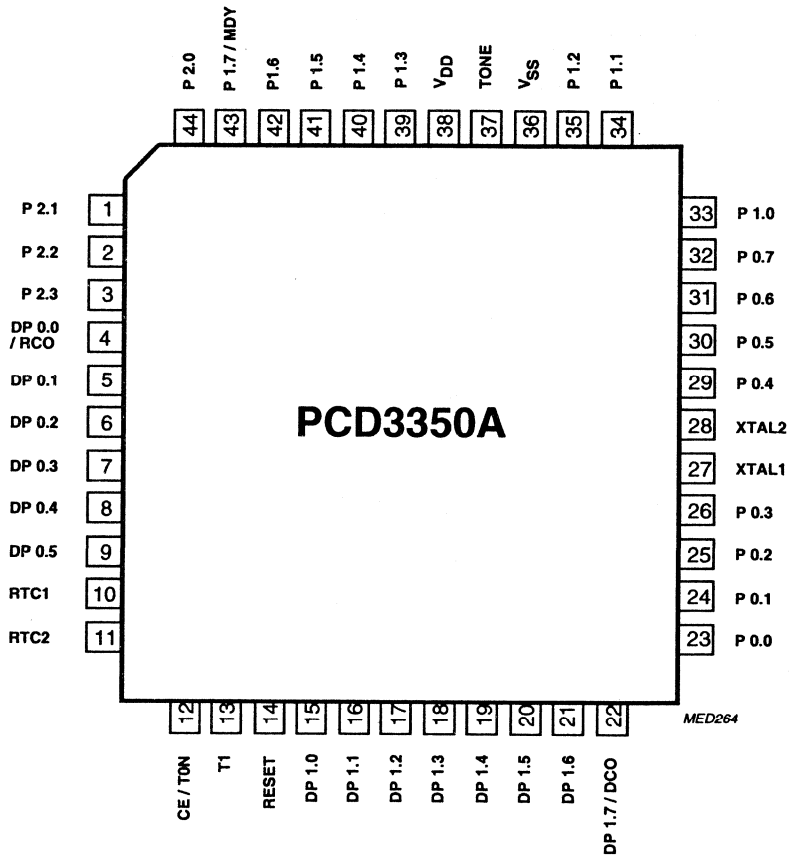



Fig.2 Pin configuration.

## PURCHASE OF PHILIPS I²C COMPONENTS

Purchase of Philips I²C components conveys a license under the Philips' I²C patent to use the components in the I²C system provided the system conforms to the I²C specification defined by Philips. This specification can be ordered using the code 9398 393 40011.

# Single-chip 8-bit Telecom Microcontroller

# PCD3351A/52A/53A

## FOR DETAILED INFORMATION SEE THE LATEST ISSUE OF HANDBOOK IC03 OR DATASHEET

### FEATURES

- 8-bit CPU, ROM, RAM, I/O in a single 28-lead package
- 2 k ROM bytes; 64 RAM bytes (PCD3351A)
- 4 k ROM bytes; 128 RAM bytes (PCD3352A)
- 6 k ROM bytes; 128 RAM bytes (PCD3353A)
- 128 EEPROM bytes
- Over 100 instructions (based on MAB8048) all of 1 or 2 cycles
- 20 quasi-bidirectional I/O port lines
- One 8-bit programmable timer/event counter
- Two 8-bit reloadable timers
- 3 single-level vectored interrupts: external, 8-bit programmable timer/event counter 1, derivative (triggered by reloadable timer 2)
- Two test inputs, one of which also serves as the external interrupt input
- DTMF tone generator
- Reference for supply and temperature-independent tone output
- Filtering for low output distortion (CEPT CS203 compatible)
- Melody output for ringer application
- Power-on reset
- Stop and idle modes
- Logic supply from 1.8 V to 6 V (DTMF tone output and EEPROM erase/write from 2.5 V)
- Clock frequency from 1 MHz to 16 MHz (3.58 MHz for DTMF suggested)
- Manufactured in silicon gate CMOS process

### GENERAL DESCRIPTION

This data sheet details the specific properties of the PCD3351A , PCD3352A and PCD3353A. The shared characteristics of the PCD33XXA family of microcontrollers are described in the PCD33XXA family data sheet, which should be read in conjunction with this publication.

The PCD3351A, PCD3352A and PCD3353A are microcontrollers orientated towards telephony applications which contain an on-chip dual tone multi-frequency (DTMF) generator. In addition to dialling, the generated frequencies can be made available as square waves on port line P1.7/MDY for melody generation thus providing ringer operation. The PCD3351A/52A/53A also incorporate 128 bytes of electrically erasable programmable read-only memory (EEPROM), permitting data storage without battery back-up. The EEPROM can be used for storing telephone numbers, particularly for implementing redial functions. The instruction set is based on that of the MAB8048 and is software compatible with the PCD33XXA family.

### LIMITING VALUES

In accordance with the Absolute Maximum System (IEC 134).

| SYMBOL | PARAMETER | MIN. | MAX. | UNIT |
|---|---|---|---|---|
| $V_{DD}$ | positive supply voltage | −0.8 | +7.0 | V |
| $V_I$ | all input voltages | −0.5 | $V_{DD}$+0.5 | V |
| $I_I$,$I_O$ | DC input or output current | −10 | +10 | mA |
| $I_{SS}$ | ground supply current | −50 | +50 | mA |
| $P_{tot}$ | total power dissipation | − | 125 | mW |
| $P_O$ | power dissipation per output | − | 30 | mW |
| $T_{stg}$ | storage temperature range | −55 | +150 | °C |
| $T_j$ | operating junction temperature | − | 90 | °C |

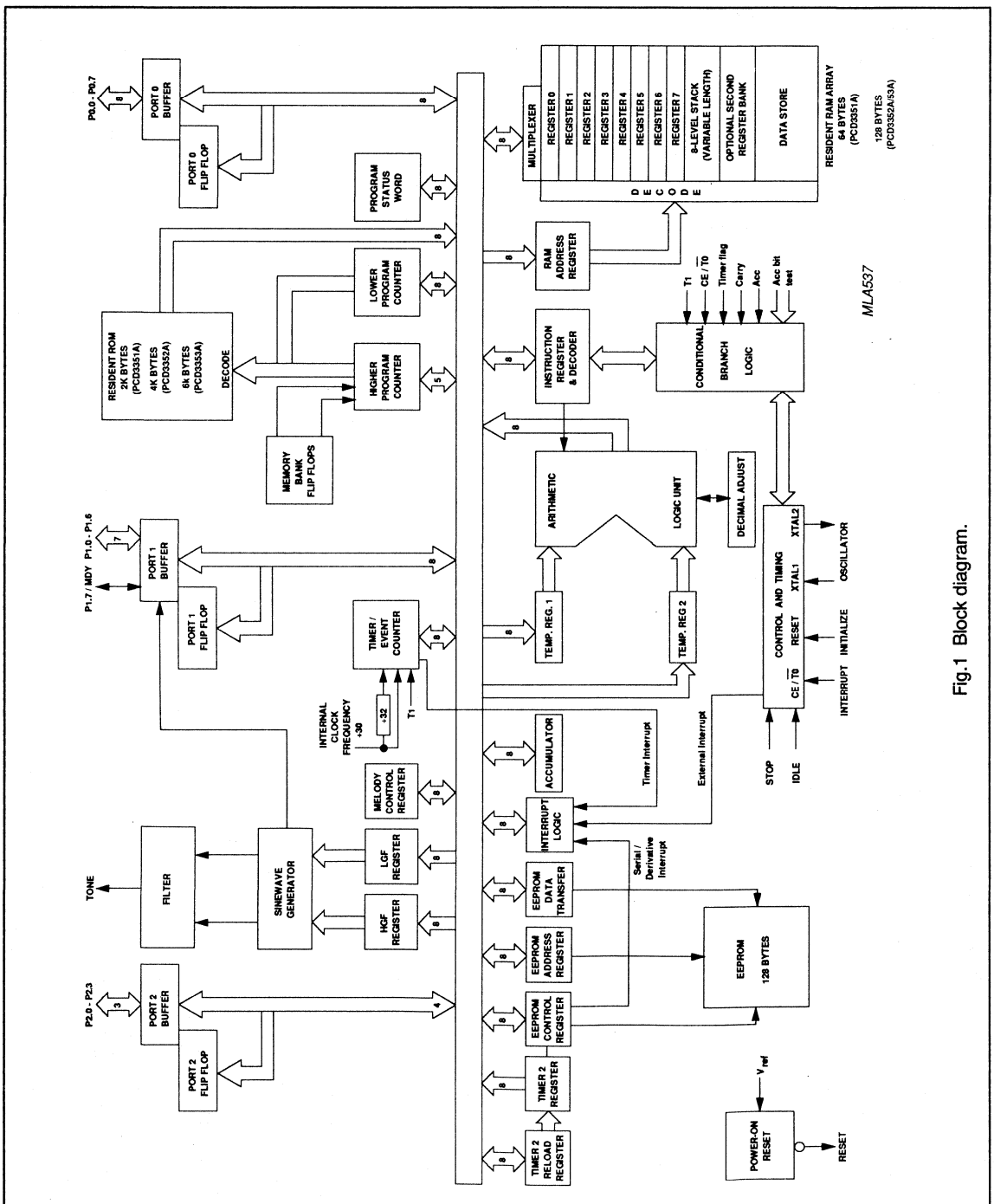



Fig.1 Block diagram.

## ORDERING INFORMATION

| EXTENDED TYPE NUMBER | PACKAGE | | | |
|---|---|---|---|---|
| | PINS | PIN POSITION | MATERIAL | CODE |
| PCD3351AP/52AP/53AP | 28 | DIL | plastic | SOT117 |
| PCD3351AT/52AT/53AT | 28 | mini-pack | plastic | SOT136A |

### Note to the Ordering Information

Full and up-to-date data for this device is available upon request via your Philips local sales office.

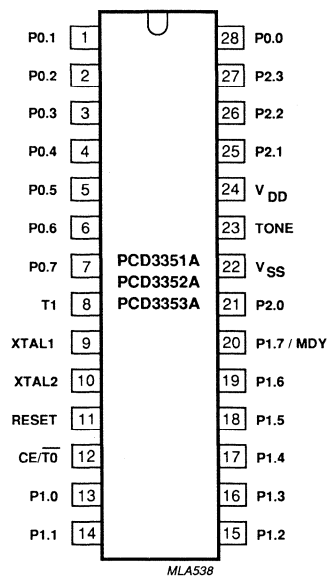



Fig.2 Pin configuration.

### PINNING

| SYMBOL | PIN | I/O | DESCRIPTION |
|---|---|---|---|
| P0.1-P0.7 | 1-7 | I/O | Port 0: quasi-bidirectional I/O lines |
| T1 | 8 | I | Test 1/count input of 8-bit timer/event counter 1 |
| XTAL1 | 9 | I | crystal oscillator/ external clock input |
| XTAL2 | 10 | O | crystal oscillator output |
| RESET | 11 | I | reset input |
| CE/$\overline{T0}$ | 12 | I | chip enable/test 0 |
| P1.0-P1.6 | 13-19 | I/O | Port 1: quasi-bidirectional I/O line |
| P1.7/MDY | 20 | I/O | Port 1: quasi-bidirectional I/O line/melody output |
| P2.0 | 21 | I/O | Port 2: quasi-bidirectional I/O line |
| $V_{SS}$ | 22 | P | ground |
| TONE | 23 | O | DTMF output |
| $V_{DD}$ | 24 | P | positive supply voltage |
| P2.1-P2.3 | 25-27 | I/O | Port 2: quasi-bidirectional I/O lines |
| P0.0 | 28 | I/O | Port 0: quasi-bidirectional I/O line |

## PURCHASE OF PHILIPS I²C COMPONENTS

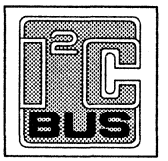
Purchase of Philips I²C components conveys a license under the Philips' I²C patent to use the components in the I²C system provided the system conforms to the I²C specification defined by Philips. This specification can be ordered using the code 9398 393 40011.

# Single-chip 8-bit Telecom Microcontroller with on-chip DTMF

# PCD3354A

**FOR DETAILED INFORMATION SEE THE LATEST ISSUE OF HANDBOOK IC03 OR DATASHEET**

## FEATURES




- 8-bit CPU, ROM, RAM, EEPROM, I/O in a 44-lead quad flat pack
- 8k ROM bytes; 256 RAM bytes
- 256 EEPROM bytes
- Over 100 instructions (based on MAB8048) all of 1 or 2 cycles
- 36 quasi-bidirectional I/O port lines
- 8-bit programmable timer/event counter 1
- 8-bit reloadable timer 2
- 3 single-level vectored interrupts: external, 8-bit programmable timer/event counter 1, derivative (triggered by reloadable timer 2)
- Two test inputs, one of which also serves as the external interrupt input
- DTMF tone generator
- Reference for supply and temperature-independent tone output
- Filtering for low output distortion (CEPT CS203 compatible)
- Melody output for ringer application
- Programmable DTMF clock divider
- Power-on reset
- Stop and idle modes
- Logic supply $V_{DD}$: 1.8 V to 6 V (DTMF tone output and EEPROM erase/write from 2.5 V)
- CPU Clock frequency: 1 MHz to 16 MHz (3.58 MHz, 10.74 MHz for DTMF)
- Operating temperature range: –25 °C to 70 °C
- Manufactured in silicon gate CMOS process

## GENERAL DESCRIPTION

This data sheet details the specific properties of the PCD3354A. The shared characteristics of the PCD33xxA family of microcontrollers are described in the PCD33xxA family data sheet, which should be read in conjunction with this publication.

The PCD3354A is a microcontroller oriented towards telephony applications. It includes 8k ROM bytes, 256 RAM bytes, 36 I/O lines, and an on-chip dual tone multi-frequency (DTMF) generator. In addition to dialling, the generated frequencies can be made available as square waves on port line P1.7/MDY for melody generation, providing ringer operation. The PCD3354A also incorporates 256 bytes of electrically erasable programmable read-only memory (EEPROM), permitting data storage without battery back-up. The EEPROM can be used for storing telephone numbers. The instruction set is based on that of the MAB8048 and is software compatible with the PCD33xxA family.

## ORDERING INFORMATION

| EXTENDED TYPE NUMBER | PACKAGE | | | |
|---|---|---|---|---|
| | PINS | PIN POSITION | MATERIAL | CODE |
| PCD3354AH | 44 | QFP | plastic | SOT205AG |

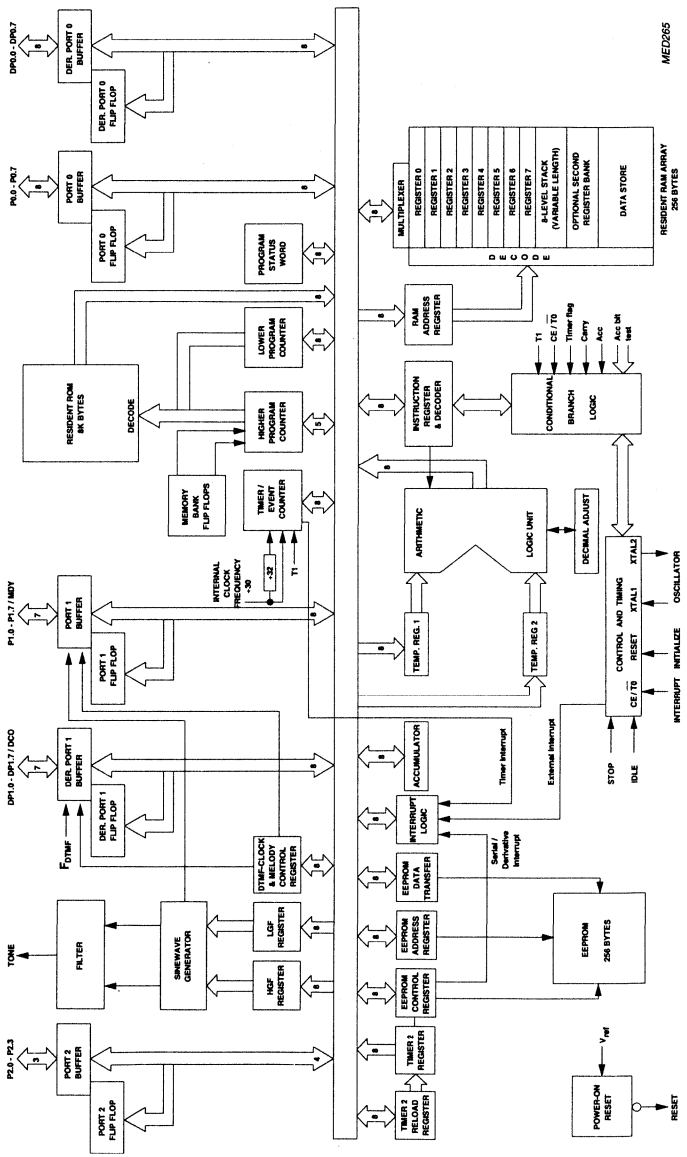



Fig.1 Block diagram.

## PINNING

| SYMBOL | PIN | DESCRIPTION |
|---|---|---|
| P2.1-P2.3 | 1-3 | **Port 2:** quasi-bidirectional I/O lines |
| DP0.0 - DP0.7 | 4-11 | **Derivative port 0:** quasi bidirectional I/O line |
| CE/T0N | 12 | **Enable:** Chip enable / Test 0 |
| T1 | 13 | **Test 1:** count input of 8-bit timer/event counter 1 |
| RESET | 14 | **Reset:** reset input |
| DP1.0 - DP1.6 | 15-21 | **Derivative port 1:** quasi bidirectional I/O line |
| DP1.7 / DCO | 22 | **Derivative port 1:** quasi bidirectional I/O line / DTMF clock input |
| P0.0 - P0.3 | 23-26 | **Port 0:** quasi-bidirectional I/O line |
| XTAL1 | 27 | **Crystal oscillator:** external clock input |
| XTAL2 | 28 | **Crystal oscillator:** output |
| P0.4 - P0.7 | 29-32 | **Port 0:** quasi bidirectional I/O lines |
| P1.0 - 1.2 | 33-35 | **Port 1:** quasi-bidirectional I/O lines |
| $V_{SS}$ | 36 | **Ground** |
| TONE | 37 | **DTMF:** output |
| $V_{DD}$ | 38 | **Supply voltage** |
| P1.3 - P1.6 | 39-42 | **Port 1:** quasi-bidirectional I/O lines |
| P1.7 / MDY | 43 | **Port 1:** quasi-bidirectional I/O line / melody output |
| P2.0 | 44 | **Port 2:** quasi bidirectional I/O line |

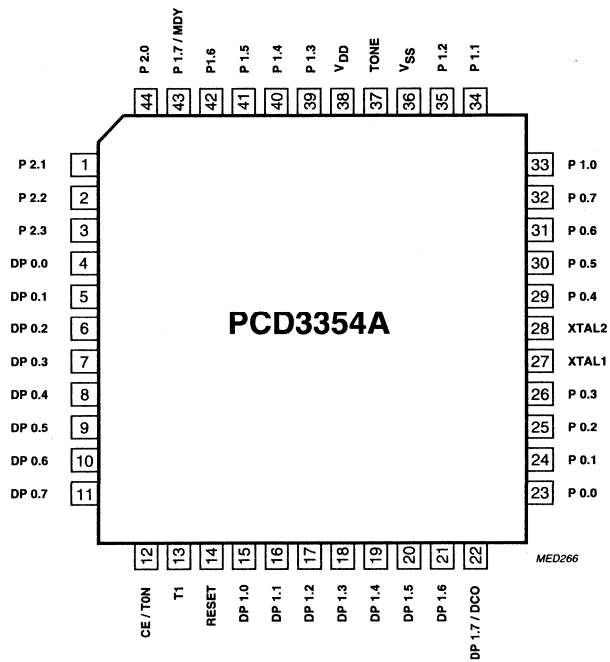


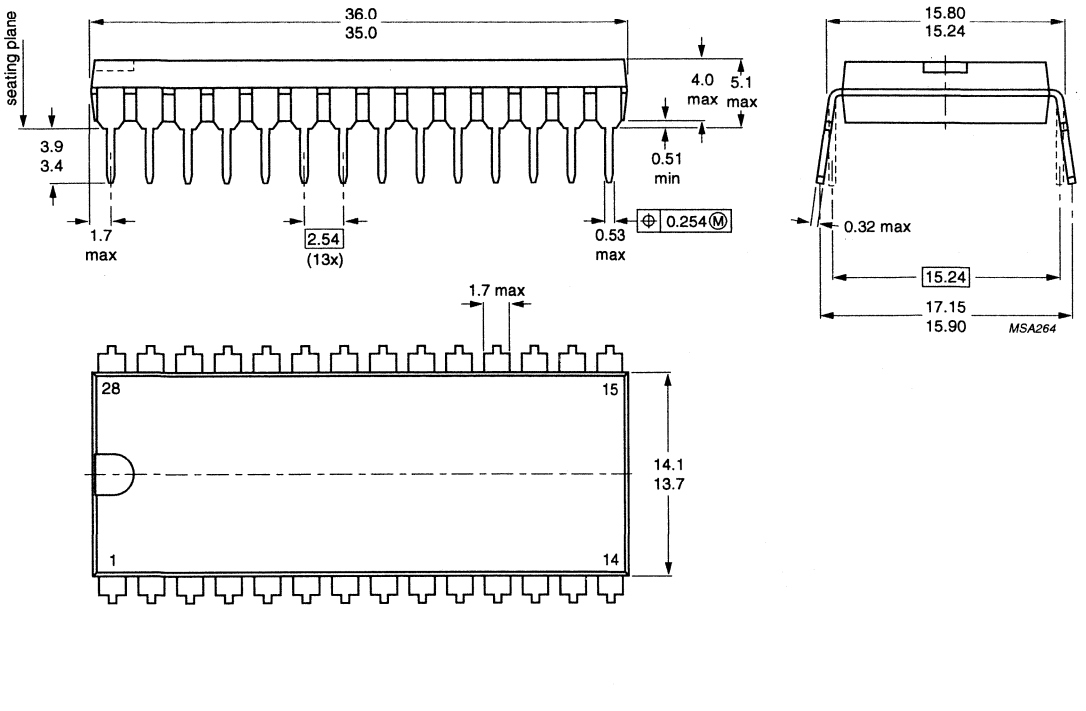

Fig.2 Pin configuration.

## PURCHASE OF PHILIPS I²C COMPONENTS

Purchase of Philips I²C components conveys a license under the Philips' I²C patent to use the components in the I²C system provided the system conforms to the I²C specification defined by Philips. This specification can be ordered using the code 9398 393 40011.

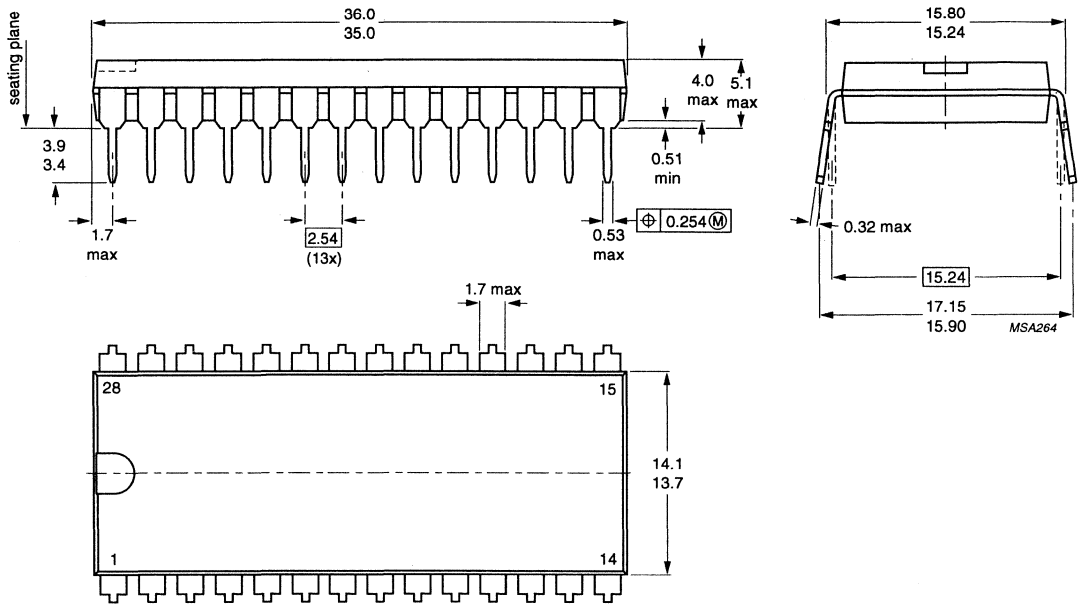
# SECTION 5 PACKAGE INFORMATION

## PACKAGE OUTLINES



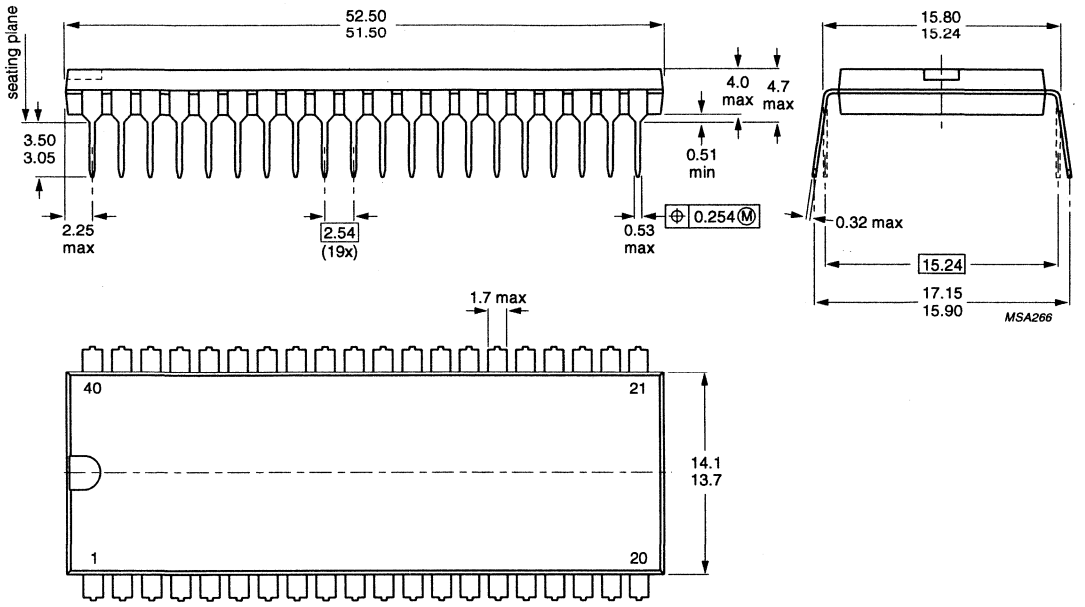

Dimensions in mm.

Fig.1 28-lead dual in-line; plastic (SOT117).



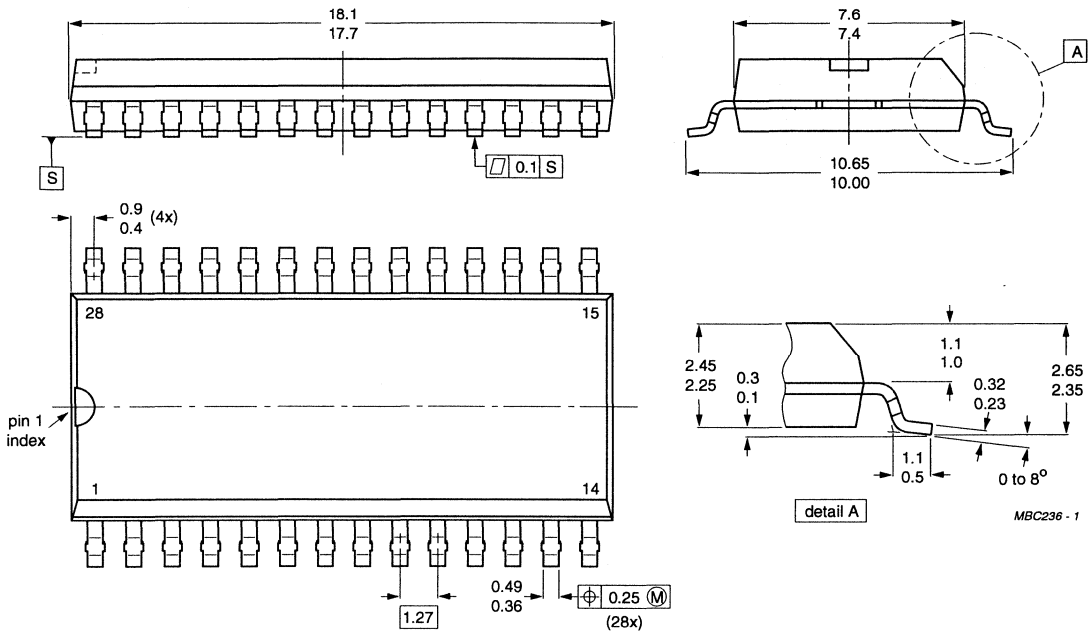

Dimensions in mm.

Fig.2 28-lead dual in line; plastic with internal heat spreader (SOT117).



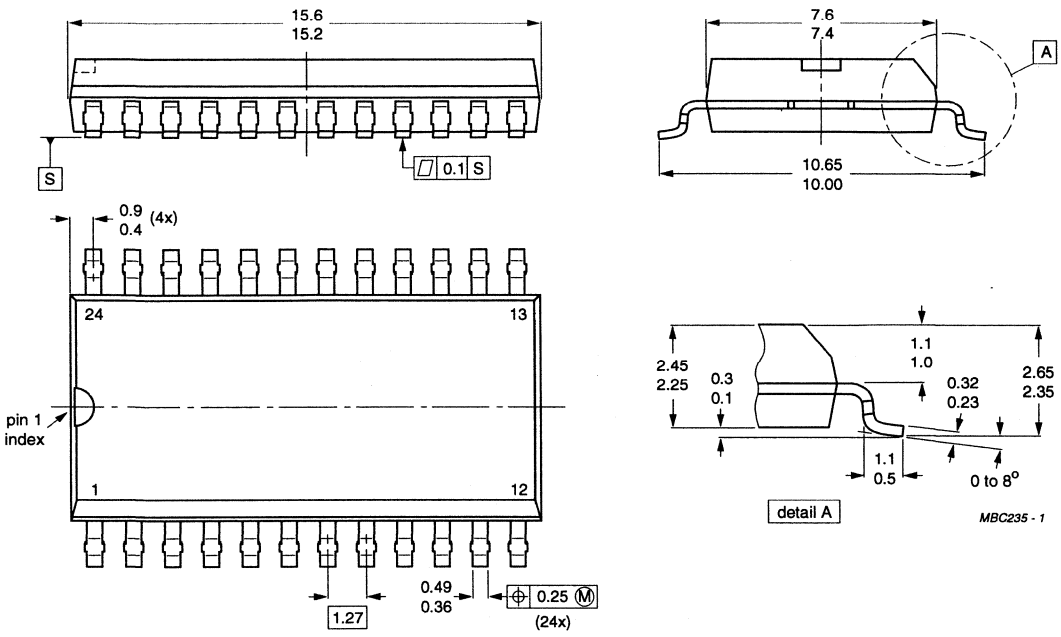

Dimensions in mm.

Fig.3 40-lead dual in-line; plastic (SOT129).



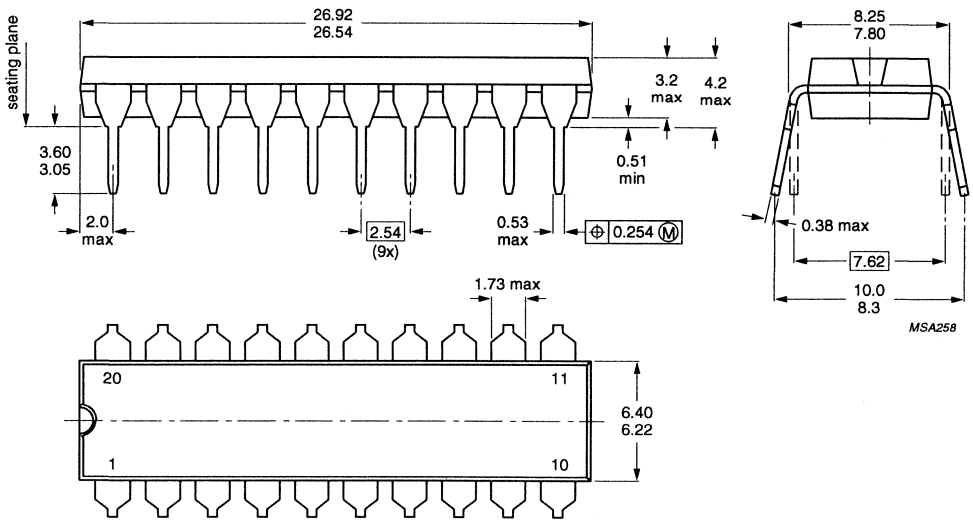

Dimensions in mm.

Fig.4 28-lead mini-pack; plastic (SO28; SOT136A).




Dimensions in mm.

Fig.5 24-lead small-outline; plastic (SO24L; SOT137A).




Dimensions in mm.

Fig.6 20-lead dual in-line; plastic (SOT146).

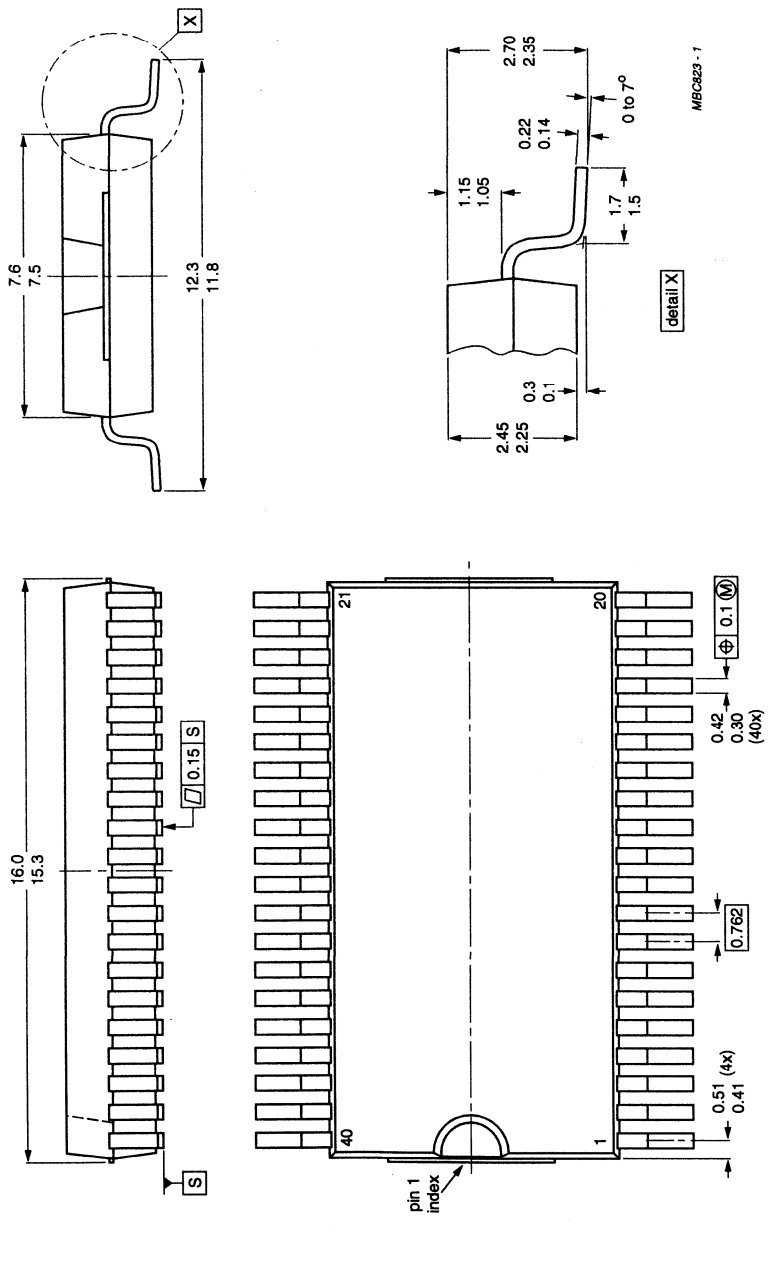
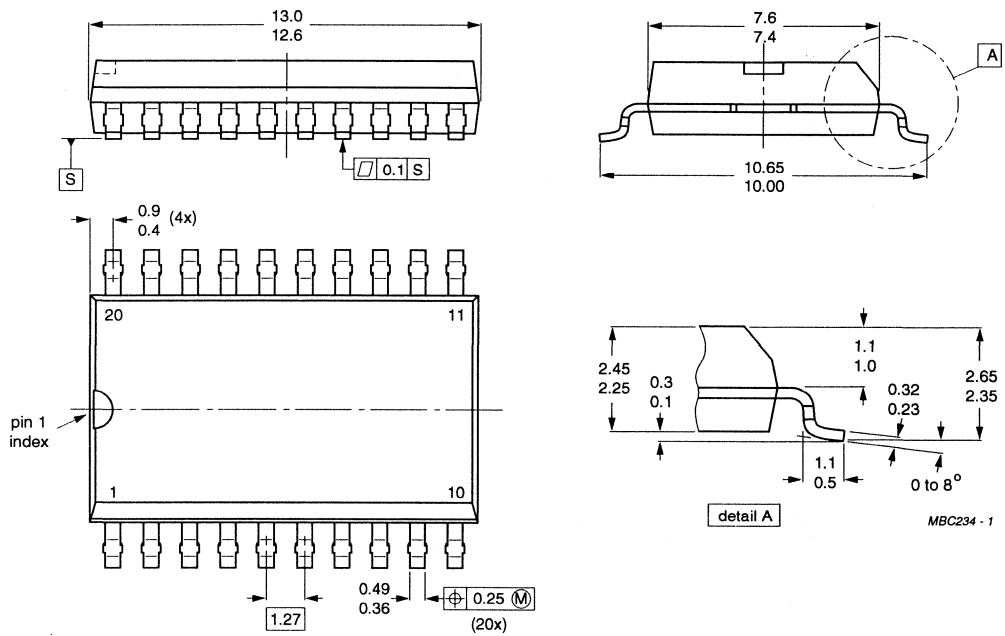


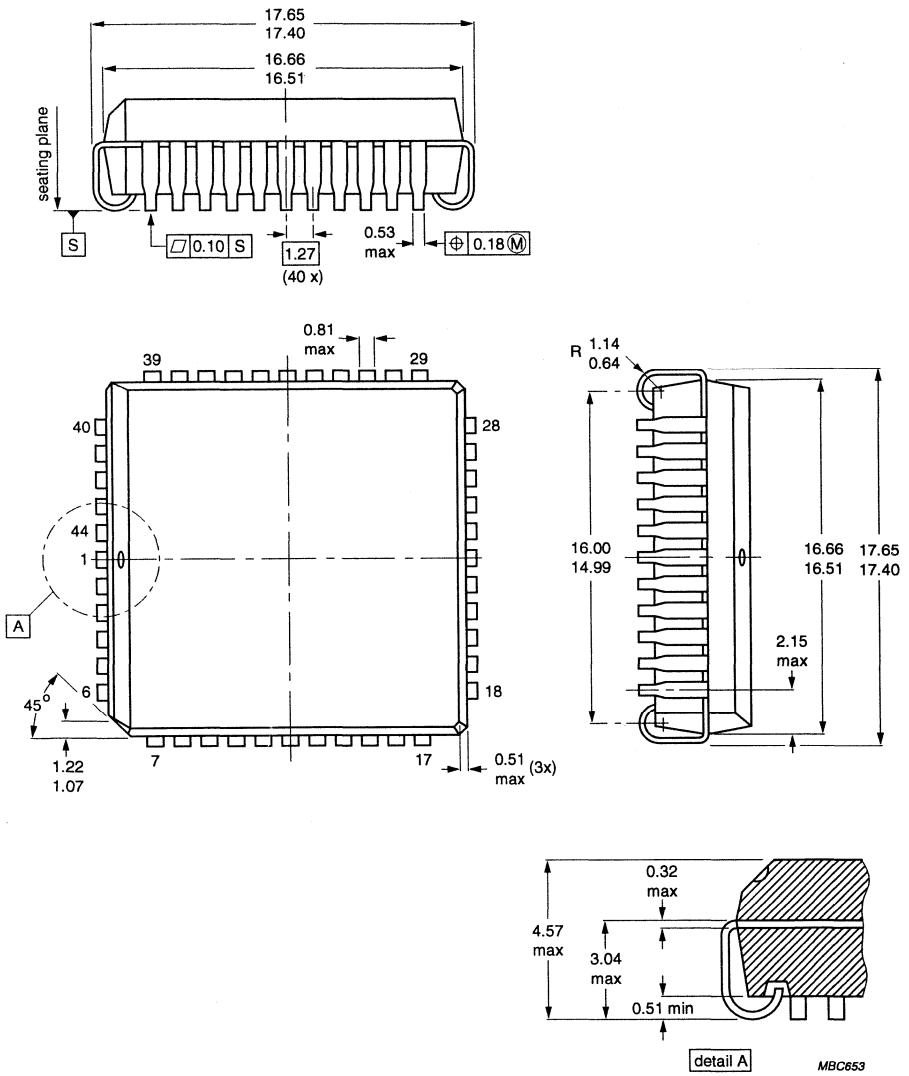

Fig.7 40-lead mini-pack; plastic (VSO40; SOT158A).

Dimensions in mm.



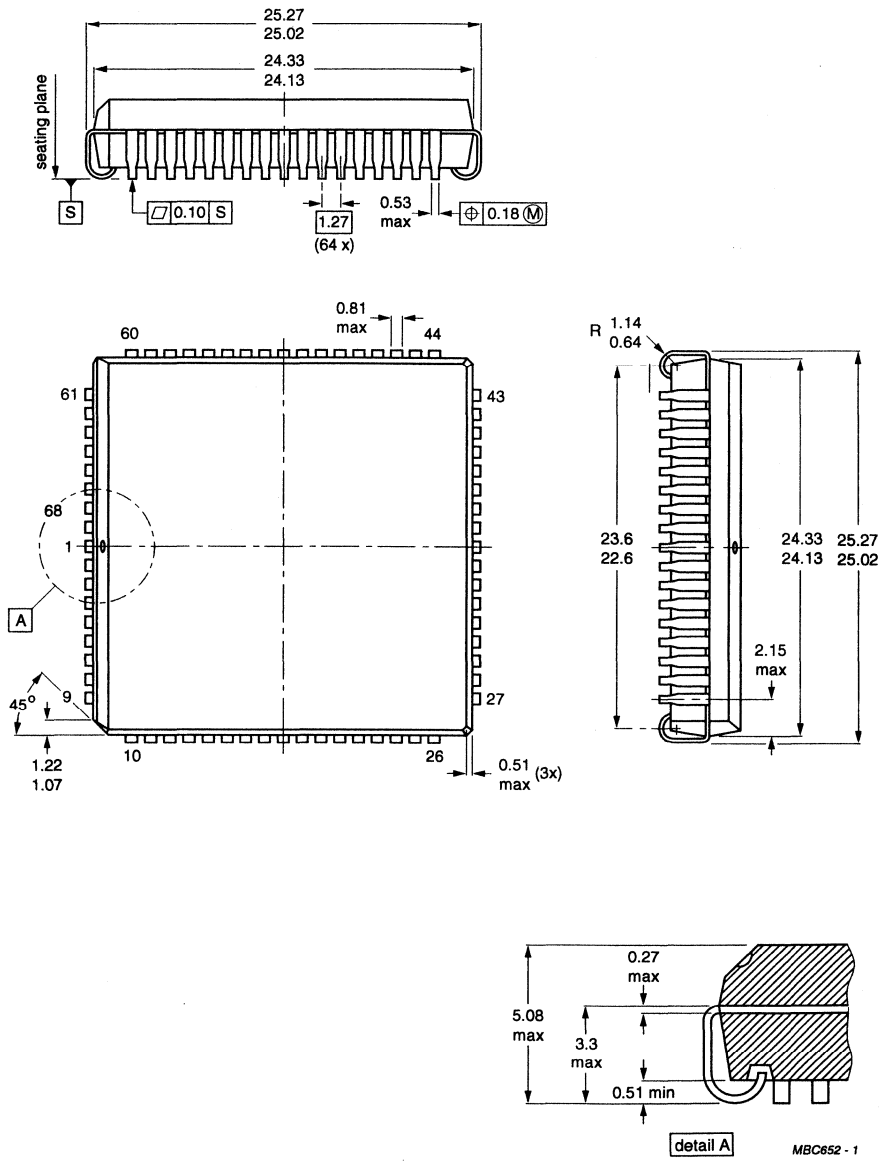

Dimensions in mm.

Fig.8 20-lead mini-pack; plastic (SO20; SOT163A).




Dimensions in mm.

Fig.9 44-lead plastic leaded chip carrier (PLCC); (SOT187AA).




Dimensions in mm.

Fig.10 68-lead plastic leaded chip carrier (PLCC); (SOT188AA, AGA).

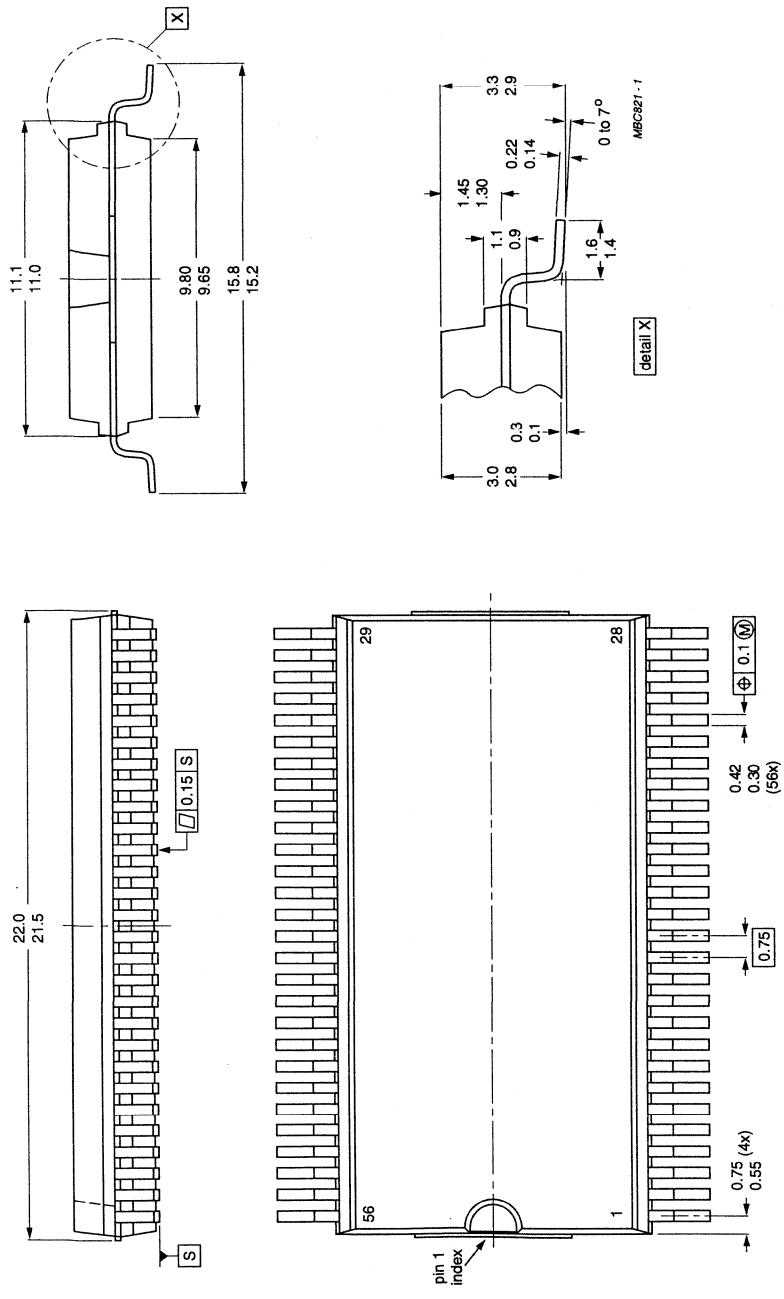
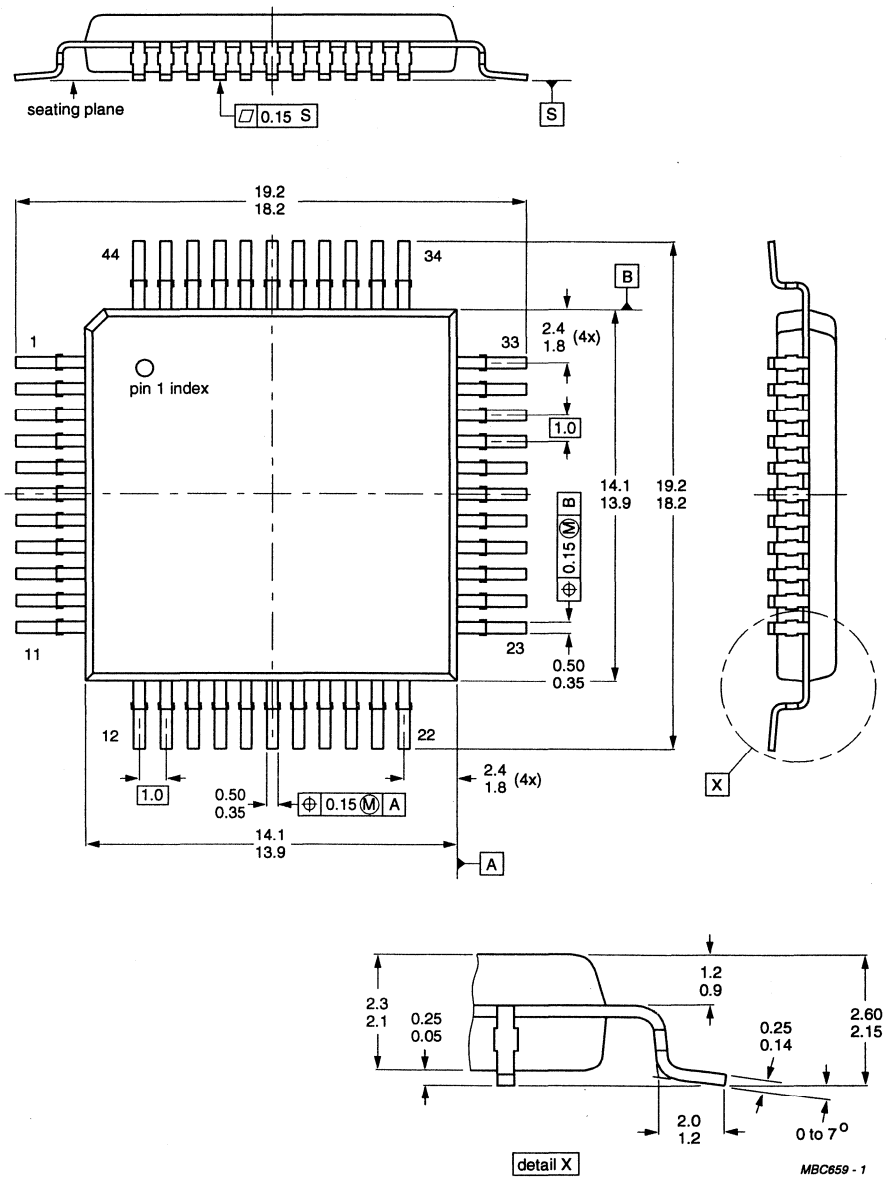


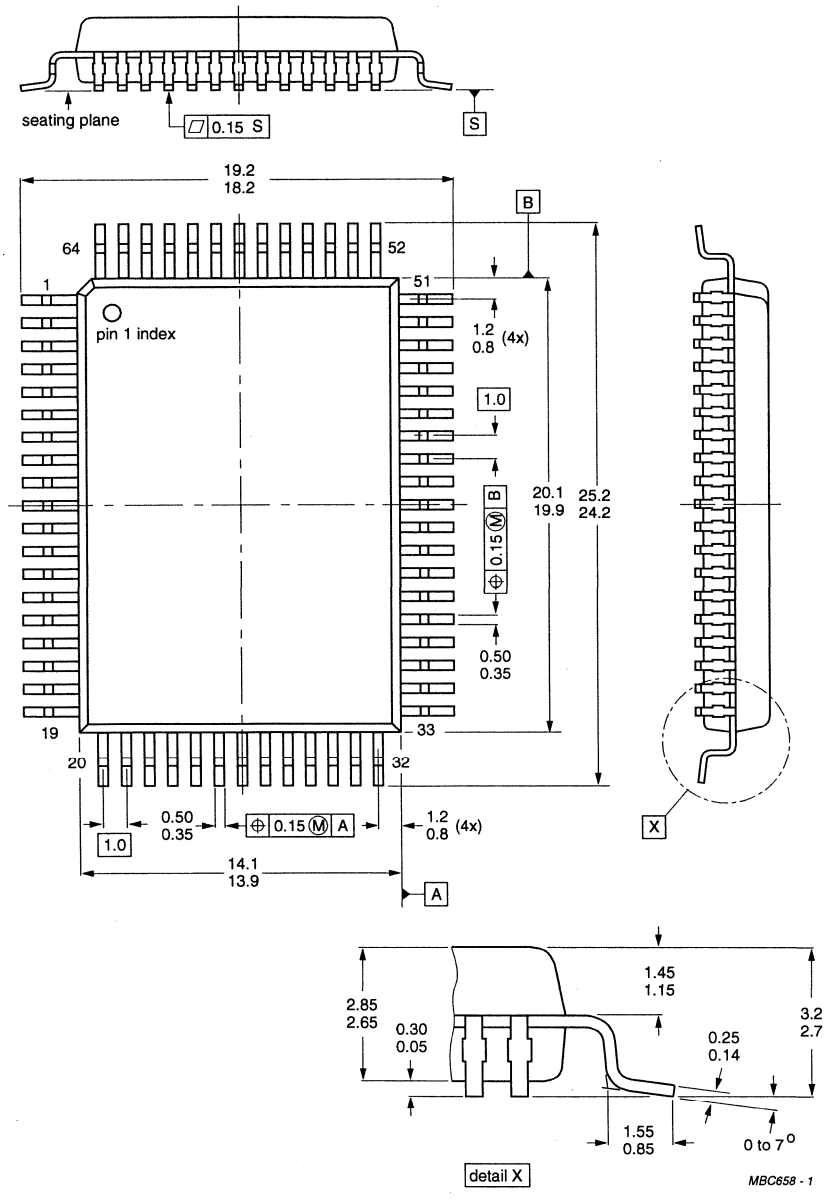

Fig.11 56-lead mini-pack; plastic (VSO56; SOT190).

Dimensions in mm.



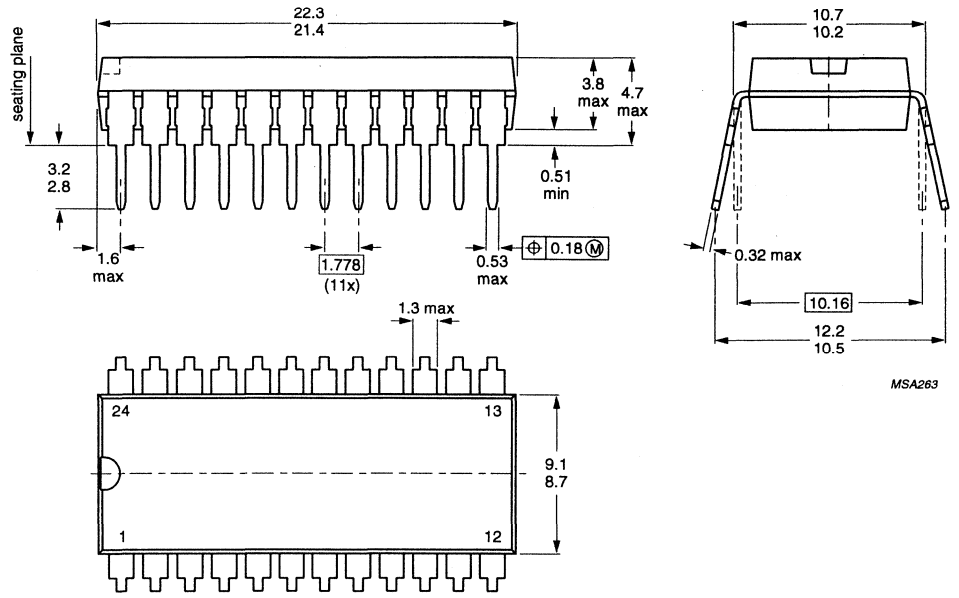

Dimensions in mm.

Fig.12 44-lead quad flat-pack; plastic (SOT205AG).



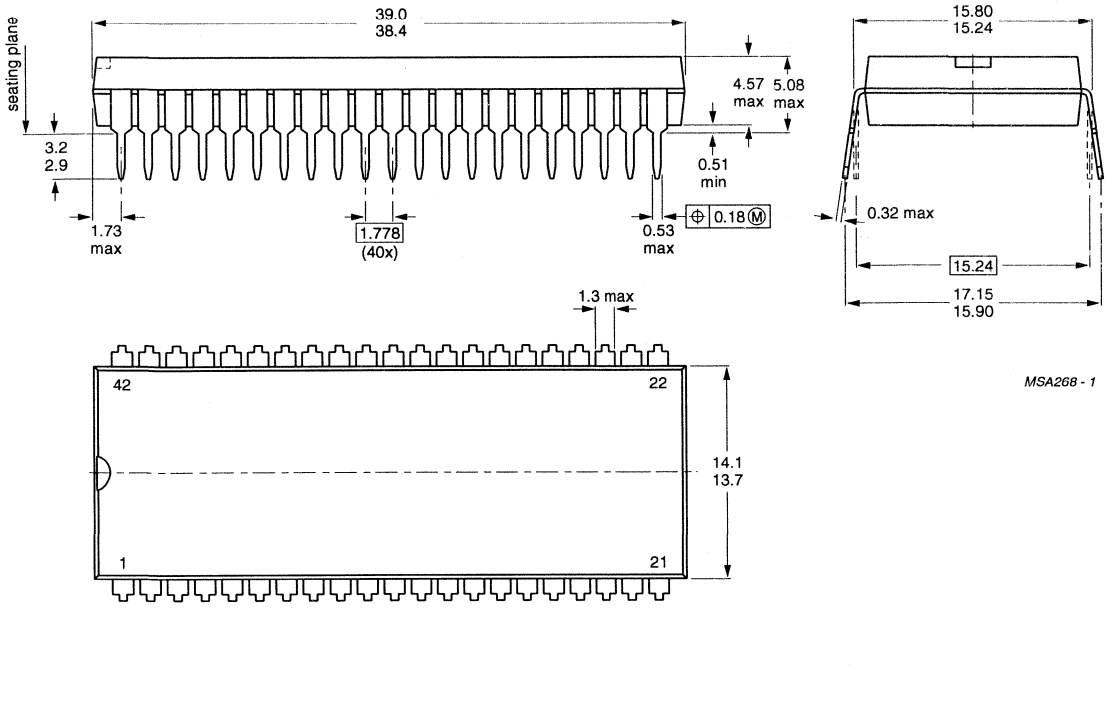

Dimensions in mm.

Fig.13 64-lead quad flat-pack; plastic (SOT208).




Dimensions in mm.

Fig.14 24-lead shrink dual in-line; plastic (SDIL24; SOT234A).




Dimensions in mm.

Fig.15 42-lead shrink dual in-line; plastic (SDIL42; SOT270).

## SOLDERING

### Plastic mini-packs, plastic quad flat-packs and plastic leaded chip carriers

BY WAVE

During placement and before soldering, the component must be fixed with a droplet of adhesive. After curing the adhesive, the component can be soldered. The adhesive can be applied by screen printing, pin transfer or syringe dispensing.

Maximum permissible solder temperature is 260 °C, and maximum duration of package immersion in solder bath is 10 s, if allowed to cool to less than 150 °C within 6 s. Typical dwell time is 4 s at 250 °C.

A modified wave soldering technique is recommended using two solder waves (dual-wave), in which a turbulent wave with high upward pressure is followed by a smooth laminar wave. Using a mildly-activated flux eliminates the need for removal of corrosive residues in most applications.

BY SOLDER PASTE REFLOW

Reflow soldering requires the solder paste (a suspension of fine solder particles, flux and binding agent) to be applied to the substrate by screen printing, stencilling or pressure-syringe dispensing before device placement.

Several techniques exist for reflowing; for example, thermal conduction by heated belt, infrared, and vapour-phase reflow. Dwell times vary between 50 and 300 s according to 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 min at 45 °C.

REPAIRING SOLDERED JOINTS (BY HAND-HELD SOLDERING IRON OR PULSE-HEATED SOLDER TOOL)

Fix the component by first soldering two, diagonally opposite, end pins. Apply the heating tool to the flat part of the pin only. Contact time must be limited to 10 s at up to 300 °C. When using proper tools, all other pins can be soldered in one operation within 2 to 5 s at between 270 and 320 °C. (Pulse-heated soldering is not recommended for SO packages.)

For pulse-heated solder tool (resistance) soldering of VSO packages, solder is applied to the substrate by dipping or by an extra thick tin/lead plating before package placement.

### Plastic dual in-line packages

BY DIP OR WAVE

The maximum permissible temperature of the solder is 260 °C; this temperature must not be in contact with the joint for more than 5 s. The total contact time of successive solder waves must not exceed 5 s.

The device may be mounted up to the seating plane, but the temperature of the plastic body must not exceed the specified storage maximum. If the printed-circuit board has been pre-heated, forced cooling may be necessary immediately after soldering to keep the temperature within the permissible limit.

REPAIRING SOLDERED JOINTS

Apply the soldering iron below the seating plane (or not more than 2 mm above it). If its temperature is below 300 °C, it must not be in contact for more than 10 s; if between 300 and 400 °C, for not more than 5 s.

# SECTION 6 DATA HANDBOOK SYSTEM

# 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.

Catalogues are available for selected product ranges (some catalogues are also on floppy discs).

Our data handbook titles are listed here.

### Integrated circuits

| *Book* | *Title* |
|---|---|
| IC01 | Semiconductors for Radio and Audio Systems |
| IC02 | Semiconductors for Television and Video Systems |
| IC03 | Semiconductors for Telecom Systems |
| IC04 | CMOS HE4000B Logic Family |
| IC05 | Advanced Low-power Schottky (ALS) Logic Series |
| IC06 | High-speed CMOS Logic Family |
| IC08 | 100K ECL Logic Family |
| IC10 | Memories |
| IC11 | General-purpose/Linear ICs |
| IC12 | Display Drivers and Microcontroller Peripherals (planned) |
| IC13 | Programmable Logic Devices (PLD) |
| IC14 | 8048-based 8-bit Microcontrollers |
| IC15 | FAST TTL Logic Series |
| IC16 | ICs for Clocks and Watches |
| IC17 | RF/Wireless Communications |
| IC18 | Semiconductors for In-car Electronics and General Industrial Applications (planned) |
| IC19 | Semiconductors for Datacom: LANs, UARTs, Multi-protocol Controllers and Fibre Optics |
| IC20 | 8051-based 8-bit Microcontrollers |
| IC21 | 68000-based 16-bit Microcontrollers (planned) |
| IC22 | ICs for Multi-media Systems (planned) |
| IC23 | QUBIC Advanced BiCMOS Interface Logic ABT, MULTIBYTE™ |
| IC24 | Low Voltage CMOS Logic |

### Discrete semiconductors

| *Book* | *Title* |
|---|---|
| SC01 | Diodes |
| SC02 | Power Diodes |
| SC03 | Thyristors and Triacs |
| SC04 | Small-signal Transistors |
| SC05 | Low-frequency Power Transistors and Hybrid IC Power Modules |
| SC06 | High-voltage and Switching NPN Power Transistors |
| SC07 | Small-signal Field-effect Transistors |
| SC08a | RF Power Bipolar Transistors |
| SC08b | RF Power MOS Transistors |
| SC09 | RF Power Modules |
| SC10 | Surface Mounted Semiconductors |
| SC13 | PowerMOS Transistors including TOPFETs and IGBTs |
| SC14 | RF Wideband Transistors, Video Transistors and Modules |
| SC15 | Microwave Transistors |
| SC16 | Wideband Hybrid IC Modules |
| SC17 | Semiconductor Sensors |

### Professional components

| | |
|---|---|
| PC01 | High-power Klystrons and Accessories |
| PC06 | Circulators and Isolators |

### MORE INFORMATION FROM PHILIPS SEMICONDUCTORS?

For more information about Philips Semiconductors data handbooks, catalogues 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 enquiries 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 Display Components<br>Colour TV Picture Tubes and Assemblies<br>Colour Monitor Tube Assemblies |
| DC02 | Monochrome Monitor Tubes and Deflection Units |
| DC03 | Television Tuners, Coaxial Aerial Input Assemblies |
| DC05 | Flyback Transformers, Mains Transformers and General-purpose FXC Assemblies |

### Magnetic products

| | |
|---|---|
| MA01 | Soft Ferrites |
| MA03 | Piezoelectric Ceramics<br>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 |
| PA07 | Quartz Crystals for Special and Industrial Applications |
| PA08 | Fixed Resistors |
| PA10 | Quartz Crystals for Automotive and Standard Applications |
| PA11 | Quartz Oscillators |

### Professional components

| | |
|---|---|
| PC04 | Photo Multipliers |
| PC05 | Plumbicon Camera Tubes and Accessories |
| PC07 | Vidicon and Newvicon Camera Tubes and Deflection Units |
| PC08 | Image Intensifiers |
| PC12 | Electron Multipliers |

## MORE INFORMATION FROM PHILIPS COMPONENTS?

For more information contact your nearest Philips Components national organization shown in the following list.

**Argentina:** Buenos Aires, Tel. (541)786 7635, Fax. (541)786 9367.
**Australia:** NORTH RYDE, Tel. (02)805 4455, Fax. (02)805 4466.
**Austria:** WIEN, Tel. (01)60101 1820, Fax. (01)60101 1210.
**Belgium:** EINDHOVEN, Tel. (31)40 783749, Fax. (31)40 788399.
**Brazil:** SÃo Paulo, Tel. (011)821 2333, Fax. (011)829 1849.
**Canada:** SCARBOROUGH, Tel. (0416)292 5161, Fax. (0416)754 6248.
**Chile:** SANTIAGO, Tel. (02)77 38 16, Fax. (02)735 3594.
**China** (Peoples Republic of): Shanghai, Tel. (021)326 4140,Fax. (021)320 2160.
**Colombia:** BOGOTA, Tel. (571)249 7624/(571)217 4609, Fax. (571)217 4549.
**Denmark:** COPENHAGEN, Tel. (032)883 333, Fax. (031)571 949.
**Finland:** ESPOO, Tel. (9)0-50261, Fax. (9)0-520971.
**France:** SURESNES, Tel. (01)4099 6161, Fax. (01)4099 6431.
**Germany:** HAMBURG, Tel. (040)3296-0, Fax. (040)3296 213.
**Greece:** TAVROS, Tel. (01)489 4339/(01)489 4911, Fax. (01)481 5180.
**Hong Kong:** KWAI CHUNG, Tel. (852)424 5121, Fax. (852)428 6729.
**India:** BOMBAY, Tel. (022)4938 541, Fax. (022)4938 722.
**Indonesia:** JAKARTA, Tel. (021)5201122, Fax. (021)5205189.
**Ireland:** DUBLIN, Tel. (01)640 203, Fax. (01)640 210.
**Israel:** Tel Aviv Tel. (9723)6450333, Fax. (9723)493272.
**Italy:** MILANO, Tel. (02)6752.3302, Fax. (02)6752 3300.
**Japan:** TOKIO, Tel. (03)3740 5143, Fax. (03)3740 5035.
**Korea** (Republic of): SEOUL, Tel. (02)709-1412, Fax. (02)709-1415.
**Malaysia:** Kuala Lumpur, Tel. (03)757 5511, Fax. (03)757 4880.
**Mexico:** CHI HUA HUA, Tel. (016)18-67-01/(016)18-67-02, Fax. (016)778 0551.
**Netherlands:** EINDHOVEN, Tel. (040)783749, Fax, (040)788399.
**New Zealand:** AUCKLAND, Tel. (09)849-4160, Fax. (09)849-7811.
**Norway:** OSLO, Tel. (22)74 8000, Fax. (22)74 8341.
**Pakistan:** KARACHI, Tel. (021)587 4641-49, Fax. (021)577035/5874546.
**Philippines:** MANILA, Tel. (02)810-0161, Fax. (02)817-3474.
**Portugal:** Linda-A-Velha, Tel. (01)14163160/4163333, Fax. (01)14163174/4163366.
**Singapore:** SINGAPORE, Tel. (65)350 2000, Fax. (65)355 1758.
**South Africa:** JOHANNESBURG, Tel. (011)470-5911, Fax. (011)470-5494.
**Spain:** BARCELONA, Tel. (03)301 6312, Fax. (03)301 4243.
**Sweden:** STOCKHOLM, Tel. (08)632 2000, Fax. (08)632 2745.
**Switzerland:** ZÜRICH, Tel. (01)488 2211, Fax. (01)481 77 30.
**Taiwan:** TAIPEI , Tel. (02)388 7666, Fax. (02)382 4382.
**Thailand:** BANGKOK, Tel. (662)398-0141, Fax. (662)398-3319.
**Turkey:** ISTANBUL, Tel. (0212)279 2770, Fax. (0212)269 3094.
**United Kingdom:** LONDON, Tel. (071)580 6633, Fax. (071)636 0394.
**United States:** RIVIERA BEACH, Tel.( 407)881-3200, Fax. (407)881 3300.
**Uruguay:** MONTEVIDEO, Tel. (02)704 044, Fax. (02)920 601.

For all other countries apply to:
**Philips Components,**
Marketing Communications,
P.O. Box 218,
5600 MD, EINDHOVEN, The Netherlands
Telex 35000 phtcnl, Fax. +31-40-724547.

NOTES

# *Philips Semiconductors – a worldwide company*


**Argentina:** IEROD, Av. Juramento 1992 - 14.b, (1428)
BUENOS AIRES, Tel. (541)786 7633, Fax. (541)786 9367

**Australia:** 34 Waterloo Road, NORTH RYDE, NSW 2113,
Tel. (02)805 4455, Fax. (02)805 4466

**Austria:** Triester Str. 64, A-1101 WIEN, P.O. Box 213,
Tel. (01)60 101-1236, Fax. (01)60 101-1211

**Belgium:** Postbus 90050, 5600 PB EINDHOVEN, The Netherlands,
Tel. (31)40 783 749, Fax. (31)40 788 399

**Brazil:** Rua do Rocio 220 - 5th floor, Suite 51,
CEP: 04552-903-SÃO PAULO-SP, Brazil.
P.O. Box 7383 (01064-970).
Tel. (011)821-2333, Fax. (011)829-1849

**Canada:** INTEGRATED CIRCUITS:
Tel. (800)234-7381, Fax. (708)296-8556
DISCRETE SEMICONDUCTORS: 601 Milner Ave,
SCARBOROUGH, ONTARIO, M1B 1M8,
Tel. (0416)292 5161 ext. 2336, Fax. (0416)292 4477

**Chile:** Av. Santa Maria 0760, SANTIAGO,
Tel. (02)773 816, Fax. (02)777 6730

**Colombia:** IPRELENSO LTDA, Carrera 21 No. 56-17,
77621 BOGOTA, Tel. (571)249 7624/(571)217 4609,
Fax. (571)217 4549

**Denmark:** Prags Boulevard 80, PB 1919, DK-2300 COPENHAGEN S,
Tel. (032)88 2636, Fax. (031)57 1949

**Finland:** Sinikalliontie 3, FIN-02630 ESPOO,
Tel. (9)0-50261, Fax. (9)0-520971

**France:** 4 Rue du Port-aux-Vins, BP317,
92156 SURESNES Cedex,
Tel. (01)4099 6161, Fax. (01)4099 6427

**Germany:** PHILIPS COMPONENTS UB der Philips G.m.b.H.,
P.O. Box 10 63 23, 20043 HAMBURG,
Tel. (040)3296-0, Fax. (040)3296 213.

**Greece:** No. 15, 25th March Street, GR 17778 TAVROS,
Tel. (01)4894 339/4894 911, Fax. (01)4814 240

**Hong Kong:** PHILIPS HONG KONG Ltd., Components Div.,
6/F Philips Ind. Bldg., 24-28 Kung Yip St., KWAI CHUNG, N.T.,
Tel. (852)424 5121, Fax. (852)428 6729

**India:** Philips INDIA Ltd, Components Dept,
Shivsagar Estate, A Block ,
Dr. Annie Besant Rd. Worli, Bombay 400 018
Tel. (022)4938 541, Fax. (022)4938 722

**Indonesia:** Philips House, Jalan H.R. Rasuna Said Kav. 3-4,
P.O. Box 4252, JAKARTA 12950,
Tel. (021)5201 122, Fax. (021)5205 189

**Ireland:** Newstead, Clonskeagh, DUBLIN 14,
Tel. (01)640 000, Fax. (01)640 200

**Italy:** PHILIPS COMPONENTS S.r.l.,
Viale F. Testi, 327, 20162 MILANO,
Tel. (02)6752.3302, Fax. (02)6752 3300.

**Japan:** Philips Bldg 13-37, Kohnan 2-chome, Minato-ku, TOKYO 108,
Tel. (03)3740 5028, Fax. (03)3740 0580

**Korea:** (Republic of) Philips House, 260-199 Itaewon-dong,
Yongsan-ku, SEOUL, Tel. (02)794-5011, Fax. (02)798-8022

**Malaysia:** No. 76 Jalan Universiti, 46200 PETALING JAYA,
SELANGOR, Tel. (03)750 5214, Fax. (03)757 4880

**Mexico:** Philips Components, 5900 Gateway East, Suite 200,
EL PASO, TX 79905, Tel. 9-5(800)234-7381, Fax. (708)296-8556

**Netherlands:** Postbus 90050, 5600 PB EINDHOVEN, Bldg. VB
Tel. (040)783749, Fax. (040)788399

**New Zealand:** 2 Wagener Place, C.P.O. Box 1041, AUCKLAND,
Tel. (09)849-4160, Fax. (09)849-7811

**Norway:** Box 1, Manglerud 0612, OSLO,
Tel. (022)74 8000, Fax. (022)74 8341

**Pakistan:** Philips Electrical Industries of Pakistan Ltd.,
Exchange Bldg. ST-2/A, Block 9, KDA Scheme 5, Clifton,
KARACHI 75600, Tel. (021)587 4641-49,
Fax. (021)577035/5874546.

**Philippines:** PHILIPS SEMICONDUCTORS PHILIPPINES Inc,
106 Valero St. Salcedo Village, P.O. Box 2108 MCC, MAKATI,
Metro MANILA, Tel. (02)810 0161, Fax. (02)817 3474

**Portugal:** PHILIPS PORTUGUESA, S.A.,
Rua dr. António Loureiro Borges 5, Arquiparque - Miraflores,
Apartado 300, 2795 LINDA-A-VELHA,
Tel. (01)14163160/4163333, Fax. (01)14163174/4163366.

**Singapore:** Lorong 1, Toa Payoh, SINGAPORE 1231,
Tel. (65)350 2000, Fax. (65)251 6500

**South Africa:** S.A. PHILIPS Pty Ltd., Components Division,
195-215 Main Road Martindale, 2092 JOHANNESBURG,
P.O. Box 7430 Johannesburg 2000,
Tel. (011)470-5911, Fax. (011)470-5494.

**Spain:** Balmes 22, 08007 BARCELONA,
Tel. (03)301 6312, Fax. (03)301 42 43

**Sweden:** Kottbygatan 7, Akalla. S-164 85 STOCKHOLM,
Tel. (0)8-632 2000, Fax. (0)8-632 2745

**Switzerland:** Allmendstrasse 140, CH-8027 ZÜRICH,
Tel. (01)488 2211, Fax. (01)481 77 30

**Taiwan:** PHILIPS TAIWAN Ltd., 23-30F, 66, Chung Hsiao West
Road, Sec. 1. Taipeh, Taiwan ROC, P.O. Box 22978,
TAIPEI 100, Tel. (02)388 7666, Fax. (02)382 4382.

**Thailand:** PHILIPS ELECTRONICS (THAILAND) Ltd.,
209/2 Sanpavuth-Bangna Road Prakanong,
Bangkok 10260, THAILAND,
Tel. (662)398-0141, Fax. (662)398-3319.

**Turkey:** Talatpasa Cad. No. 5, 80640 GÜLTEPE/ISTANBUL,
Tel. (0212)279 2770, Fax. (0212)269 3094

**United Kingdom:** Philips Semiconductors Limited, P.O. Box 65,
Philips House, Torrington Place, LONDON, WC1E 7HD,
Tel. (071)436 41 44, Fax. (071)323 03 42

**United States:** INTEGRATED CIRCUITS:
811 East Arques Avenue, SUNNYVALE, CA 94088-3409,
Tel. (800)234-7381, Fax. (708)296-8556
DISCRETE SEMICONDUCTORS: 2001 West Blue Heron Blvd.,
P.O. Box 10330, RIVIERA BEACH, FLORIDA 33404,
Tel. (800)447-3762 and (407)881-3200, Fax. (407)881-3300

**Uruguay:** Coronel Mora 433, MONTEVIDEO,
Tel. (02)70-4044, Fax. (02)92 0601

**For all other countries apply to:** Philips Semiconductors,
International Marketing and Sales, Building BAF-1,
P.O. Box 218, 5600 MD, EINDHOVEN, The Netherlands,
Telex 35000 phtcnl, Fax. +31-40-724825





Printed in The Netherlands

453051/13000/02/pp530 Date of release: June 1994
Document order number: 9398 652 40011


**Philips Semiconductors**

PHILIPS

**PHILIPS**