Am186™EM and Am188™EM Microcontrollers

User's Manual

AMD

Trademarks

IF YOU HAVE QUESTIONS, WE'RE HERE TO HELP YOU.

Customer Service

The AMD customer service network includes U.S. offices, international offices, and a customer training center. Expert technical assistance is available from the worldwide staff of AMD field application engineers and factory support staff to answer E86™ family hardware and software development questions.

Hotline and World Wide Web Support

For answers to technical questions, AMD provides a toll-free number for direct access to our corporate applications hotline. Also available is the AMD World Wide Web home page and FTP site, which provides the latest E86 family product information, including technical information and data on upcoming product releases.

Corporate Applications Hotline

(800) 222-9323	toll-free for U.S. and Canada
44-(0) 1276-803-299	U.K. and Europe hotline

World Wide Web Home Page and FTP Site

To access the AMD home page, go to http://www.amd.com.

To download documents and software, ftp to ftp.amd.com and log on as anonymous using your E-mail address as a password. Or via your web browser, go to ftp://ftp.amd.com.

Questions, requests, and input concerning AMD's WWW pages can be sent via E-mail to webmaster@amd.com.

Documentation and Literature

Free E86 family information such as data books, user's manuals, data sheets, application notes, the FusionE86SM Partner Solutions Catalog, and other literature is available with a simple phone call. Internationally, contact your local AMD sales office for complete E86 family literature.

Literature Ordering

800-222-9323	toll-free for U.S. and Canada
512-602-5651	direct dial worldwide
512-602-7639	fax
800-222-9323	AMD Facts-On-Demand™ faxback service toll-free for U.S. and Canada

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

INTRODUCTION AND OVERVIEW

DESIGN PHILOSOPHY

AMD's Am186 and Am188 family of microcontrollers is based on the architecture of the original 8086 and 8088 microcontrollers, and currently includes the 80C186, 80C188, 80L186, 80L188, Am186™EM, Am188™EM, Am186EMLV, Am188EMLV, Am186ES, Am188ES, Am186ESLV, Am188ESLV, Am186ER, and Am188ER microcontrollers. The Am186EM and Am188EM microcontrollers provide a natural migration path for 80C186/188 designs that need performance and cost enhancements.

The Am186EM and Am188EM microcontrollers provide a low-cost, high-performance solution for embedded system designers who want to use the x86 architecture. By integrating multiple functional blocks with the CPU, the Am186EM and Am188EM microcontrollers eliminate the need for off-chip system-interface logic. It is possible to implement a fully functional system with ROM and RAM, serial interfaces, and custom I/O capability without additional system-interface logic.

The Am186EM and Am188EM microcontrollers can operate at frequencies up to 40 MHz. The microcontrollers include an on-board PLL so that the input clock can be one-to-one with the internal processor clock. The Am186EM and Am188EM microcontrollers are available in versions operating at 20, 25, 33, and 40 MHz.

PURPOSE OF THIS MANUAL

This manual describes the technical features and programming interface of the Am186EM and Am188EM microcontrollers. The complete instruction set is documented in the *Am186 and Am188 Family Instruction Set Manual,* order #21267.

INTENDED AUDIENCE

This manual is intended for computer hardware and software engineers and system architects who are designing or are considering designing systems based on the Am186EM and Am188EM microcontrollers.

USER'S MANUAL OVERVIEW

This manual contains information on the Am186EM and Am188EM microcontrollers and is essential for system architects and design engineers. Additional information is available in the form of data sheets, application notes, and other documentation that is provided with software products and hardware-development tools.

The information in this manual is organized into 12 chapters and 1 appendix.

- Chapter 1 introduces the **features and performance** aspects of the Am186EM and Am188EM microcontrollers.
- Chapter 2 describes the **programmer's model** of the Am186 and Am188 family microcontrollers, including an instruction set overview and register model.
- Chapter 3 provides an overview of the **system interfaces**, along with clocking features.

- Chapter 4 provides a description of the **peripheral control block** along with power management and reset configuration.
- Chapter 5 provides a description of the **chip select unit**.
- Chapter 6 provides a description of the **refresh control unit**.
- Chapter 7 provides a description of the **on-chip interrupt controller**.
- Chapter 8 describes the **timer control unit**.
- Chapter 9 describes the **DMA controller**.
- Chapter 10 describes the **asynchronous serial port**.
- Chapter 11 describes the **synchronous serial interface**.
- Chapter 12 describes the **programmable I/O pins**.
- Appendix A includes a complete summary of **peripheral registers and fields**.

For complete information on the Am186EM and Am188EM microcontroller pin lists, timing, thermal characteristics, and physical dimensions, please refer to the *Am186EM/EMLV and Am188EM/EMLV Microcontrollers Data Sheet* (order# 19168).

AMD DOCUMENTATION

E86 Family

ORDER NO.	DOCUMENT TITLE
19168	**Am186EM/EMLV and Am188EM/EMLV Microcontrollers Data Sheet** Hardware documentation: pin descriptions, functional descriptions, absolute maximum ratings, operating ranges, switching characteristics and waveforms, connection diagrams and pinouts, and package physical dimensions.
21267	**Am186 and Am188 Family Instruction Set Manual** Provides a detailed description and examples for each instruction included in the Am186 and Am188 Family Instruction Set.
19255	**FusionE86[SM] Catalog** Provides information on tools that speed an E86 family embedded product to market. Includes products from expert suppliers of embedded development solutions.
20071	**E86 Family Support Tools Brief** Lists available E86 family software and hardware development tools, as well as contact information for suppliers.
21058	**FusionE86 Development Tools Reference CD** Provides a single-source multimedia tool for customer evaluation of AMD products, as well as Fusion partner tools and technologies that support the E86 family of microcontrollers and microprocessors. Technical documentation for the E86 family is included on the CD in PDF format.

To order literature, contact the nearest AMD sales office or call 800-222-9323 (in the U.S. and Canada) or direct dial from any location 512-602-5651.

Literature is also available in postscript and PDF formats on the AMD web site. To access the AMD home page, go to http://www.amd.com. To download documents and software, ftp to ftp.amd.com and log on as anonymous using your E-mail address as a password. Or via your web browser, go to ftp://ftp.amd.com.

CHAPTER

1 FEATURES AND PERFORMANCE

Compared to the 80C186/188 microcontrollers, the Am186™EM and Am188™EM microcontrollers enable designers to increase performance and functionality, while reducing the cost, size, and power consumption of embedded systems. The Am186EM and Am188EM microcontrollers are cost-effective, enhanced versions of the AMD 80C186/188 devices.

The Am186EM and Am188EM microcontrollers are the ideal upgrade for 80C186/188 designs requiring 80C186/188-compatibility, increased performance, serial communications, and a glueless bus interface. Developed exclusively for the embedded marketplace, the Am186EM and Am188EM microcontrollers increase the performance of existing 80C186/188 systems while decreasing their cost.

Because the Am186EM and Am188EM microcontrollers integrate on-chip peripherals and offer up to twice the performance of an 80C186/188, they are ideal upgrade solutions for customers requiring more integration and performance than their present x86 solution delivers.

1.1 KEY FEATURES AND BENEFITS

The Am186EM and Am188EM microcontrollers extend the AMD family of microcontrollers based on the industry-standard x86 architecture. The Am186EM and Am188EM microcontrollers deliver higher performance and more integration than the 80C186/188 core microcontrollers. Upgrading to the Am186EM or Am188EM microcontrollers is attractive for the following reasons:

- **Minimized total system cost**—The new peripherals and on-chip system-interface logic reduce the cost of existing 80C186 designs.
- **x86 software compatibility**—80C186/188-compatible and upward-compatible with the AMD E86 family.
- **Enhanced performance**—The Am186EM and Am188EM microcontrollers can provide increased performance over 80C186/188 systems, and the nonmultiplexed address bus offers faster, unbuffered access to memory.
- **No wait-state operation**—At 40 MHz with 70-ns memories.
- **Enhanced functionality**—The new and enhanced on-chip peripherals of the Am186EM and Am188EM microcontrollers include an asynchronous serial port, a watchdog timer interrupt, an additional interrupt pin, a high-speed synchronous serial interface, a PSRAM controller, a 16-bit Reset Configuration register, enhanced chip-select functionality, 32 programmable I/Os, and additional interrupt signals.

The Am186EM and Am188EM microcontrollers are part of the AMD E86 family of embedded microcontrollers and microprocessors based on the x86 architecture. The 16-bit members of the E86 family, referred to throughout this manual as the Am186 and Am188 family, include the 80C186, 80C188, 80L186, 80L188, Am186EMLV, Am188EMLV, Am186ES, Am188ES, Am186ESLV, Am188ESLV, Am186ER, and Am188ER microcontrollers.

The Am186EM and Am188EM microcontrollers are designed to meet the most common requirements of embedded products developed for the office automation, mass storage, communications, and general embedded markets. Applications include disk drives, hand-held terminals, fax machines, terminals, printers, photocopiers, feature phones, cellular phones, PBXs, multiplexers, modems, and industrial controls.

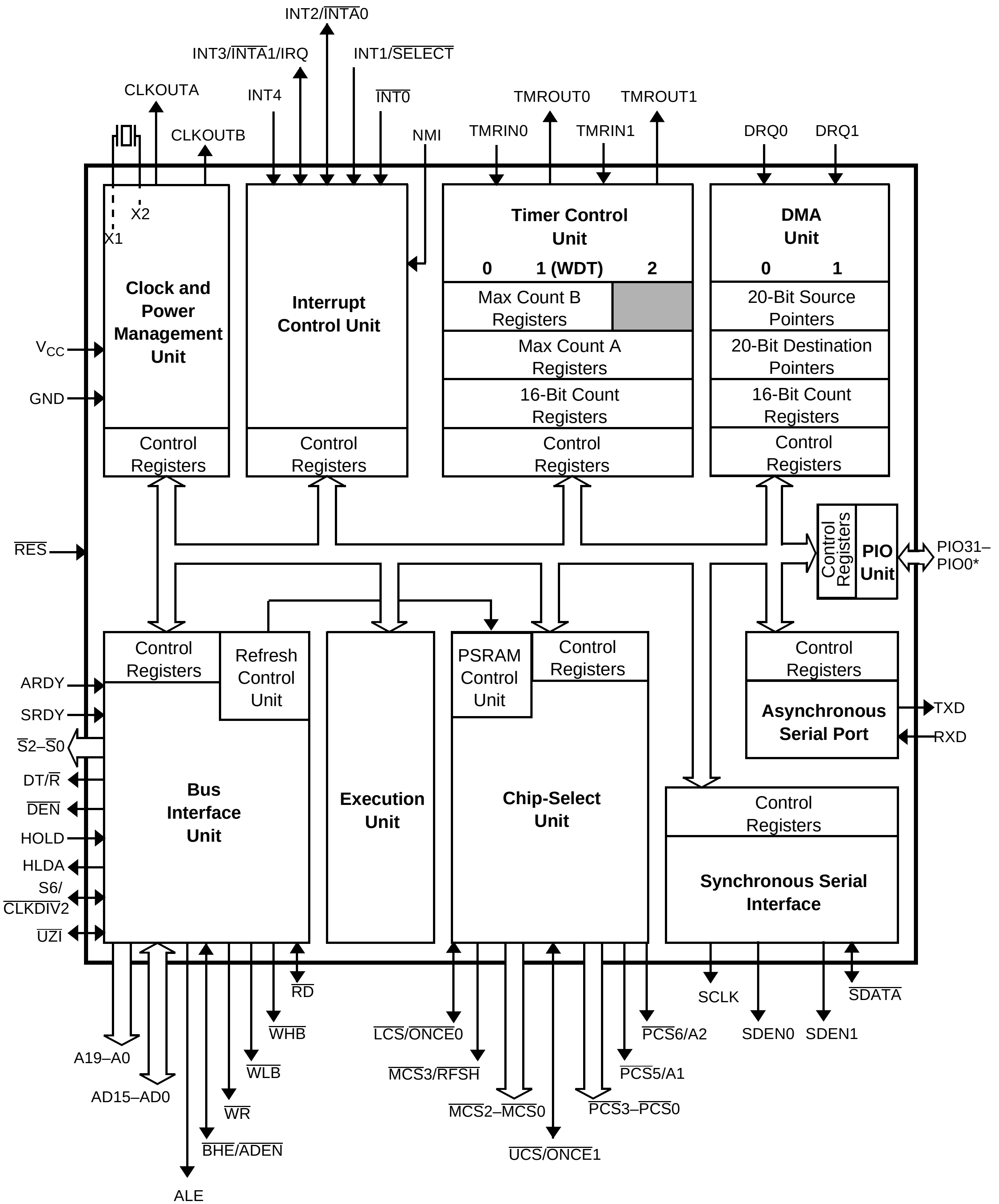
1.2 DISTINCTIVE CHARACTERISTICS

A block diagram of each microcontroller is shown in Figure 1-1 and Figure 1-2. The Am186EM microcontroller uses a 16-bit external bus, while the Am188EM microcontroller has an 8-bit external bus.

The Am186EM and Am188EM microcontrollers provide the following features:

- High performance:
  - 20-, 25-, 33-, and 40-MHz operating frequencies
  - Support for zero wait-state operation at 40 MHz with 70-ns memory
  - 1-Mbyte memory address space and 64-Kbyte I/O space
- New features remove the requirement for a 2x clock input and provide faster access to memory:
  - Phase-locked loop (PLL) allows processor to operate at the clock input frequency
  - Nonmultiplexed address bus
- New integrated peripherals increase functionality while reducing system cost:
  - 32 programmable I/O (PIO) pins
  - Asynchronous serial port allows full-duplex, 7-bit or 8-bit data transfers
  - Pseudo-static RAM (PSRAM) controller includes auto refresh capability
  - Reset Configuration register
  - Synchronous serial interface allows high-speed, half-duplex, bidirectional data transfer to and from application-specific integrated circuits (ASICs)
  - Additional external interrupts
- Familiar 80C186 peripherals:
  - Two independent DMA channels
  - Programmable interrupt controller with five external interrupts
  - Three programmable 16-bit timers
  - Timer 1 can be configured to provide a watchdog timer interrupt
  - Programmable memory and peripheral chip-select logic
  - Programmable wait-state generator
  - Power-save mode
- Software-compatible with the 80C186/188 microcontroller
- Widely available native development tools, applications, and system software
- Available in the following packages:
  - 100-pin, thin quad flat pack (TQFP)
  - 100-pin, plastic quad flat pack (PQFP)

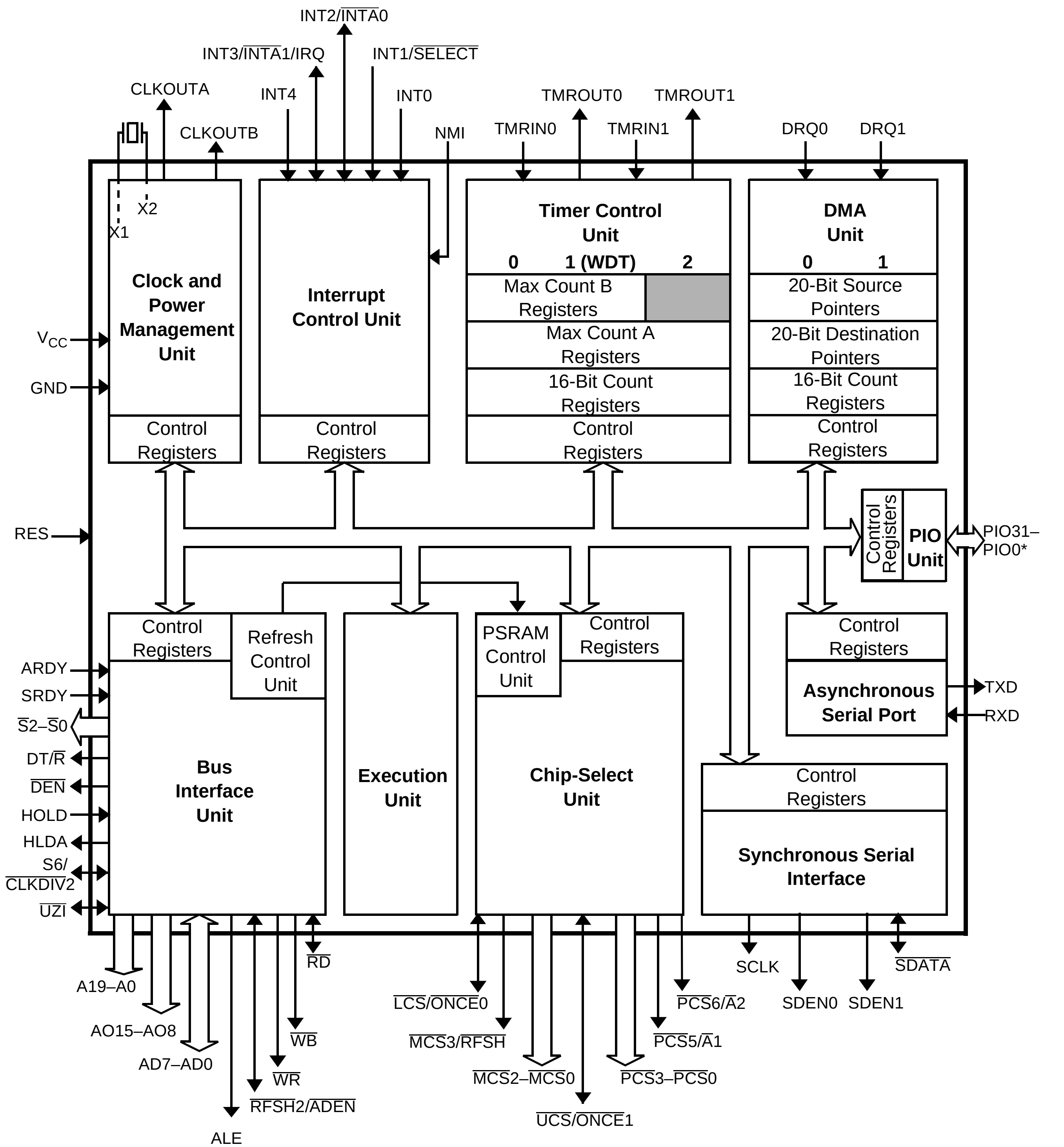
Figure 1-1   Am186EM Microcontroller Block Diagram



Note:

** All PIO signals are shared with other physical pins. See the pin descriptions in Chapter 3 and Table 3-1 on page 3-9 for information on shared functions.*

Figure 1-2 Am188EM Microcontroller Block Diagram



Note:

** All PIO signals are shared with other physical pins. See the pin descriptions in Chapter 3 and Table 3-1 on page 3-9 for information on shared functions.*

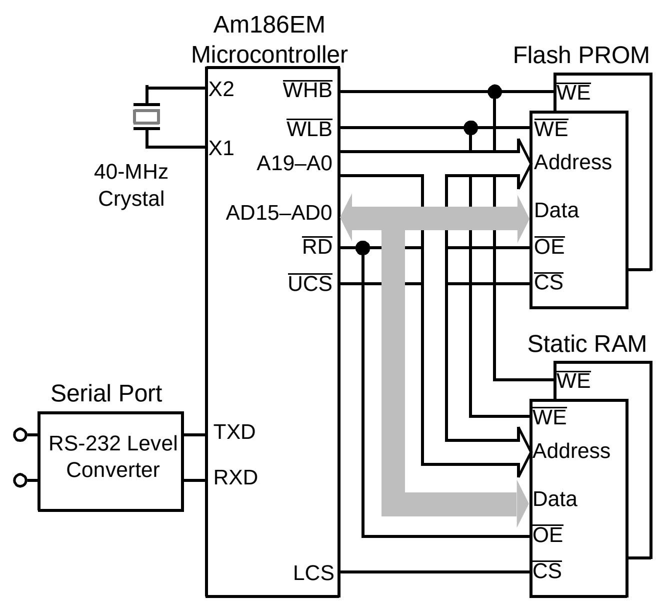
1.3 APPLICATION CONSIDERATIONS

The integration enhancements of the Am186EM and Am188EM microcontrollers provide a high-performance, low-system-cost solution for 16-bit embedded microcontroller designs. The nonmultiplexed address bus (A19–A0) eliminates system-interface logic for memory devices, while the multiplexed address/data bus maintains the value of existing customer-specific peripherals and circuits within the upgraded design.

The nonmultiplexed address bus is available in addition to the 80C186 and 80C188 microcontrollers' multiplexed address/data bus (AD15–AD0). The two buses can operate simultaneously or the AD15–AD0 bus can be configured to operate only during the data phase of a bus cycle. See the $\overline{\text{BHE}}$/$\overline{\text{ADEN}}$ and $\overline{\text{RFSH2}}$/$\overline{\text{ADEN}}$ pin descriptions in Chapter 3, and see section 5.5.1 and section 5.5.2 for additional information regarding the AD15–AD0 address enabling and disabling.

Figure 1-3 illustrates a functional system design that uses the integrated peripheral set to achieve high performance with reduced system cost.

Figure 1-3 Basic Functional System Design



1.3.1 Clock Generation

The integrated PLL clock-generation circuitry of the Am186EM and Am188EM microcontrollers allows the use of a *times-one* crystal frequency. The design in Figure 1-3 achieves 40-MHz CPU operation with a 40-MHz crystal.

The integrated PLL lowers system cost by reducing the cost of the crystal and reduces electromechanical interference (EMI) in the system.

1.3.2 Memory Interface

The integrated memory controller logic of the Am186EM and Am188EM microcontrollers provides a direct address bus interface to memory devices. The use of an external address latch controlled by the address latch enable (ALE) signal is not required.

Individual byte write-enable signals are provided to eliminate the need for external high/low-byte, write-enable circuitry. The maximum bank size programmable for the memory chip-select signals is increased to 512 Kbytes to facilitate the use of high-density memory devices.

Improved memory timing specifications enables the use of no-wait-state memories with 70-ns access times at 40-MHz CPU operation. This reduces overall system cost significantly by allowing the use of commonly available memory devices.

Figure 1-3 illustrates an Am186EM microcontroller-based SRAM configuration. The memory interface requires the following:

- The processor A19–A0 bus connects to the memory address inputs.
- The AD bus connects directly to the data inputs/outputs.
- The chip selects connect to the memory chip-select inputs.

Read operations require that the $\overline{RD}$ output connects to the SRAM Output Enable ($\overline{OE}$) input pins. Write operations require that the byte write enables connect to the SRAM Write Enable ($\overline{WE}$) input pins.

The design uses 2-Mbit (256-Kbyte) memory technology to fully populate the available address space. Two Flash PROM devices provide 512 Kbytes of nonvolatile program storage, and two static RAM devices provide 512 Kbytes of variable storage area.

1.3.3 Serial Communications Port

The integrated universal asynchronous receiver/transmitter (UART) controller in the Am186EM and Am188EM microcontrollers eliminates the need for external logic to implement a communications interface. The integrated UART generates the serial clock from the CPU clock so that no external time-base oscillator is required.

Figure 1-3 shows a minimal implementation of an RS-232 console or modem communications port. The RS-232 to CMOS voltage-level converter is required for the proper electrical interface with the external device.

The Am186EM and Am188EM microcontrollers also include a synchronous serial interface. For more information, see Chapter 11.

1.4 THIRD-PARTY DEVELOPMENT SUPPORT PRODUCTS

The FusionE86 Program of Partnerships for Application Solutions provides the customer with an array of products designed to meet critical time-to-market needs. Products and solutions available from the AMD FusionE86 partners include emulators, hardware and software debuggers, board-level products, and software development tools, among others.

In addition, mature development tools and applications for the x86 platform are widely available in the general marketplace.

CHAPTER

2 PROGRAMMING

All members of the Am186 and Am188 family of microcontrollers, including the Am186EM and Am188EM, contain the same basic set of registers, instructions, and addressing modes, and are compatible with the original industry-standard 186/188 parts.

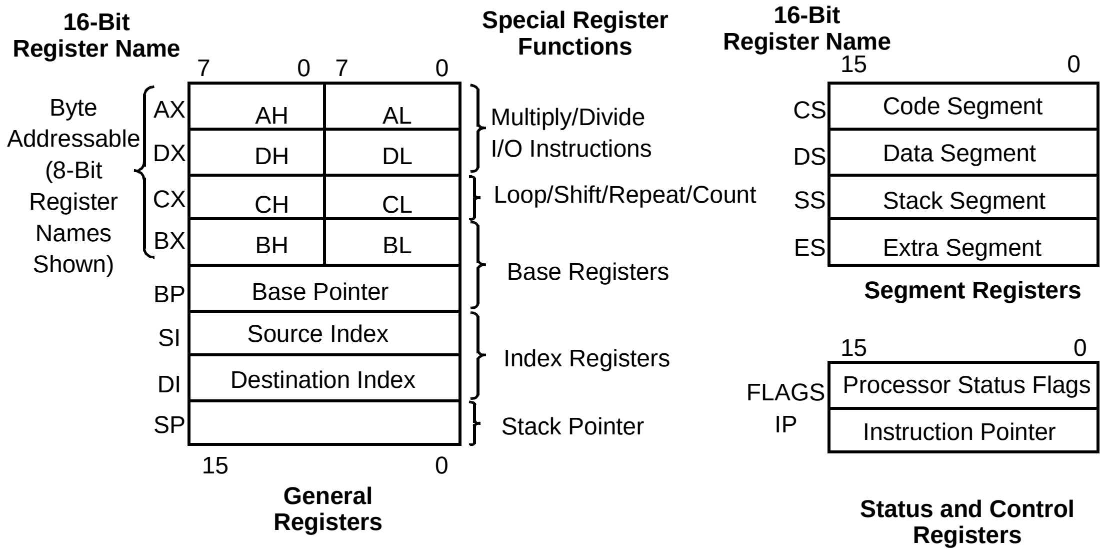
2.1 REGISTER SET

The base architecture of the Am186EM and Am188EM microcontrollers has 14 registers, as shown in Figure 2-1. These registers are grouped into the following categories:

- **General Registers**—Eight 16-bit general purpose registers can be used for arithmetic and logical operands. Four of these (AX, BX, CX, and DX) can be used as 16-bit registers or split into pairs of separate 8-bit registers (AH, AL, BH, BL, CH, CL, DH, and DL). The Destination Index (DI) and Source Index (SI) general-purpose registers are used for data movement and string instructions. The Base Pointer (BP) and Stack Pointer (SP) general-purpose registers are used for the stack segment and point to the bottom and top of the stack, respectively.
  - **Base and Index Registers**—Four of the general-purpose registers (BP, BX, DI, and SI) can also be used to determine offset addresses of operands in memory. These registers can contain base addresses or indexes to particular locations within a segment. The addressing mode selects the specific registers for operand and address calculations.
  - **Stack Pointer Register**—All stack operations (POP, POPA, POPF, PUSH, PUSHA, PUSHF) utilize the stack pointer. The Stack Pointer register is always offset from the Stack Segment (SS) register, and no segment override is allowed.
- **Segment Registers**—Four 16-bit special-purpose registers (CS, DS, ES, and SS) select, at any given time, the segments of memory that are immediately addressable for code (CS), data (DS and ES), and stack (SS) memory. (For usage, refer to section 2.2.)
- **Status and Control Registers**—Two 16-bit special-purpose registers record or alter certain aspects of the processor state—the Instruction Pointer (IP) register contains the offset address of the next sequential instruction to be executed and the Processor Status Flags (FLAGS) register contains status and control flag bits (see Figure 2-1 and Figure 2-2).

Note that the Am186EM and Am188EM microcontrollers have additional on-chip peripheral registers, which are external to the processor. These external registers are not accessible by the instruction set. However, because the processor treats these peripheral registers like memory, instructions that have operands that access memory can also access peripheral registers. The above processor registers, as well as the additional on-chip peripheral registers, are described in the chapters that follow.

Figure 2-1 Register Set

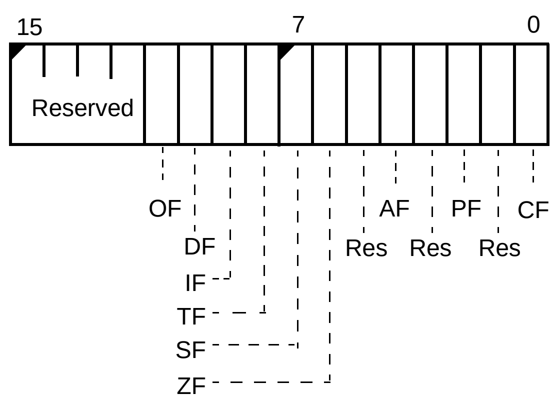


2.1.1 Processor Status Flags Register

The 16-bit processor Status Flags register (Figure 2-2) records specific characteristics of the result of logical and arithmetic instructions (bits 0, 2, 4, 6, 7, and 11) and controls the operation of the microcontroller within a given operating mode (bits 8, 9, and 10).

After an instruction is executed, the value of the flags may be set (to 1), cleared/reset (set to 0), unchanged, or undefined. The term *undefined* means that the flag value prior to the execution of the instruction is not preserved, and the value of the flag after the instruction is executed cannot be predicted.

Figure 2-2 Processor Status Flags Register (F)



Bits 15–12—Reserved

Bit 11: Overflow Flag (OF)—Set if the signed result cannot be expressed within the number of bits in the destination operand, cleared otherwise.

Bit 10: Direction Flag (DF)—Causes string instructions to auto-decrement the appropriate index registers when set. Clearing DF causes auto-increment.

Bit 9: Interrupt-Enable Flag (IF)—When set, enables maskable interrupts to cause the CPU to transfer control to a location specified by an interrupt vector.

Bit 8: Trace Flag (TF)—When set, a trace interrupt occurs after instructions execute. TF is cleared by the trace interrupt after the processor status flags are pushed onto the stack. The trace service routine can continue tracing by popping the flags back with an interrupt return (IRET) instruction.

Bit 7: Sign Flag (SF)—Set equal to high-order bit of result (0 if 0 or positive, 1 if negative).

Bit 6: Zero Flag (ZF)—Set if result is 0; cleared otherwise.

Bit 5: Reserved

Bit 4: Auxiliary Carry (AF)—Set on carry from or borrow to the low-order 4 bits of the AL general-purpose register; cleared otherwise.

Bit 3: Reserved

Bit 2: Parity Flag (PF)—Set if low-order 8 bits of result contain an even number of 1 bits; cleared otherwise.

Bit 1: Reserved

Bit 0: Carry Flag (CF)—Set on high-order bit carry or borrow; cleared otherwise.

2.2 MEMORY ORGANIZATION AND ADDRESS GENERATION

Memory is organized in sets of segments. Each segment is a linear contiguous sequence of 64K (2^{16}) 8-bit bytes. Memory is addressed using a two-component address that consists of a 16-bit segment value and a 16-bit offset. The offset is the number of bytes from the beginning of the segment (the segment address), to the data or instruction that is being accessed.

The processor forms the physical address of the target location by taking the segment address, shifting it to the left 4 bits (multiplying by 16), and adding this to the 16-bit offset. The result is the 20-bit address of the target data or instruction. This allows for a 1-Mbyte physical address size.

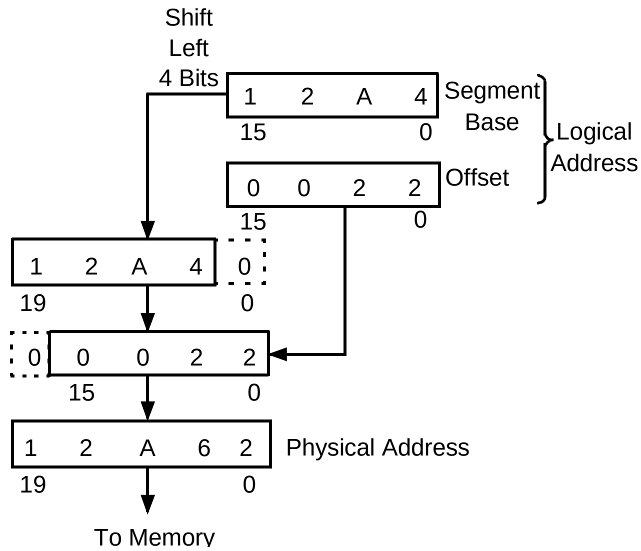
For example, if the segment register is loaded with 12A4h and the offset is 0022h, the resultant address is 12A62h (see Figure 2-3). To find the result:

1. The segment register contains 12A4h.
2. The segment register is shifted 4 places and is now 12A40h.
3. The offset is 0022h.
4. The shifted segment address (12A40h) is added to the offset (00022h) to get 12A62h.
5. This address is placed on the pins of the controller.

All instructions that address operands in memory must specify (implicitly or explicitly) a 16-bit segment value and a 16-bit offset value. The 16-bit segment values are contained in one of four internal segment registers (CS, DS, ES, and SS). See “Addressing Modes” on page 2-10 for more information on calculating the offset value. See “Segments” on page 2-8 for more information on CS, DS, ES, and SS.

In addition to memory space, all Am186 and Am188 family processors provide 64K of I/O space (see Figure 2-4).

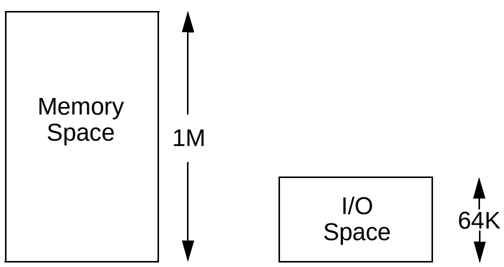
Figure 2-3 Physical Address Generation



2.3 I/O SPACE

The I/O space consists of 64K 8-bit or 32K 16-bit ports. The IN and OUT instructions address the I/O space with either an 8-bit port address specified in the instruction, or a 16-bit port address in the DX register. Eight-bit port addresses are zero-extended so that A15–A8 are Low. I/O port addresses 00F8h through 00FFh are reserved. The Am186EM and Am188EM microcontrollers provide specific instructions for addressing I/O space.

Figure 2-4 Memory and I/O Space



2.4 INSTRUCTION SET

Each member of the Am186 and Am188 family of microcontrollers, including the Am186EM and Am188EM, share the standard 186 instruction set. An instruction can reference from zero to several operands. An operand can reside in a register, in the instruction itself, or in memory. Specific operand addressing modes are discussed on page 2-10.

Table 2-1 lists the instructions for the Am186EM and Am188EM microcontrollers in alphabetical order. The Am186 and Am188 Family Instruction Set Manual, PID #21076, provides detailed information on the format and function of the following instructions.

Table 2-1 Instruction Set

Mnemonic	Instruction Name
AAA	ASCII adjust for addition
AAD	ASCII adjust for division
AAM	ASCII adjust for multiplication
AAS	ASCII adjust for subtraction
ADC	Add byte or word with carry
ADD	Add byte or word
AND	Logical AND byte or word
BOUND	Detects values outside prescribed range
CALL	Call procedure
CBW	Convert byte to word
CLC	Clear carry flag
CLD	Clear direction flag
CLI	Clear interrupt-enable flag
CMC	Complement carry flag
CMP	Compare byte or word
CMPS	Compare byte or word string
CWD	Convert word to doubleword
DAA	Decimal adjust for addition
DAS	Decimal adjust for subtraction
DEC	Decrement byte or word by 1
DIV	Divide byte or word unsigned
ENTER	Format stack for procedure entry
ESC	Escape to extension processor
HLT	Halt until interrupt or reset
IDIV	Integer divide byte or word
IMUL	Integer multiply byte or word
IN	Input byte or word
INC	Increment byte or word by 1
INS	Input bytes or word string
INT	Interrupt
INTO	Interrupt if overflow
IRET	Interrupt return
JA/JNBE	Jump if above/not below or equal
JAE/JNB	Jump if above or equal/not below

Mnemonic	Instruction Name
JB/JNAE	Jump if below/not above or equal
JBE/JNA	Jump if below or equal/not above
JC	Jump if carry
JCXZ	Jump if register CX = 0
JE/JZ	Jump if equal/zero
JG/JNLE	Jump if greater/not less or equal
JGE/JNL	Jump if greater or equal/not less
JL/JNGE	Jump if less/not greater or equal
JLE/JNG	Jump if less or equal/not greater
JMP	Jump
JNC	Jump if not carry
JNE/JNZ	Jump if not equal/not zero
JNO	Jump if not overflow
JNP/JPO	Jump if not parity/parity odd
JNS	Jump if not sign
JO	Jump if overflow
JP/JPE	Jump if parity/parity even
JS	Jump if sign
LAHF	Load AH register from flags
LDS	Load pointer using DS
LEA	Load effective address
LEAVE	Restore stack for procedure exit
LES	Load pointer using ES
LOCK	Lock bus during next instruction
LODS	Load byte or word string
LOOP	Loop
LOOPE/ LOOPZ	Loop if equal/zero
LOOPNE/ LOOPNZ	Loop if not equal/not zero
MOV	Move byte or word
MOVS	Move byte or word string
MUL	Multiply byte or word unsigned
NEG	Negate byte or word
NOP	No operation
NOT	Logical NOT byte or word

Mnemonic	Instruction Name
OR	Logical Inclusive OR byte or word
OUT	Output byte or word
POP	Pop word off stack
POPA	Pop all general register off stack
POPF	Pop flags off stack
PUSH	Push word onto stack
PUSHA	Push all general registers onto stack
PUSHF	Push flags onto stack
RCL	Rotate left through carry byte or word
RCR	Rotate right through carry byte or word
REP	Repeat
REPE/REPZ	Repeat while equal/zero
REPNE/ REPNZ	Repeat while not equal/not zero
RET0	Return from procedure
ROL	Rotate left byte or word
ROR	Rotate right byte or word
SAHF	Store AH register in flags SF, ZF, AF, PF, and CF
SAL	Shift left arithmetic byte or word
SAR	Shift right arithmetic byte or word
SBB	Subtract byte or word with borrow
SCAS	Scan byte or word string
SHL	Shift left logical byte or word
SHR	Shift right logical byte or word
STC	Set carry flag
STD	Set direction flag
STI	Set interrupt-enable flag
STOS	Store byte or word string
SUB	Subtract byte or word
TEST	Test (Logical AND, flags only set) byte or word
XCHG	Exchange byte or word
XLAT	Translate byte
XOR	Logical exclusive OR byte or word

2.5 SEGMENTS

The Am186EM and Am188EM use four segment registers:

1. **Data Segment (DS):** The processor assumes that all accesses to the program's variables are from the 64K space pointed to by the DS register. The data segment holds data, operands, etc.
2. **Code Segment (CS):** This 64K space is the default location for all instructions. All code must be executed from the code segment.
3. **Stack Segment (SS):** The processor uses the SS register to perform operations that involve the stack, such as pushes and pops. The stack segment is used for temporary space.
4. **Extra Segment (ES):** Usually this segment is used for large string operations and for large data structures. Certain string instructions assume the extra segment as the segment portion of the address. The extra segment is also used (by using segment override) as a spare data segment.

When a segment is not defined for a data movement instruction, it's assumed to be a data segment. An instruction prefix can be used to override the segment register. For speed and compact instruction encoding, the segment register used for physical address generation is implied by the addressing mode used (see Table 2-1).

Table 2-1 Segment Register Selection Rules

Memory Reference Needed	Segment Register Used	Implicit Segment Selection Rule
Local Data	Data (DS)	All data references
Instructions	Code (CS)	Instructions (including immediate data)
Stack	Stack (SS)	All stack pushes and pops Any memory references that use the BP register
External Data (Global)	Extra (ES)	All string instruction references that use the DI register as an index

2.6 DATA TYPES

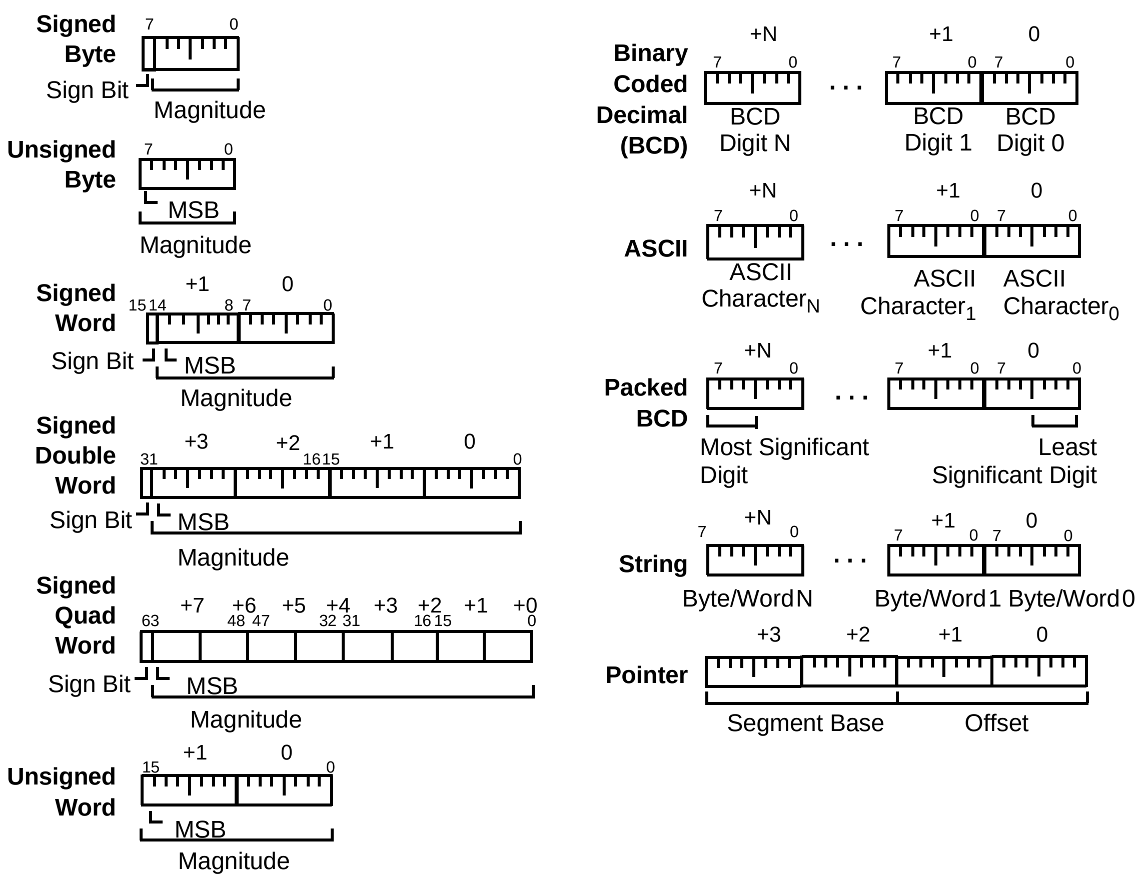
The Am186EM and Am188EM microcontrollers directly support the following data types:

- **Integer**—A signed binary numeric value contained in an 8-bit byte or a 16-bit word. All operations assume a two's complement representation.
- **Ordinal**—An unsigned binary numeric value contained in an 8-bit byte or a 16-bit word.
- **Double Word**—A signed binary numeric value contained in two sequential 16-bit addresses, or in a DX::AX register pair.
- **Quad Word**—A signed binary numeric value contained in four sequential 16-bit addresses.
- **BCD**—An unpacked byte representation of the decimal digits 0–9.
- **ASCII**—A byte representation of alphanumeric and control characters using the ASCII standard of character representation.
- **Packed BCD**—A packed byte representation of two decimal digits (0–9). One digit is stored in each nibble (4 bits) of the byte.

- **String**—A contiguous sequence of bytes or words. A string can contain from 1 byte up to 64 Kbyte.
- **Pointer**—A 16-bit or 32-bit quantity, composed of a 16-bit offset component or a 16-bit segment base component plus a 16-bit offset component.

In general, individual data elements must fit within defined segment limits. Figure 2-5 graphically represents the data types supported by the Am186EM and Am188EM microcontrollers.

Figure 2-5 Supported Data Types



2.7 ADDRESSING MODES

The Am186EM and Am188EM microcontrollers use eight categories of addressing modes to specify operands. Two addressing modes are provided for instructions that operate on register or immediate operands; six modes are provided to specify the location of an operand in a memory segment.

Register and Immediate Operands

- **Register Operand Mode**—The operand is located in one of the 8- or 16-bit registers.
- **Immediate Operand Mode**—The operand is included in the instruction.

Memory Operands

A memory-operand address consists of two 16-bit components: a segment value and an offset. The segment value is supplied by a 16-bit segment register either implicitly chosen by the addressing mode or explicitly chosen by a segment override prefix. The offset, also called the effective address, is calculated by summing any combination of the following three address elements:

1. **Displacement**—an 8-bit or 16-bit immediate value contained in the instruction
2. **Base**—contents of either the BX or BP base registers
3. **Index**—contents of either the SI or DI index registers

Any carry from the 16-bit addition is ignored. Eight-bit displacements are sign-extended to 16-bit values.

Combinations of the above three address elements define the following six memory addressing modes (see Table 2-2):

1. **Direct Mode**—The operand offset is contained in the instruction as an 8- or 16-bit displacement element.
2. **Register Indirect Mode**—The operand offset is in one of the registers BP, BX, DI, or SI.
3. **Based Mode**—The operand offset is the sum of an 8- or 16-bit displacement and the contents of a base register (BX or BP).
4. **Indexed Mode**—The operand offset is the sum of an 8- or 16-bit displacement and the contents of an index register (DI or SI).
5. **Based Indexed Mode**—The operand offset is the sum of the contents of a base register (BP or BX) and an index register (DI or SI).
6. **Based Indexed Mode with Displacement**—The operand offset is the sum of a base register's contents, an index register's contents, and an 8-bit or 16-bit displacement.

Table 2-2 Memory Addressing Mode Examples

Addressing Mode	Example
Direct	`mov ax, ds:4`
Register Indirect	`mov ax, [si]`
Based	`mov ax, [bx]4`
Indexed	`mov ax, [si]4`
Based Indexed	`mov ax, [si][bx]`
Based Indexed with Displacement	`mov ax, [si][bx]4`

CHAPTER

3 SYSTEM OVERVIEW

This chapter contains descriptions of the Am186EM and Am188EM microcontroller pins, the bus interface unit, the clock and power management unit, and power-save operation.

3.1 PIN DESCRIPTIONS

Pin Terminology

The following terms are used to describe the pins:

Input—An input-only pin.

Output—An output-only pin.

Input/Output—A pin that can be either input or output.

Synchronous—Synchronous inputs must meet setup and hold times in relation to CLKOUTA. Synchronous outputs are synchronous to CLKOUTA.

Asynchronous—Inputs or outputs that are asynchronous to CLKOUTA.

A19–A0 **Address Bus (output, three-state, synchronous)**

The A19–A0 pins supply nonmultiplexed memory or I/O addresses to the system one-half of a CLKOUTA period earlier than the multiplexed address and data bus (AD15–AD0 on the Am186EM or AO15–AO8 and AD7–AD0 on the Am188EM). During a bus hold or reset condition, the address bus is in a high-impedance state.

AD7–AD0 **Address and Data Bus (input/output, three-state, synchronous, level-sensitive)**

These time-multiplexed pins supply partial memory or I/O addresses, as well as data, to the system. This bus supplies the low-order 8 bits of an address to the system during the first period of a bus cycle (t_1), and it supplies data to the system during the remaining periods of that cycle (t_2, t_3, and t_4).

The address phase of these pins can be disabled. See the $\overline{\text{ADEN}}$ description with the $\overline{\text{BHE}}$/$\overline{\text{ADEN}}$ pin. When $\overline{\text{WLB}}$ is not asserted, these pins are three-stated during t_2, t_3, and t_4.

During a bus hold or reset condition, the address and data bus is in a high-impedance state.

During a power-on reset, the address and data bus pins (AD15–AD0 for the Am186EM, AO15–AO8 and AD7–AD0 for the Am188EM) can also be used to load system configuration information into the internal Reset Configuration register.

AD15–AD8

Address and Data Bus, Am186EM Microcontroller Only (input/output, three-state, synchronous, level-sensitive)

AD15–AD8—These time-multiplexed pins supply partial memory or I/O addresses, as well as data, to the system. This bus supplies an address to the system during the first period of a bus cycle (t_1), and it supplies data to the system during the remaining periods of that cycle (t_2, t_3, and t_4).

The address phase of these pins can be disabled. See the $\overline{\text{ADEN}}$ description with the $\overline{\text{BHE}}/\overline{\text{ADEN}}$ pin. When $\overline{\text{WHB}}$ is not asserted, these pins are three-stated during t_2, t_3, and t_4.

During a bus hold or reset condition, the address and data bus is in a high-impedance state. During a power-on reset, the address and data bus pins (AD15–AD0 for the Am186EM, AO15–AO8 and AD7–AD0 for the Am188EM) can also be used to load system configuration information into the internal Reset Configuration register.

AO15–AO8

Address-Only Bus, Am188EM Microcontroller Only (output, three-state, synchronous, level-sensitive)

AO15–AO8—The address-only bus (AO15–AO8) contains valid high-order address bits from bus cycles t_1–t_4. These outputs are floated during a bus hold or reset.

On the Am188EM microcontroller, AO15–AO8 combine with AD7–AD0 to form a complete multiplexed address bus while AD7–AD0 is the 8-bit data bus.

The address phase of these pins can be disabled during t_1. See the $\overline{\text{ADEN}}$ description with the $\overline{\text{BHE}}/\overline{\text{ADEN}}$ pin.

During a power-on reset on the Am188EM microcontroller, the AO15–AO8 and AD7–AD0 pins can also be used to load system configuration information into an internal register for later use.

ALE

Address Latch Enable (output, synchronous)

ALE—This pin indicates to the system that an address appears on the address and data bus (AD15–AD0 for the Am186EM or AO15–AO8 and AD7–AD0 for the Am188EM). The address is guaranteed valid on the trailing edge of ALE.

ARDY

Asynchronous Ready (input, asynchronous, level-sensitive)

This pin indicates to the microcontroller that the addressed memory space or I/O device will complete a data transfer. The ARDY pin accepts a rising edge that is asynchronous to CLKOUTA and is active High. The falling edge of ARDY must be synchronized to CLKOUTA. To always assert the ready condition to the microcontroller, tie ARDY High. If the system does not use ARDY, tie the pin Low to yield control to SRDY.

$\overline{BHE}$/$\overline{ADEN}$

Bus High Enable, Am186EM Microcontroller Only (three-state, output, synchronous)
Address Enable, Am186EM Microcontroller Only (input, internal pullup)

$\overline{BHE}$—During a memory access, this pin and the least significant address bit (AD0 and A0) indicate to the system which bytes of the data bus (upper, lower, or both) participate in a bus cycle. The $\overline{BHE}$/$\overline{ADEN}$ and AD0 pins are encoded as shown in the following table.

$\overline{BHE}$	AD0	Type of Bus Cycle
0	0	Word Transfer
0	1	High Byte Transfer (Bits 15–8)
1	0	Low Byte Transfer (Bits 7–0)
1	1	Refresh

$\overline{BHE}$ is asserted during t_1 and remains asserted through t_3 and t_W. $\overline{BHE}$ does not need to be latched. $\overline{BHE}$ floats during bus hold and reset.

On the Am186EM microcontroller, $\overline{WLB}$ and $\overline{WHB}$ implement the functionality of $\overline{BHE}$ and AD0 for high and low byte write enables.

$\overline{BHE}$/$\overline{ADEN}$ also signals DRAM refresh cycles when using the multiplexed address and data (AD) bus. A refresh cycle is indicated when both $\overline{BHE}$/$\overline{ADEN}$ and AD0 are High. During refresh cycles, the A bus and the AD bus are not guaranteed to provide the same address during the address phase of the AD bus cycle. For this reason, the A0 signal cannot be used in place of the AD0 signal to determine refresh cycles. PSRAM refreshes also provide an additional $\overline{RFSH}$ signal (see the $\overline{MCS3}$/$\overline{RFSH}$ pin description).

$\overline{ADEN}$—If $\overline{BHE}$/$\overline{ADEN}$ is held High or left floating during power-on reset, the address portion of the AD bus (AD15–AD0) is enabled or disabled during LCS and UCS bus cycles based on the DA bit in the Upper Memory Chip Select (UMCS) and Low Memory Chip Select (LMCS) registers. If the DA bit is set, the memory address is accessed on the A19–A0 pins. This mode of operation reduces power consumption.

If $\overline{BHE}$/$\overline{ADEN}$ is held Low on power-on reset, the AD bus always drives both addresses and data. The pin is sampled one crystal clock cycle after the rising edge of $\overline{RES}$.

See section 5.5.1 and section 5.5.2 for additional information on enabling and disabling the AD bus during the address phase of a bus cycle.

CLKOUTA

Clock Output A (output, synchronous)

This pin supplies the internal clock to the system. Depending on the value of the Power-Save Control (PDCON) register, CLKOUTA operates at either the crystal input frequency (X1), the power-save frequency, or is three-stated. CLKOUTA remains active during reset and bus hold conditions.

CLKOUTB — **Clock Output B (output, synchronous)**

This pin supplies an additional clock to the system. Depending on the value of the Power-Save Control (PDCON) register, CLKOUTB operates at either the crystal input frequency (X1), the power-save frequency, or is three-stated. CLKOUTB remains active during reset and bus hold conditions.

$\overline{\text{DEN}}$ — **Data Enable (output, three-state, synchronous)**

This pin supplies an output enable to an external data-bus transceiver. $\overline{\text{DEN}}$ is asserted during memory, I/O, and interrupt acknowledge cycles. $\overline{\text{DEN}}$ is deasserted when DT/$\overline{\text{R}}$ changes state. $\overline{\text{DEN}}$ floats during a bus hold or reset condition.

DRQ1–DRQ0 — **DMA Requests (input, synchronous, level-sensitive)**

These pins indicate to the microcontroller that an external device is ready for DMA channel 1 or 0 to perform a transfer. DRQ1–DRQ0 are level triggered and internally synchronized.

The DRQ signals are not latched and must remain active until serviced.

DT/$\overline{\text{R}}$ — **Data Transmit or Receive (output, three-state, synchronous)**

This pin indicates which direction data should flow through an external data-bus transceiver. When DT/$\overline{\text{R}}$ is asserted High, the microcontroller transmits data. When this pin is deasserted Low, the microcontroller receives data. DT/$\overline{\text{R}}$ floats during a bus hold or reset condition.

GND — **Ground**

These pins connect the system ground to the microcontroller.

HLDA — **Bus Hold Acknowledge (output, synchronous)**

This pin is asserted High to indicate to an external bus master that the microcontroller has relinquished control of the local bus. When an external bus master requests control of the local bus (by asserting HOLD), the microcontroller completes the bus cycle in progress and then relinquishes control of the bus to the external bus master by asserting HLDA and floating $\overline{\text{DEN}}$, $\overline{\text{RD}}$, $\overline{\text{WR}}$, $\overline{\text{S2}}$–$\overline{\text{S0}}$, AD15–AD0, S6, A19–A0, $\overline{\text{BHE}}$, $\overline{\text{WHB}}$, $\overline{\text{WLB}}$, and DT/$\overline{\text{R}}$, and then driving the chip selects $\overline{\text{UCS}}$, $\overline{\text{LCS}}$, $\overline{\text{MCS3}}$–$\overline{\text{MCS0}}$, $\overline{\text{PCS6}}$–$\overline{\text{PCS5}}$, and $\overline{\text{PCS3}}$–$\overline{\text{PCS0}}$ High.

When the external bus master has finished using the local bus, it indicates this to the microcontroller by deasserting HOLD. The microcontroller responds by deasserting HLDA.

If the microcontroller requires access to the bus (i.e., for refresh), it will deassert HLDA before the external bus master deasserts HOLD. The external bus master must be able to deassert HOLD and allow the microcontroller access to the bus.

HOLD — **Bus Hold Request (input, synchronous, level-sensitive)**

This pin indicates to the microcontroller that an external bus master needs control of the local bus. For more information, see the HLDA pin description.

The Am186EM and Am188EM microcontrollers' HOLD latency time, that is, the time between HOLD request and HOLD acknowledge, is a function of the activity occurring in the processor when the HOLD request is received. A HOLD request is second only to DRAM refresh requests in priority of activity requests received by the processor. This implies that if a HOLD request is received just as a DMA transfer begins, the HOLD latency can be as great as 4 bus cycles. This occurs if a DMA word transfer operation is taking place (Am186EM microcontroller only) from an odd address to an odd address. This is a total of 16 clock cycles or more if wait states are required. In addition, if locked transfers are performed, the HOLD latency time is increased by the length of the locked transfer.

INT0 — **Maskable Interrupt Request 0 (input, asynchronous)**

This pin indicates to the microcontroller that an interrupt request has occurred. If the INT0 pin is not masked, the microcontroller transfers program execution to the location specified by the INT0 vector in the microcontroller interrupt vector table. Interrupt requests are synchronized internally, and can be edge-triggered or level-triggered. To guarantee the interrupt is recognized, the device issuing the request must continue asserting INT0 until the request is acknowledged.

INT1/$\overline{\text{SELECT}}$ — **Maskable Interrupt Request 1 (input, asynchronous)**
Slave Select (input, asynchronous)

INT1—This pin indicates to the microcontroller that an interrupt request has occurred. If the INT1 pin is not masked, the microcontroller transfers program execution to the location specified by the INT1 vector in the microcontroller interrupt vector table. Interrupt requests are synchronized internally, and can be edge-triggered or level-triggered. To guarantee the interrupt is recognized, the device issuing the request must continue asserting INT1 until the request is acknowledged.

$\overline{\text{SELECT}}$—When the microcontroller interrupt control unit is operating as a slave to an external master interrupt controller, this pin indicates to the microcontroller that an interrupt type appears on the address and data bus. The INT0 pin must indicate to the microcontroller that an interrupt has occurred before the $\overline{\text{SELECT}}$ pin indicates to the microcontroller that the interrupt type appears on the bus.

INT2/$\overline{\text{INTA}}$0 — **Maskable Interrupt Request 2 (input, asynchronous)**
Interrupt Acknowledge 0 (output, synchronous)

INT2—This pin indicates to the microcontroller that an interrupt request has occurred. If the INT2 pin is not masked, the microcontroller transfers program execution to the location specified by the INT2 vector in the microcontroller interrupt vector table. Interrupt requests are synchronized internally, and can be edge-triggered or level-triggered. To guarantee the interrupt is recognized, the device issuing the request must continue asserting INT2 until the request is acknowledged. INT2 becomes $\overline{\text{INTA}}$0 when INT0 is configured in cascade mode.

$\overline{\text{INTA}}$0—When the microcontroller interrupt control unit is operating in cascade mode, this pin indicates to the system that the microcontroller needs an interrupt type to process the interrupt request on INT0. The peripheral issuing the interrupt request must provide the microcontroller with the corresponding interrupt type.

INT3/$\overline{\text{INTA}}$1/IRQ **Maskable Interrupt Request 3 (input, asynchronous)**
Interrupt Acknowledge 1 (output, synchronous)
Slave Interrupt Request (output, synchronous)

INT3—This pin indicates to the microcontroller that an interrupt request has occurred. If the INT3 pin is not masked, the microcontroller then transfers program execution to the location specified by the INT3 vector in the microcontroller interrupt vector table. Interrupt requests are synchronized internally, and they can be edge-triggered or level-triggered. To guarantee the interrupt is recognized, the device issuing the request must continue asserting INT3 until the request is acknowledged. INT3 becomes $\overline{\text{INTA}}$1 when INT1 is configured in cascade mode.

$\overline{\text{INTA}}$1—When the microcontroller interrupt control unit is operating in cascade mode, this pin indicates to the system that the microcontroller needs an interrupt type to process the interrupt request on INT1. The peripheral issuing the interrupt request must provide the microcontroller with the corresponding interrupt type.

IRQ—When the microcontroller interrupt control unit is operating as a slave to an external master interrupt controller, this pin lets the microcontroller issue an interrupt request to the external master interrupt controller.

INT4 **Maskable Interrupt Request 4 (input, asynchronous)**

This pin indicates to the microcontroller that an interrupt request has occurred. If the INT4 pin is not masked, the microcontroller then transfers program execution to the location specified by the INT4 vector in the microcontroller interrupt vector table. Interrupt requests are synchronized internally, and they can be edge-triggered or level-triggered. To guarantee the interrupt is recognized, the device issuing the request must continue asserting INT4 until the request is acknowledged.

$\overline{\text{LCS}}$/$\overline{\text{ONCE}}$0 **Lower Memory Chip Select (output, synchronous, internal pullup)**
ONCE Mode Request 0 (input)

$\overline{\text{LCS}}$—This pin indicates to the system that a memory access is in progress to the lower memory block. The base address and size of the lower memory block are programmable up to 512 Kbytes. $\overline{\text{LCS}}$ is held High during a bus hold condition.

$\overline{\text{ONCE}}$0—During reset this pin and $\overline{\text{UCS}}$/$\overline{\text{ONCE}}$1 indicate to the microcontroller the mode in which it should operate. $\overline{\text{ONCE}}$0 and $\overline{\text{ONCE}}$1 are sampled on the rising edge of $\overline{\text{RES}}$. If both pins are asserted Low, the microcontroller enters ONCE mode; otherwise, it operates normally.

In ONCE mode, all pins assume a high-impedance state and remain in that state until a subsequent reset occurs. To guarantee that the microcontroller does not inadvertently enter ONCE mode, $\overline{\text{ONCE}}$0 has a weak internal pullup resistor that is active only during a reset.

MCS3/RFSH — **Midrange Memory Chip Select 3 (output, synchronous, internal pullup) Automatic Refresh (output, synchronous)**

MCS3—This pin indicates to the system that a memory access is in progress to the fourth region of the midrange memory block. The base address and size of the midrange memory block are programmable. MCS3 is held High during a bus hold condition. In addition, this pin has a weak internal pullup resistor that is active during reset.

RFSH—This pin provides a signal timed for auto refresh to PSRAM devices. It is only enabled to function as a refresh pulse when the PSRAM mode bit is set in the LMCS register. An active Low pulse is generated for 1.5 clock cycles with an adequate deassertion period to ensure overall auto refresh cycle time is met.

MCS2–MCS0 — **Midrange Memory Chip Selects (output, synchronous, internal pullup)**

These pins indicate to the system that a memory access is in progress to the corresponding region of the midrange memory block. The base address and size of the midrange memory block are programmable. MCS2–MCS0 are held High during a bus hold condition. In addition, they have weak internal pullup resistors that are active during a reset.

NMI — **Nonmaskable Interrupt (input, synchronous, edge-sensitive)**

This pin indicates to the microcontroller that an interrupt request has occurred. The NMI signal is the highest priority hardware interrupt and, unlike the INT4–INT0 pins, cannot be masked. The microcontroller always transfers program execution to the location specified by the nonmaskable interrupt vector in the microcontroller interrupt vector table when NMI is asserted.

Although NMI is the highest priority interrupt source, it does not participate in the priority resolution process of the maskable interrupts. There is no bit associated with NMI in the interrupt in-service or interrupt request registers. This means that a new NMI request can interrupt an executing NMI interrupt service routine. As with all hardware interrupts, the IF (interrupt flag) is cleared when the processor takes the interrupt, disabling the maskable interrupt sources. However, if maskable interrupts are re-enabled by software in the NMI interrupt service routine, via the STI instruction for example, the fact that an NMI is currently in service will not have any effect on the priority resolution of maskable interrupt requests. For this reason, it is strongly advised that the interrupt service routine for NMI does not enable the maskable interrupts.

An NMI transition from Low to High is latched and synchronized internally, and it initiates the interrupt at the next instruction boundary. To guarantee that the interrupt is recognized, the NMI pin must be asserted for at least one CLKOUTA period.

PCS3–PCS0 — **Peripheral Chip Selects (output, synchronous)**

These pins indicate to the system that a memory access is in progress to the corresponding region of the peripheral memory block (either I/O or memory address space). The base address of the peripheral memory block is programmable. PCS3–PCS0 are held High during a bus hold or reset condition. Unlike the $\overline{\text{UCS}}$ and $\overline{\text{LCS}}$ chip selects, the $\overline{\text{PCS}}$ outputs assert with the multiplexed AD address bus.

Note: $\overline{\text{PCS}}$4 is not available on the Am186EM and Am188EM microcontrollers. Note also that each peripheral chip select asserts over a 256-byte address range, which is twice the address range covered by peripheral chip selects in the 80C186 and 80C188 microcontrollers.

$\overline{\text{PCS}}$5/A1

Peripheral Chip Select 5 (output, synchronous)
Latched Address Bit 1 (output, synchronous)

$\overline{\text{PCS}}$5—This pin indicates to the system that a memory access is in progress to the sixth region of the peripheral memory block (either I/O or memory address space). The base address of the peripheral memory block is programmable. $\overline{\text{PCS}}$5 is held High during a bus hold or reset condition. It is also held High during reset.

Note: Unlike the $\overline{\text{UCS}}$ and $\overline{\text{LCS}}$ chip selects, the $\overline{\text{PCS}}$ outputs assert with the multiplexed AD address bus. Note also that each peripheral chip select asserts over a 256-byte address range, which is twice the address range covered by peripheral chip selects in the 80C186 and 80C188 microcontrollers.

A1—When the EX bit in the $\overline{\text{MCS}}$ and $\overline{\text{PCS}}$ Auxiliary register is 0, this pin supplies an internally latched address bit 1 to the system. During a bus hold condition, A1 retains its previously latched value.

$\overline{\text{PCS}}$6/A2

Peripheral Chip Select 6 (output, synchronous)
Latched Address Bit 2 (output, synchronous)

$\overline{\text{PCS}}$6—This pin indicates to the system that a memory access is in progress to the seventh region of the peripheral memory block (either I/O or memory address space). The base address of the peripheral memory block is programmable. $\overline{\text{PCS}}$6 is held High during a bus hold or reset condition.

Note: Unlike the $\overline{\text{UCS}}$ and $\overline{\text{LCS}}$ chip selects, the $\overline{\text{PCS}}$ outputs assert with the multiplexed AD address bus. Note also that each peripheral chip select asserts over a 256-byte address range, which is twice the address range covered by peripheral chip selects in the original 80C186 and 80C188 microcontrollers.

A2—When the EX bit in the $\overline{\text{MCS}}$ and $\overline{\text{PCS}}$ Auxiliary register is 0, this pin supplies an internally latched address bit 2 to the system. During a bus hold condition, A2 retains its previously latched value.

PIO31–PIO0 (Shared)

Programmable I/O Pins (input/output, asynchronous, open-drain)

The Am186EM and Am188EM microcontrollers provide 32 individually programmable I/O pins. The pins that are multiplexed with PIO31–PIO0 are listed in Table 3-1 and Table 3-2. Each PIO can be programmed with the following attributes: PIO function (enabled/disabled), direction (input/output), and weak pullup or pulldown. See Chapter 12 for the PIO control registers.

After power-on reset, the PIO pins default to various configurations. The column titled *Power-On Reset State* in Table 3-1 and Table 3-2 lists the defaults for the PIOs. The system initialization code must reconfigure any PIOs as required.

The A19–A17 address pins default to normal operation on power-on reset, allowing the processor to correctly begin fetching instructions at the boot address FFFF0h. The DT/$\overline{\text{R}}$, $\overline{\text{DEN}}$, and SRDY pins also default to normal operation on power-on reset.

Table 3-1 PIO Pin Assignments—Numeric Listing

PIO No.	Associated Pin	Power-On Reset Status
0	TMRIN1	Input with pullup
1	TMROUT1	Input with pulldown
2	$\overline{\text{PCS}}$6/A2	Input with pullup
3	$\overline{\text{PCS}}$5/A1	Input with pullup
4	DT/$\overline{\text{R}}$	Normal operation[3]
5	$\overline{\text{DEN}}$	Normal operation[3]
6	SRDY	Normal operation[4]
7[1]	A17	Normal operation[3]
8[1]	A18	Normal operation[3]
9[1]	A19	Normal operation[3]
10	TMROUT0	Input with pulldown
11	TMRIN0	Input with pullup
12	DRQ0	Input with pullup
13	DRQ1	Input with pullup
14	$\overline{\text{MCS}}$0	Input with pullup
15	$\overline{\text{MCS}}$1	Input with pullup
16	$\overline{\text{PCS}}$0	Input with pullup
17	$\overline{\text{PCS}}$1	Input with pullup
18	$\overline{\text{PCS}}$2	Input with pullup
19	$\overline{\text{PCS}}$3	Input with pullup
20	SCLK	Input with pullup
21	SDATA	Input with pullup
22	SDEN0	Input with pulldown
23	SDEN1	Input with pulldown
24	$\overline{\text{MCS}}$2	Input with pullup
25	$\overline{\text{MCS}}$3/$\overline{\text{RFSH}}$	Input with pullup
26[1,2]	$\overline{\text{UZI}}$	Input with pullup
27	TXD	Input with pullup
28	RXD	Input with pullup
29[1,2]	S6/$\overline{\text{CLKDIV}}$2	Input with pullup
30	INT4	Input with pullup
31	INT2	Input with pullup

Notes:

1. *These pins are used by emulators. (Emulators also use* $\overline{\text{S}}$2–$\overline{\text{S}}$0, $\overline{\text{RES}}$*, NMI, CLKOUTA,* $\overline{\text{BHE}}$*, ALE, AD15–AD0, and A16–A0.*
2. *These pins revert to normal operation if* $\overline{\text{BHE}}$/$\overline{\text{ADEN}}$ *(Am186EM) or* $\overline{\text{RFSH}}$2/$\overline{\text{ADEN}}$ *(Am188EM) is held Low during power-on reset.*
3. *When used as a PIO, input with pullup option available.*
4. *When used as a PIO, input with pulldown option available.*

Table 3-2 PIO Pin Assignments—Alphabetic Listing

Associated Pin	PIO No.	Power-On Reset Status
A17[1]	7	Normal operation[3]
A18[1]	8	Normal operation[3]
A19[1]	9	Normal operation[3]
$\overline{\text{DEN}}$	5	Normal operation[3]
DRQ0	12	Input with pullup
DRQ1	13	Input with pullup
DT/$\overline{\text{R}}$	4	Normal operation[3]
INT2	31	Input with pullup
INT4	30	Input with pullup
$\overline{\text{MCS}}$0	14	Input with pullup
$\overline{\text{MCS}}$1	15	Input with pullup
$\overline{\text{MCS}}$2	24	Input with pullup
$\overline{\text{MCS}}$3/$\overline{\text{RFSH}}$	25	Input with pullup
$\overline{\text{PCS}}$0	16	Input with pullup
$\overline{\text{PCS}}$1	17	Input with pullup
$\overline{\text{PCS}}$2	18	Input with pullup
$\overline{\text{PCS}}$3	19	Input with pullup
$\overline{\text{PCS}}$5/A1	3	Input with pullup
$\overline{\text{PCS}}$6/A2	2	Input with pullup
RXD	28	Input with pullup
S6/$\overline{\text{CLKDIV2}}$[1,2]	29	Input with pullup
SCLK	20	Input with pullup
SDATA	21	Input with pullup
SDEN0	22	Input with pulldown
SDEN1	23	Input with pulldown
SRDY	6	Normal operation[4]
TMRIN0	11	Input with pullup
TMRIN1	0	Input with pullup
TMROUT0	10	Input with pulldown
TMROUT1	1	Input with pulldown
TXD	27	Input with pullup
$\overline{\text{UZI}}$[1,2]	26	Input with pullup

Notes:

1. *These pins are used by emulators. (Emulators also use* $\overline{\text{S}}$2–$\overline{\text{S}}$0, $\overline{\text{RES}}$, *NMI, CLKOUTA,* $\overline{\text{BHE}}$, *ALE, AD15–AD0, and A16–A0.*
2. *These pins revert to normal operation if* $\overline{\text{BHE}}$/$\overline{\text{ADEN}}$ *(Am186EM) or* $\overline{\text{RFSH}}$2/$\overline{\text{ADEN}}$ *(Am188EM) is held Low during power-on reset.*
3. *When used as a PIO, input with pullup option available.*
4. *When used as a PIO, input with pulldown option available.*

$\overline{RD}$ — **Read Strobe (output, synchronous, three-state)**

$\overline{RD}$—This pin indicates to the system that the microcontroller is performing a memory or I/O read cycle. $\overline{RD}$ is guaranteed not to be asserted before the address and data bus is floated during the address-to-data transition. $\overline{RD}$ floats during a bus hold condition.

$\overline{RES}$ — **Reset (input, asynchronous, level-sensitive)**

This pin causes the microcontroller to perform a reset. When $\overline{RES}$ is asserted, the microcontroller immediately terminates its present activity, clears its internal logic, and CPU control is transferred to the reset address FFFF0h. $\overline{RES}$ must be held Low for at least 1 ms. The assertion of $\overline{RES}$ can be asynchronous to CLKOUTA because $\overline{RES}$ is synchronized internally. For proper initialization, V_{CC} must be within specifications, and CLKOUTA must be stable for more than four CLKOUTA periods during which $\overline{RES}$ is asserted. The microcontroller begins fetching instructions approximately 6.5 CLKOUTA periods after $\overline{RES}$ is deasserted. This input is provided with a Schmitt trigger to facilitate power-on $\overline{RES}$ generation via an RC network.

$\overline{RFSH2}/\overline{ADEN}$ — **Refresh 2 (three-state, output, synchronous)**
Address Enable (input, internal pullup)

$\overline{RFSH2}$—Available on the Am188EM microcontroller only, $\overline{RFSH2}/\overline{ADEN}$ is asserted Low to signify a DRAM refresh bus cycle. The use of $\overline{RFSH2}/\overline{ADEN}$ to signal a refresh is not valid when PSRAM mode is selected. Instead, the $\overline{MCS3}/\overline{RFSH}$ signal is provided to the PSRAM.

$\overline{ADEN}$—If $\overline{RFSH2}/\overline{ADEN}$ is held High or left floating on power-on reset, the AD bus (AO15–AO8 and AD7–AD0) is enabled or disabled during the address portion of LCS and UCS bus cycles based on the DA bit in the LMCS and UMCS registers. If the DA bit is set, the memory address is accessed on the A19–A0 pins. This mode of operation reduces power consumption. There is a weak internal pullup resistor on $\overline{RFSH2}/\overline{ADEN}$, so no external pullup is required.

If $\overline{RFSH2}/\overline{ADEN}$ is held Low on power-on reset, the AD bus drives both addresses and data. The pin is sampled one crystal clock cycle after the rising edge of $\overline{RES}$. $\overline{RFSH2}/\overline{ADEN}$ is three-stated during bus holds and ONCE mode.

See section 5.5.1 and section 5.5.2 for additional information on enabling and disabling the AD bus during the address phase of a bus cycle.

RXD — **Receive Data (input, asynchronous)**

This pin supplies asynchronous serial receive data to the microcontroller UART.

$\overline{S2}$–$\overline{S0}$ — **Bus Cycle Status (output, three-state, synchronous)**

These pins indicate to the system the type of bus cycle in progress. $\overline{S2}$ can be used as a logical memory or I/O indicator, and $\overline{S1}$ can be used as a data transmit or receive indicator. $\overline{S2}$–$\overline{S0}$ float during bus hold and hold acknowledge conditions. The $\overline{S2}$–$\overline{S0}$ pins are encoded as shown in the following table.

$\overline{\text{S2}}$	$\overline{\text{S1}}$	$\overline{\text{S0}}$	**Bus Cycle**
0	0	0	Interrupt acknowledge
0	0	1	Read data from I/O
0	1	0	Write data to I/O
0	1	1	Halt
1	0	0	Instruction fetch
1	0	1	Read data from memory
1	1	0	Write data to memory
1	1	1	None (passive)

S6/$\overline{\text{CLKDIV}}$2 — **Bus Cycle Status Bit 6 (output, synchronous)**
Clock Divide by 2 (input, internal pullup)

S6—During the second and remaining periods of a cycle (t_2, t_3, and t_4), this pin is asserted High to indicate a DMA-initiated bus cycle. During a bus hold or reset condition, S6 floats.

$\overline{\text{CLKDIV}}$2—If S6/$\overline{\text{CLKDIV}}$2 is held Low during power-on reset, the chip enters clock divide-by-2 mode where the processor clock is derived by dividing the external clock input by 2. If this mode is selected, the PLL is disabled. The pin is sampled on the rising edge of $\overline{\text{RES}}$.

If S6 is to be used as PIO29 in input mode, the device driving PIO29 must not drive the pin Low during power-on reset. S6/PIO29 defaults to a PIO input with pullup, so the pin does not need to be driven High externally.

SCLK — **Serial Clock (output, synchronous, three-state)**

This pin supplies the synchronous serial interface (SSI) clock to a slave device, allowing transmit and receive operations to be synchronized between the microcontroller and the slave. SCLK is derived from the microcontroller internal clock and then divided by 2, 4, 8, or 16, depending on register settings. An access to any of the SSR or SSD registers activates SCLK for eight SCLK cycles (see Figure 11-5 and Figure 11-6 on page 11-8). When SCLK is inactive, it is held High by the microcontroller.

SDATA — **Serial Data (input/output, synchronous)**

This pin transmits and receives synchronous serial interface (SSI) data to and from a slave device. When SDATA is inactive, a weak keeper holds the last value of SDATA on the pin.

SDEN1–SDEN0 — **Serial Data Enables (output, synchronous)**

These pins enable data transfers on ports 1 and 0 of the synchronous serial interface (SSI). The microcontroller asserts either SDEN1 or SDEN0 at the beginning of a transfer and deasserts it after the transfer is complete. When SDEN1–SDEN0 are inactive, they are held Low by the microcontroller.

SRDY **Synchronous Ready (input, synchronous, level-sensitive)**

This pin indicates to the microcontroller that the addressed memory space or I/O device will complete a data transfer. The SRDY pin accepts an active-High input synchronized to CLKOUTA. Using SRDY instead of ARDY allows a relaxed system timing because of the elimination of the one-half clock period required to internally synchronize ARDY. To always assert the ready condition to the microcontroller, tie SRDY High. If the system does not use SRDY, tie the pin Low to yield control to ARDY.

TMRIN0 **Timer Input 0 (input, synchronous, edge-sensitive)**

This pin supplies a clock or control signal to the internal microcontroller timer 0. After internally synchronizing a Low-to-High transition on TMRIN0, the microcontroller increments the timer. TMRIN0 must be tied High if not being used.

TMRIN1 **Timer Input 1 (input, synchronous, edge-sensitive)**

This pin supplies a clock or control signal to the internal microcontroller timer 1. After internally synchronizing a Low-to-High transition on TMRIN1, the microcontroller increments the timer. TMRIN1 must be tied High if not being used.

TMROUT0 **Timer Output 0 (output, synchronous)**

This pin supplies to the system either a single pulse or a continuous waveform with a programmable duty cycle. TMROUT0 is floated during a bus hold or reset.

TMROUT1 **Timer Output 1 (output, synchronous)**

This pin supplies to the system either a single pulse or a continuous waveform with a programmable duty cycle. It can also be programmed as a watchdog timer. TMROUT1 is floated during a bus hold or reset.

TXD **Transmit Data (output, asynchronous)**

This pin supplies asynchronous serial transmit data from the microcontroller UART to the system.

$\overline{\text{UCS}}$/$\overline{\text{ONCE}}$1 **Upper Memory Chip Select (output, synchronous)**
ONCE Mode Request 1 (input, internal pullup)

$\overline{\text{UCS}}$—This pin indicates to the system that a memory access is in progress to the upper memory block. The base address and size of the upper memory block are programmable up to 512 Kbytes. $\overline{\text{UCS}}$ is held High during a bus hold condition.

After power-on reset, $\overline{\text{UCS}}$ is asserted because the processor begins executing at FFFF0h and the default configuration for the $\overline{\text{UCS}}$ chip select is 64 Kbytes from F0000h to FFFFFh. See section 5.5.1.

$\overline{\text{ONCE}}$1—During reset this pin and $\overline{\text{ONCE}}$0 indicate to the microcontroller the mode in which it should operate. $\overline{\text{ONCE}}$0 and $\overline{\text{ONCE}}$1 are sampled on the rising edge of $\overline{\text{RES}}$. If both pins are asserted Low, the microcontroller enters ONCE mode; otherwise, it operates normally. In ONCE mode, all pins assume a high-impedance state and remain in that state until a subsequent reset occurs. To guarantee that the microcontroller does not inadvertently enter ONCE mode, $\overline{\text{ONCE}}$1 has a weak internal pullup resistor that is active only during a reset.

$\overline{UZI}$ **Upper Zero Indicate (output, synchronous)**

This pin lets the designer determine whether an access to the interrupt vector table is in progress by ORing it with bits 15–10 of the address and data bus (AD15–AD10 on the Am186EM and AO15–AO10 on the Am188EM). $\overline{UZI}$ is the logical OR of the inverted A19–A16 bits, and it asserts in the first period of a bus cycle and is held throughout the cycle.

This pin should be allowed to float or should be pulled High at reset. If this pin is Low at the negation of reset, the Am186EM and Am188EM microcontrollers will enter a reserved clock test mode.

V_{CC} **Power Supply (input)**

These pins supply power (+5 V) to the microcontroller.

$\overline{WHB}$ **Write High Byte, Am186EM Microcontroller Only (output, three-state, synchronous)**

This pin and $\overline{WLB}$ indicate to the system which bytes of the data bus (upper, lower, or both) participate in a write cycle. In 80C186 designs, this information is provided by $\overline{BHE}$, the least-significant address bit (AD0), and by $\overline{WR}$. However, by using $\overline{WHB}$ and $\overline{WLB}$, the standard system-interface logic and external address latch that were required are eliminated.

$\overline{WHB}$ is asserted with AD15–AD8. $\overline{WHB}$ is the logical OR of $\overline{BHE}$ and $\overline{WR}$. This pin floats during reset.

$\overline{WLB}$/$\overline{WB}$ **Write Low Byte, Am186EM Microcontroller Only (output, three-state, synchronous)**

Write Byte, Am188EM Microcontroller Only (output, three-state, synchronous)

$\overline{WLB}$—This pin and $\overline{WHB}$ indicate to the system which bytes of the data bus (upper, lower, or both) participate in a write cycle. In 80C186 designs, this information is provided by $\overline{BHE}$, the least-significant address bit (AD0), and by $\overline{WR}$. However, by using $\overline{WHB}$ and $\overline{WLB}$, the standard system interface logic and external address latch that were required are eliminated.

$\overline{WLB}$ is asserted with AD7–AD0. $\overline{WLB}$ is the logical OR of AD0 and $\overline{WR}$. This pin floats during reset.

$\overline{WB}$—On the Am188EM microcontroller, this pin indicates a write to the bus. $\overline{WB}$ uses the same early timing as the nonmultiplexed address bus. $\overline{WB}$ is associated with AD7–AD0. This pin floats during reset. $\overline{WB}$ is the logical OR of $\overline{WHB}$ and $\overline{WLB}$, which are not present on the Am188EM microcontroller.

$\overline{WR}$ **Write Strobe (output, synchronous)**

$\overline{WR}$—This pin indicates to the system that the data on the bus is to be written to a memory or I/O device. $\overline{WR}$ floats during a bus hold or reset condition.

X1 **Crystal Input (input)**

This pin and the X2 pin provide connections for a fundamental mode or third-overtone parallel-resonant crystal used by the internal oscillator circuit. To provide the microcontroller with an external clock source, connect the source to the X1 pin and leave the X2 pin unconnected.

X2 **Crystal Output (output)**

This pin and the X1 pin provide connections for a fundamental mode or third-overtone parallel-resonant crystal used by the internal oscillator circuit. To provide the microcontroller with an external clock source, leave the X2 pin unconnected and connect the source to the X1 pin.

3.1.1 Pins That Are Used by Emulators

The following pins are used by emulators: A19–A0, AO15–AO8, AD7–AD0, ALE, $\overline{\text{BHE}}$/$\overline{\text{ADEN}}$ (on the Am186EM), CLKOUTA, $\overline{\text{RFSH}}$2/$\overline{\text{ADEN}}$ (on the Am188EM), $\overline{\text{RD}}$, $\overline{\text{S}}$2–$\overline{\text{S}}$0, S6/$\overline{\text{CLKDIV}}$2, and $\overline{\text{UZI}}$.

Emulators require that S6/$\overline{\text{CLKDIV}}$2 and $\overline{\text{UZI}}$ be configured in their normal functionality, that is, as S6 and $\overline{\text{UZI}}$.

If $\overline{\text{BHE}}$/$\overline{\text{ADEN}}$ (on the Am186EM) or $\overline{\text{RFSH}}$2/$\overline{\text{ADEN}}$ (on the Am188EM) is held Low during the rising edge of $\overline{\text{RES}}$, S6 and $\overline{\text{UZI}}$ are configured in their normal functionality, instead of as PIOs, at reset.

3.2 BUS OPERATION

The industry-standard 80C186 and 80C188 microcontrollers use a multiplexed address and data (AD) bus. The address is present on the AD bus only during the t_1 clock phase. The Am186EM and Am188EM microcontrollers continue to provide the multiplexed AD bus and, in addition, provide a nonmultiplexed address (A) bus. The A bus provides an address to the system for the complete bus cycle (t_1–t_4).

For systems where power consumption is a concern, it is possible to disable the address from being driven on the AD bus on the Am186EM microcontroller and on the AD and AO buses on the Am188EM microcontroller during the normal address portion of the bus cycle for accesses to UCS and/or LCS address spaces. In this mode, the affected bus is placed in a high impedance state during the address portion of the bus cycle. This feature is enabled through the DA bits in the UMCS and LMCS registers. When address disable is in effect, the number of signals that assert on the bus during all normal bus cycles to the associated address space is reduced, thus decreasing power consumption, reducing processor switching noise, and preventing bus contention with memory devices and peripherals when operating at high clock rates. On the Am188EM microcontroller, the address is driven on A015–A08 during the data portion of the bus cycle, regardless of the setting of the DA bits.

If the $\overline{\text{ADEN}}$ pin is pulled Low during processor reset, the value of the DA bits in the UMCS and LMCS registers is ignored and the address is driven on the AD bus for all accesses, thus preserving the industry-standard 80C186 and 80C188 microcontrollers' multiplexed address bus and providing support for existing emulation tools.

Figure 3-1 on page 3-17 shows the affected signals during a normal read or write operation for an Am186EM microcontroller. The address and data will be multiplexed onto the AD bus.

Figure 3-2 on page 3-17 shows an Am186EM microcontroller bus cycle when address bus disable is in effect. This results in the AD bus operating in a nonmultiplexed data-only mode. The A bus will provide the address during a read or write operation.

Figure 3-3 on page 3-18 shows the affected signals during a normal read or write operation for an Am188EM microcontroller. The multiplexed address/data mode is compatible with 80C188 microcontrollers and might be used to take advantage of existing logic or peripherals.

Figure 3-4 on page 3-18 shows an Am188EM microcontroller bus cycle when address bus disable is in effect. The address and data are not multiplexed. The AD7–AD0 signals will have only data on the bus, while the A bus will have the address during a read or write operation. The AO bus will also have the address during t_2–t_4.

Figure 3-1 Am186EM Microcontroller Address Bus—Normal Read and Write Operation

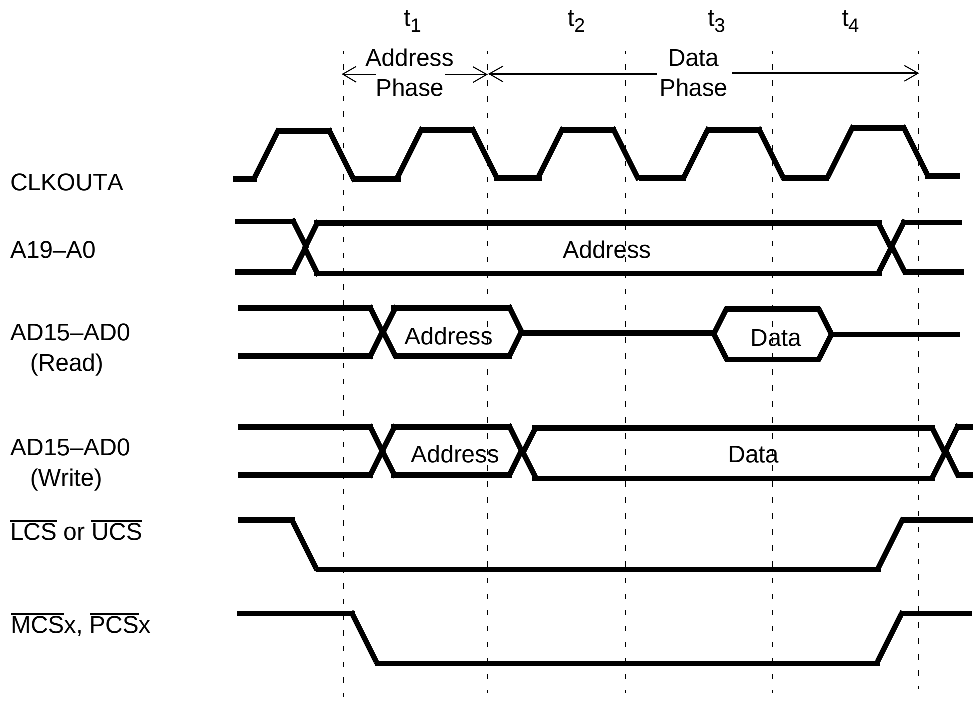


Figure 3-2 Am186EM Microcontroller—Read and Write with Address Bus Disable In Effect

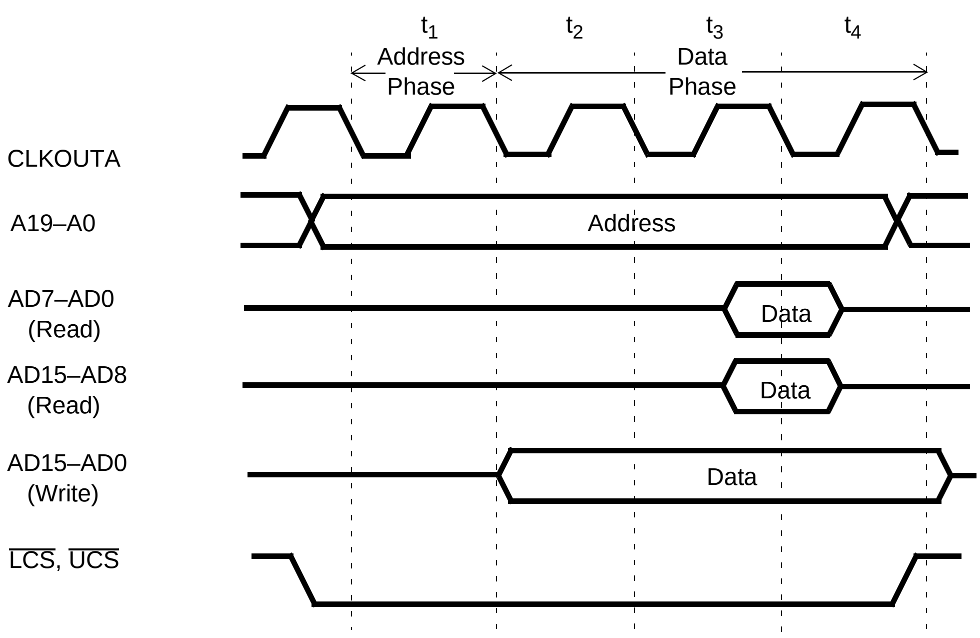


Figure 3-3 Am188EM Microcontroller Address Bus—Normal Read and Write Operation

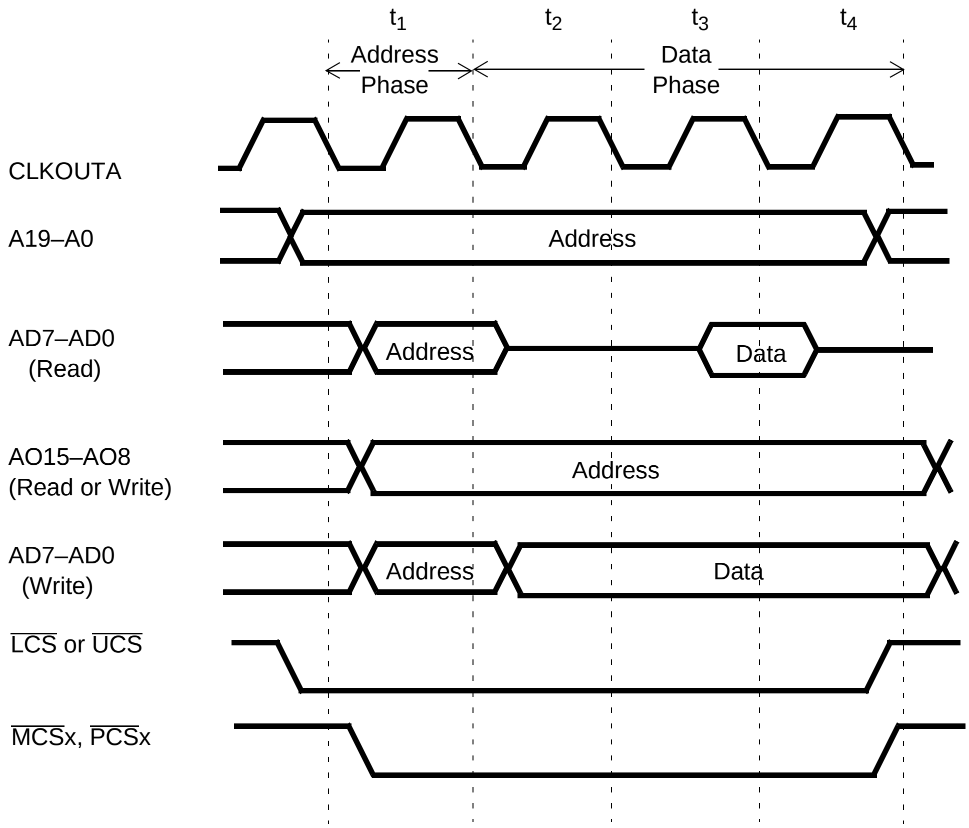
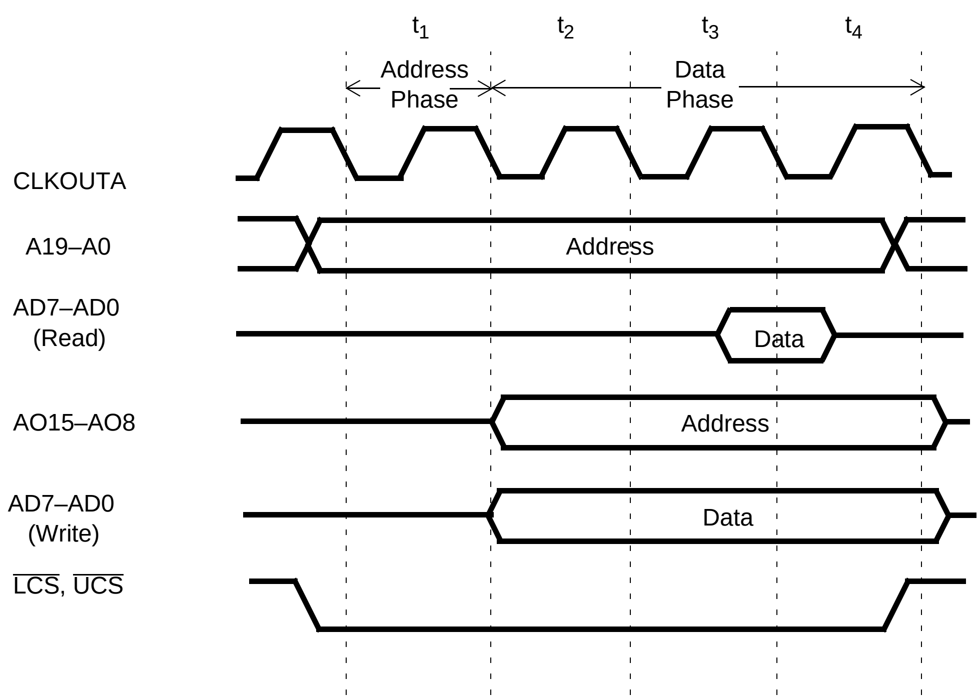


Figure 3-4 Am188EM Microcontroller—Read and Write with Address Bus Disable In Effect



3.3 BUS INTERFACE UNIT

The bus interface unit controls all accesses to external peripherals and memory devices. External accesses include those to memory devices, as well as those to memory-mapped and I/O-mapped peripherals and the peripheral control block. The Am186EM and Am188EM microcontrollers provide an enhanced bus interface unit with the following features:

- A nonmultiplexed address bus
- Separate byte write enables for high and low bytes in the Am186EM microcontroller
- Pseudo-Static RAM (PSRAM) support

The standard 80C186 multiplexed address and data bus requires system-interface logic and an external address latch. On the Am186EM and Am188EM microcontrollers, new byte write enables, PSRAM control logic, and a new nonmultiplexed address bus can reduce design costs by eliminating external logic.

Timing diagrams for the operations described in this chapter appear in the *Am186EM/EMLV and Am188EM/EMLV Microcontrollers Data Sheet*, order# 19168.

3.3.1 Nonmultiplexed Address Bus

The nonmultiplexed address bus (A19–A0) is valid one-half CLKOUTA cycle in advance of the address on the AD bus. When used in conjunction with the modified $\overline{\text{UCS}}$ and $\overline{\text{LCS}}$ outputs and the byte write enable signals, the A19–A0 bus provides a seamless interface to SRAM, PSRAM, and Flash/EPROM memory systems.

3.3.2 Byte Write Enables

The Am186EM microcontroller provides two signals that act as byte write enables—$\overline{\text{WHB}}$ (Write High Byte, AD15–AD8) and $\overline{\text{WLB}}$ (Write Low Byte, AD7–AD0). $\overline{\text{WHB}}$ is the logical OR of $\overline{\text{BHE}}$ and $\overline{\text{WR}}$ ($\overline{\text{WHB}}$ is Low when both $\overline{\text{BHE}}$ and $\overline{\text{WR}}$ are Low). $\overline{\text{WLB}}$ is the logical OR of AD0 and $\overline{\text{WR}}$ ($\overline{\text{WLB}}$ is Low when both AD0 and $\overline{\text{WR}}$ are both Low).

The Am188EM microcontroller provides one signal for byte write enables—$\overline{\text{WB}}$ (Write Byte). $\overline{\text{WB}}$ is the logical OR of $\overline{\text{WHB}}$ and $\overline{\text{WLB}}$, which are not present on the Am188EM microcontroller.

The byte write enables are driven in conjunction with the demultiplexed address bus as required for the write timing requirements of common SRAMs.

3.3.3 Pseudo Static RAM (PSRAM) Support

The Am186EM and Am188EM microcontrollers support the use of PSRAM devices in low memory chip select (LCS) space only. When PSRAM mode is enabled, the timing for the $\overline{\text{LCS}}$ signal is modified by the chip select control unit to provide a $\overline{\text{CS}}$ precharge period during PSRAM accesses. The 40-MHz timing of the Am186EM microcontroller is appropriate to allow 70-ns PSRAM to run with one wait state. PSRAM mode is enabled through a bit in the Low Memory Chip Select (LMCS) register. (See section 5.5.2 on page 5-6.) The PSRAM feature is disabled on CPU reset.

In addition to the $\overline{\text{LCS}}$ timing changes for PSRAM precharge, the PSRAM devices also require periodic refresh of all internal row addresses to retain their data. Although refresh of PSRAM can be accomplished several ways, the Am186EM and Am188EM microcontrollers implement auto refresh only. The microcontroller generates a refresh signal, $\overline{\text{RFSH}}$, to the PSRAM devices when PSRAM mode is enabled. No refresh address is required by the PSRAM when using the auto refresh mechanism. The $\overline{\text{RFSH}}$ signal is multiplexed with the $\overline{\text{MCS3}}$ signal pin. When PSRAM mode is enabled, $\overline{\text{MCS3}}$ is not available for use as a chip select signal.

The refresh control unit must be programmed before accessing PSRAM in LCS space. The refresh counter in the Clock Prescaler (CDRAM) register must be configured with the required refresh interval value. The ending address of LCS space and the ready and wait-state generation in the LMCS register must also be programmed.

The refresh counter reload value in the CDRAM register should not be set to less than 18 (12h) in order to provide time for processor cycles within refresh. In PSRAM mode, the refresh address counter must be set to 0000h to prevent another chip select from asserting. $\overline{\text{LCS}}$ is held High during a refresh cycle. The A19–A0 bus is not used during refresh cycles. The LMCS register must be configured to external Ready ignored (R2=1) with one wait state (R1–R0=01b), and the PSRAM mode enable bit (PSE) must be set to 1. See section 5.5.2 on page 5-6.

3.4 CLOCK AND POWER MANAGEMENT UNIT

The clock and power management unit of the Am186EM and Am188EM microcontrollers includes a phase-locked loop (PLL) and a second programmable system clock output (CLKOUTB).

3.4.1 Phase-Locked Loop (PLL)

In a traditional 80C186/188 design, the crystal frequency is twice that of the desired internal clock. Because of the internal PLL on the Am186EM and Am188EM microcontrollers, the internal clock generated by the microcontroller (CLKOUTA) is the same frequency as the crystal. The PLL takes the crystal inputs (X1 and X2) and generates a 45/55% (worst case) duty cycle intermediate system clock of the same frequency. This feature removes the need for an external 2x oscillator, thereby reducing system cost. The PLL is reset during power-on reset by an on-chip power-on reset (POR) circuit.

3.4.2 Crystal-Driven Clock Source

The internal oscillator circuit of the microcontroller is designed to function with a parallel resonant fundamental or third overtone crystal. Because of the PLL, the crystal frequency is equal to the processor frequency. Replacement of a crystal with an LC or RC equivalent is not recommended.

The X1 and X2 signals are connected to an internal inverting amplifier (oscillator) which provides, along with the external feedback loading, the necessary phase shift (Figure 3-5). In such a positive feedback circuit, the inverting amplifier has an output signal (X2) 180 degrees out of phase of the input signal (X1). The external feedback network provides an additional 180-degree phase shift. In an ideal system, the input to X1 will have 360 or zero degrees of phase shift.

The external feedback network is designed to be as close as possible to ideal. If the feedback network is not providing necessary phase shift, negative feedback will dampen the output of the amplifier and negatively affect the operation of the clock generator. Values for the loading on X1 and X2 must be chosen to provide the necessary phase shift and crystal operation.

3.4.2.1 Selecting a Crystal

When selecting a crystal, the load capacitance should always be specified (C_L). This value can cause variance in the oscillation frequency from the desired specified value (resonance). The load capacitance and the loading of the feedback network have the following relationship:

$$C_L = \frac{(C_1 \cdot C_2)}{(C_1 + C_2)} + C_S$$

where C_S is the stray capacitance of the circuit. Placing the crystal and C_L in series across the inverting amplifier and tuning these values (C_1, C_2) allows the crystal to oscillate at resonance. This relationship is true for both fundamental and third-overtone operation. Finally, there is a relationship between C_1 and C_2. To enhance the oscillation of the inverting amplifier, these values need to be offset with the larger load on the output (X2). Equal values of these loads tend to balance the poles of the inverting amplifier.

The characteristics of the inverting amplifier set limits on the following parameters for crystals:

ESR (Equivalent Series Resistance)....................80 ohm Max

Drive Level ....1 mW Max

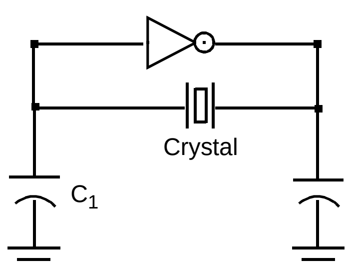
The recommended range of values for C_1 and C_2 are as follows:

C_1..15 pF ± 20%

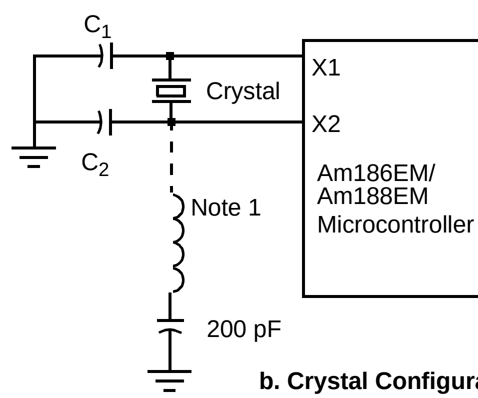
C_2..22 pF ± 20%

The specific values for C_1 and C_2 must be determined by the designer and are dependent on the characteristics of the chosen crystal and board design.

Figure 3-5 Oscillator Configurations



a. Inverting Amplifier Configuration



b. Crystal Configuration

***Note 1**: Use for Third Overtone Mode*

XTAL Frequency	***L1 Value (Max***
20 MHz	*12 µH ±20%*
25 MHz	*8.2 µH ±20%*
33 MHz	*4.7 µH ±20%*
40 MHz	*3.0 µH ±20%*

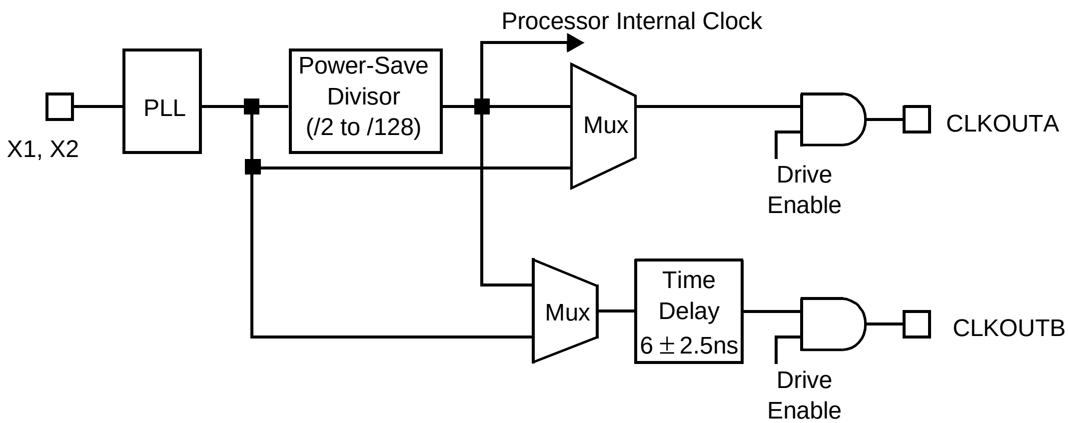
3.4.3 External Source Clock

Alternately, the internal oscillator can be driven from an external clock source. This source should be connected to the input of the inverting amplifier (X1) with the output (X2) not connected.

3.4.4 System Clocks

Figure 3-6 shows the organization of the clocks. The 80C186 microcontroller system clock has been renamed CLKOUTA. CLKOUTB is provided as an additional output.

Figure 3-6 Clock Organization



CLKOUTA and CLKOUTB operate at either the processor frequency or the PLL frequency. The output drivers for both clocks are individually programmable for drive enable or disable.

The second clock output (CLKOUTB) lets one clock run at the PLL frequency and another clock run at the power-save frequency. Individual drive enable bits allow selective enabling of just one or both of these clock outputs.

3.4.5 Power-Save Operation

The power-save mode reduces power consumption and heat dissipation, which can reduce power supply costs and size in all systems and extend battery life in portable systems. In power-save mode, operation of the CPU and internal peripherals continues at a slower clock frequency. When an interrupt occurs, the microcontroller automatically returns to its normal operating frequency on the internal clock’s next rising edge of t_3.

Note: *Power-save operation requires that clock-dependent devices be reprogrammed for clock frequency changes. Software drivers must be aware of clock frequency.*

CHAPTER

4 PERIPHERAL CONTROL BLOCK

4.1 OVERVIEW

The Am186EM and Am188EM microcontroller integrated peripherals are controlled by 16-bit read/write registers. The peripheral registers are contained within an internal 256-byte control block—the peripheral control block. Registers are physically located in the peripheral devices they control, but they are addressed as a single 256-byte block. Figure 4-1 shows a map of the peripheral control block registers.

Code that is intended to execute on the Am188EM microcontroller should perform all writes to the PCB registers as byte writes. These writes will transfer 16 bits of data to the PCB register even if an 8-bit register is named in the instruction. For example, `out dx, al` results in the value of `ax` being written to the port address in `dx`. Reads to the PCB should be done as word reads. Code written in this manner will run correctly on the Am188EM microcontroller and on the Am186EM microcontroller. Unaligned reads and writes to the PCB result in unpredictable behavior on both the Am186EM and Am188EM microcontrollers.

The peripheral control block can be mapped into either memory or I/O space. The base address of the control block must be on an even 256-byte boundary (i.e., the lower eight bits of the base address are 00h). Internal logic recognizes control block addresses and responds to bus cycles. During bus cycles to internal registers, the bus controller signals the operation externally (i.e., the $\overline{\text{RD}}$, $\overline{\text{WR}}$, status, address, and data lines are driven as in a normal bus cycle), but the data bus, SRDY, and ARDY are ignored.

At reset, the Peripheral Control Block Relocation register is set to 20FFh, which maps the control block to start at FF00h in I/O space. An offset map of the 256-byte peripheral control register block is shown in Figure 4-1. See section 4.1.1 on page 4-4 for a complete description of the Peripheral Control Block Relocation (RELREG) register.

Figure 4-1 Peripheral Control Block Register Map

Offset (Hexadecimal)	Register Name	
FE	Peripheral Control Block Relocation Register	Chapter 4
F6	Reset Configuration Register	Chapter 4
F4	Processor Release Level Register	Chapter 4
F0	PDCON Register	Chapter 4
E4	Enable RCU Register	Chapter 6
E2	Clock Prescaler Register	Chapter 6
E0	Memory Partition Register	Chapter 6
DA	DMA 1 Control Register	Chapter 9
D8	DMA 1 Transfer Count Register	Chapter 9
D6	DMA 1 Destination Address High Register	Chapter 9
D4	DMA 1 Destination Address Low Register	Chapter 9
D2	DMA 1 Source Address High Register	Chapter 9
D0	DMA 1 Source Address Low Register	Chapter 9
CA	DMA 0 Control Register	Chapter 9
C8	DMA 0 Transfer Count Register	Chapter 9
C6	DMA 0 Destination Address High Register	Chapter 9
C4	DMA 0 Destination Address Low Register	Chapter 9
C2	DMA 0 Source Address High Register	Chapter 9
C0	DMA 0 Source Address Low Register	Chapter 9
A8	$\overline{PCS}$ and $\overline{MCS}$ Auxiliary Register	Chapter 5
A6	Midrange Memory Chip Select Register	Chapter 5
A4	Peripheral Chip Select Register	Chapter 5
A2	Low Memory Chip Select Register	Chapter 5
A0	Upper Memory Chip Select Register	Chapter 5
88	Serial Port Baud Rate Divisor Register	Chapter 10
86	Serial Port Receive Register	Chapter 10
84	Serial Port Transmit Register	Chapter 10
82	Serial Port Status Register	Chapter 10
80	Serial Port Control Register	Chapter 10

Note*: Gaps in offset addresses indicate reserved registers.*

(shaded box) Changed from 80C186 microcontroller.

Offset (Hexadecimal)	Register Name	
7A	PIO Data 1 Register	Chapter 12
78	PIO Direction 1 Register	
76	PIO Mode 1 Register	
74	PIO Data 0 Register	
72	PIO Direction 0 Register	
70	PIO Mode 0 Register	
66	Timer 2 Mode/Control Register	Chapter 8
62	Timer 2 Maxcount Compare A Register	
60	Timer 2 Count Register	
5E	Timer 1 Mode/Control Register	
5C	Timer 1 Maxcount Compare B Register	
5A	Timer 1 Maxcount Compare A Register	
58	Timer 1 Count Register	
56	Timer 0 Mode/Control Register	
54	Timer 0 Maxcount Compare B Register	
52	Timer 0 Maxcount Compare A Register	
50	Timer 0 Count Register	
44	Serial Port Interrupt Control Register	Chapter 7
42	Watchdog Timer Control Register	
40	INT4 Control Register	
3E	INT3 Control Register	
3C	INT2 Control Register	
3A	INT1 Control Register	
38	INT0 Control Register	
36	DMA 1 Interrupt Control Register	
34	DMA 0 Interrupt Control Register	
32	Timer Interrupt Control Register	
30	Interrupt Status Register	
2E	Interrupt Request Register	
2C	In-service Register	
2A	Priority Mask Register	
28	Interrupt Mask Register	
26	Poll Status Register	
24	Poll Register	
22	End-of-Interrupt Register	
20	Interrupt Vector Register	
18	Synchronous Serial Receive Register	Chapter 11
16	Synchronous Serial Transmit 0 Register	
14	Synchronous Serial Transmit 1 Register	
12	Synchronous Serial Enable Register	
10	Synchronous Serial Status Register	

***Note**: Gaps in offset addresses indicate reserved registers.*

[shaded box] *Changed from 80C186 microcontroller.*

4.1.1 Peripheral Control Block Relocation Register (RELREG, Offset FEh)

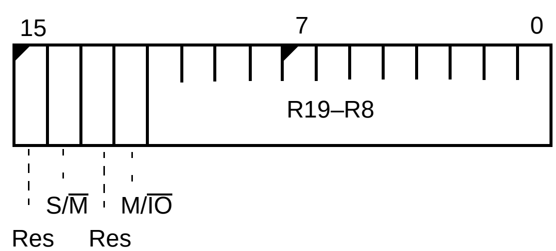
The peripheral control block is mapped into either memory or I/O space by programming the Peripheral Control Block Relocation (RELREG) register (see Figure 4-2). This register is a 16-bit register at offset FEh from the control block base address. The RELREG register provides the upper 12 bits of the base address of the control block. The control block is effectively an internal chip select range.

Other chip selects can overlap the control block only if they are programmed to zero wait states and ignore external ready. If the control register block is mapped into I/O space, the upper four bits of the base address must be programmed as 0000b (since I/O addresses are only 16 bits wide).

In addition to providing relocation information for the control block, the RELREG register contains a bit that places the interrupt controller into either slave mode or master mode.

At reset, the RELREG register is set to 20FFh, which maps the control block to start at FF00h in I/O space. An offset map of the 256-byte peripheral control register block is shown in Figure 4-1.

Figure 4-2 Peripheral Control Block Relocation Register (RELREG, offset FEh)



The value of the RELREG register is 20FFh at reset.

Bit 15: Reserved

Bit 14: Slave/Master (S/$\overline{M}$)—Configures the interrupt controller for slave mode when set to 1 and for master mode when set to 0.

Bit 13: Reserved

Bit 12: Memory/IO Space (M/$\overline{IO}$)—When set to 1, the peripheral control block (PCB) is located in memory space. When set to 0, the PCB is located in I/O space.

Bits 11–0: Relocation Address Bits (R19–R8)—R19–R8 define the upper address bits of the PCB base address. The lower eight bits (R7–R0) default to 00h. R19–R16 are ignored when the PCB is mapped to I/O space.

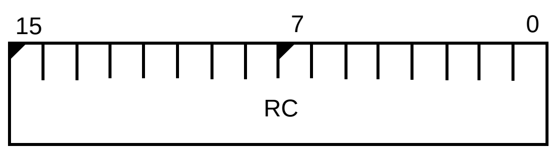
4.1.2 Reset Configuration Register (RESCON, Offset F6h)

The Reset Configuration (RESCON) register (see Figure 4-3) in the peripheral control block latches system-configuration information that is presented to the processor on the address/data bus (AD15–AD0 for the Am186EM or AO15–AO8 and AD7–AD1 for the Am188EM) during the rising edge of reset. The interpretation of this information is system-specific. The processor does not impose any predetermined interpretation, but simply provides a means for communicating this information to software.

When the $\overline{\text{RES}}$ input is asserted Low, the contents of the address/data bus are written into the RESCON register. The system can place configuration information on the address/data bus using weak external pullup or pulldown resistors, or using an external driver that is enabled during reset. The processor does not drive the address/data bus during reset.

For example, the RESCON register could be used to provide the software with the position of a configuration switch in the system. Using weak external pullup and pulldown resistors on the address and data bus, the system could provide the microcontroller with a value corresponding to the position of a jumper during a reset.

Figure 4-3 Reset Configuration Register (RESCON, offset F6h)



On reset, the RESCON register is set to the value found on AD15–AD0.

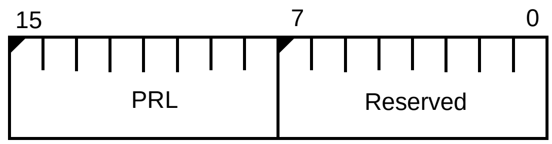
Bits 15–0: Reset Configuration (RC)—There is a one-to-one correspondence between address/data bus signals during the reset and the Reset Configuration register's bits. On the Am186EM microcontroller, AD15 corresponds to bit 15 of the Reset Configuration register, and so on. On the Am188EM microcontroller, AO15 corresponds to register bit 15, and AD7 corresponds to bit 7. Once $\overline{\text{RES}}$ is deasserted, the RESCON register holds its value. This value can be read by software to determine the configuration information.

The contents of the RESCON register are read-only and remain valid until the next processor reset.

4.1.3 Processor Release Level Register (PRL, Offset F4h)

The Processor Release Level (PRL) register (Figure 4-4) is a read-only register that specifies the processor version.

Figure 4-4 Processor Release Level Register (PRL, offset F4h)



The values of the PRL register are listed in Table 4-1.

Bits 15–8: Processor Release Level (PRL)—This field is an 8-bit, read-only identification number that specifies the processor release level. The values of the PRL field for the Am186EM and Am188EM microcontrollers are shown in Table 4-1. Each release level is numbered one higher than the previous level.

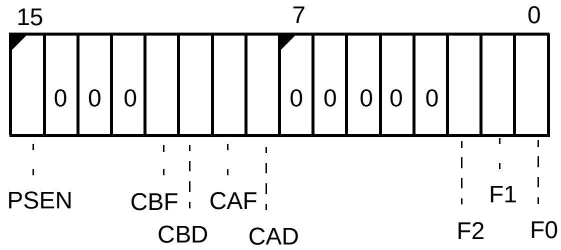
Bits 7–0: Reserved

Table 4-1 Processor Release Level (PRL) Values

PRL Value	Processor Release Level
01h	C
02h	D
03h	E
04h	F

4.1.4 Power-Save Control Register (PDCON, Offset F0h)

Figure 4-5 Power-Save Control Register (PDCON, offset F0h)



The value of the PDCON register is 0000h at reset.

Bit 15: Enable Power-Save Mode (PSEN)—When set to 1, enables Power-Save mode and divides the internal operating clock by the value in F2–F0. PSEN is automatically cleared when an external interrupt, including those generated by on-chip peripheral devices, occurs. The value of the PSEN bit is not restored by the execution of an IRET instruction. Software interrupts (INT instruction) and exceptions do not clear the PSEN bit, and interrupt service routines for these conditions should do so if desired. This bit is 0 after processor reset.

Bits 14–12: Reserved—Read back as 0.

Bit 11: CLKOUTB Output Frequency (CBF)—When set to 1, CLKOUTB follows the crystal input (PLL) frequency. When set to 0, CLKOUTB follows the internal processor frequency (after the clock divisor). Set to 0 on reset.

CLKOUTB can be used as a full-speed clock source in power-save mode.

Bit 10: CLKOUTB Drive Disable (CBD)—When set to 1, CBD three-states the clock output driver for CLKOUTB. When set to 0, CLKOUTB is driven as an output. Set to 0 on reset.

Bit 9: CLKOUTA Output Frequency (CAF)—When set to 1, CLKOUTA follows the crystal input (PLL) frequency. When set to 0, CLKOUTA follows the internal processor frequency (after the clock divisor). Set to 0 on reset.

CLKOUTA can be used as a full-speed clock source in power-save mode.

Bit 8: CLKOUTA Drive Disable (CAD)—When set to 1, CAD three-states the clock output driver for CLKOUTA. When set to 0, CLKOUTA is driven as an output. Set to 0 on reset.

Bits 7–3: Reserved—Read back as 0.

Bits 2–0: Clock Divisor Select (F2–F0)—Controls the division factor when Power-Save mode is enabled. Allowable values are as follows:

F2	F1	F0	Divider Factor
0	0	0	Divide by 1 (2^0)
0	0	1	Divide by 2 (2^1)
0	1	0	Divide by 4 (2^2)
0	1	1	Divide by 8 (2^3)
1	0	0	Divide by 16 (2^4)
1	0	1	Divide by 32 (2^5)
1	1	0	Divide by 64 (2^6)
1	1	1	Divide by 128 (2^7)

4.2 INITIALIZATION AND PROCESSOR RESET

Processor initialization or startup is accomplished by driving the $\overline{\text{RES}}$ input pin Low. $\overline{\text{RES}}$ must be Low during power-up to ensure proper device initialization. $\overline{\text{RES}}$ forces the Am186EM and Am188EM microcontrollers to terminate all execution and local bus activity. No instruction or bus activity occurs as long as $\overline{\text{RES}}$ is active.

After $\overline{\text{RES}}$ is deasserted and an internal processing interval elapses, the microcontroller begins execution with the instruction at physical location FFFF0h. $\overline{\text{RES}}$ also sets some registers to predefined values as shown in Table 4-2.

Table 4-2 Initial Register State After Reset

Register Name	Mnemonic	Value at Reset	Comments
Processor Status Flags	F	F002h	Interrupts disabled
Instruction Pointer	IP	0000h	
Code Segment	CS	FFFFh	Boot address is FFFF0h
Data Segment	DS	0000h	DS = ES = SS = 0000h
Extra Segment	ES	0000h	
Stack Segment	SS	0000h	
Processor Release Level	PRL	XXxxh	PRL XX = Revision (lower half-word is undefined)
Peripheral Control Block Relocation	RELREG	20FFh	Peripheral control block located at FF00h in I/O space and interrupt controller in master mode
Memory Partition	MDRAM	0000h	Refresh base address is 00000h
Enable RCU	EDRAM	0000h	Refresh disabled, counter = 0
Upper Memory Chip Select	UMCS	F03Bh	$\overline{\text{UCS}}$ active for 64K from F0000h to FFFFFh, 3 wait states, external Ready signal required
Low Memory Chip Select	LMCS	Undefined	
Serial Port Control	SPCT	0000h	Serial port interrupts disabled, no loopback, no break, BRKVAL low, no parity, word length = 7, 1 stop bit, transmitter and receiver disabled
PIO Direction 1	PIODIR1	FFFFh	
PIO Mode 1	PIOMODE1	0000h	
PIO Direction 0	PIODIR0	FC0Fh	
PIO Mode 0	PIOMODE0	0000h	
Serial Port Interrupt Control	SPICON	001Fh	Serial port interrupt masked, priority 7
Watchdog Timer Interrupt Control	WDCON	000Fh	Watchdog timer interrupt masked, priority 7
INT4 Control	I4CON	000Fh	Int4 interrupt masked, edge-triggered, priority 7
INT3 Control	I3CON	000Fh	Int3 interrupt masked, edge-triggered, priority 7
INT2 Control	I2CON	000Fh	Int2 interrupt masked, edge-triggered, priority 7
INT1 Control	I1CON	000Fh	Int1 interrupt masked, edge-triggered, priority 7
INT0 Control	I0CON	000Fh	Int0 interrupt masked, edge-triggered, priority 7
DMA1 Interrupt Control	DMA1CON	000Fh	DMA1 interrupts masked, edge-triggered, priority 7
DMA0 Interrupt Control	DMA0CON	000Fh	DMA0 interrupts masked, edge-triggered, priority 7
Timer Interrupt Control	TCUCON	000Fh	Timer interrupts masked, edge-triggered, priority 7
In-Service	INSERV	0000h	No interrupts are in-service
Priority Mask	PRIMSK	0007h	Allow all interrupts based on priority
Interrupt Mask	IMASK	07FDh	All interrupts masked (off)
Synchronous Serial Control	SSC	0000h	SCLK = 1/2 CLKOUTA, no data enabled
Synchronous Serial Status	SSS	0000h	Synchronous serial port not busy, no errors, no transmit or receive completed.
DMA 1 Control	D1CON	FFF9h	
DMA 0 Control	D0CON	FFF9h	

Note:
Registers not listed in this table are undefined at reset.

CHAPTER

5 CHIP SELECT UNIT

5.1 OVERVIEW

The Am186EM and Am188EM microcontrollers contain logic that provides programmable chip select generation for both memories and peripherals. In addition, the logic can be programmed to provide ready or wait-state generation and latched address bits A1 and A2. The chip select lines are active for all memory and I/O cycles in their programmed areas, whether they are generated by the CPU or by the integrated DMA unit.

The Am186EM and Am188EM microcontrollers provide six chip select outputs for use with memory devices and six more for use with peripherals in either memory space or I/O space. The six memory chip selects can be used to address three memory ranges. Each peripheral chip select addresses a 256-byte block offset from a programmable base address (see section 4.1.1 on page 4-4).

The chip selects are programmed through the use of five 16-bit peripheral registers (Table 5-1). The UMCS register, offset A0h, is used to program the Upper Memory Chip Select ($\overline{\text{UCS}}$). The LMCS register, offset A2h, is used to program the Lower Memory Chip Select ($\overline{\text{LCS}}$). The Midrange Memory Chip Selects ($\overline{\text{MCS}}$3–$\overline{\text{MCS}}$0) are programmed through the use of two registers—the Midrange Memory Chip Select (MMCS) register, offset A6h and the PCS and MCS Auxiliary (MPCS) register, offset A8h. In addition to its use in configuring the $\overline{\text{MCS}}$ chip selects, the MPCS register and the PACS register are used to program the Peripheral Chip Selects ($\overline{\text{PCS}}$6–$\overline{\text{PCS}}$5 and $\overline{\text{PCS}}$3–$\overline{\text{PCS}}$0).

Note: *The $\overline{\text{PCS}}$4 chip select is not implemented on the Am186EM and Am188EM microcontrollers.*

Table 5-1 Chip Select Register Summary

Offset	Register Mnemonic	Register Name	Affected Pins	Comments
A0h	UMCS	Upper Memory Chip Select	$\overline{\text{UCS}}$	Ending address is fixed at FFFFFh
A2h	LMCS	Lower Memory Chip Select	$\overline{\text{LCS}}$	Starting address is fixed at 00000h
A4h	PACS	Peripheral Chip Select	$\overline{\text{PCS}}$6–$\overline{\text{PCS}}$5 $\overline{\text{PCS}}$3–$\overline{\text{PCS}}$0	Block size is fixed at 256 bytes
A6h	MMCS	Midrange Chip Select	$\overline{\text{MCS}}$3–$\overline{\text{MCS}}$0	Starting address and block size are programmable
A8h	MPCS	$\overline{\text{PCS}}$ and $\overline{\text{MCS}}$ Auxiliary	$\overline{\text{PCS}}$6–$\overline{\text{PCS}}$5 $\overline{\text{PCS}}$3–$\overline{\text{PCS}}$0 $\overline{\text{MCS}}$3–$\overline{\text{MCS}}$0	Affects both $\overline{\text{PCS}}$ and $\overline{\text{MCS}}$ chip selects

Note: *A read or write will enable a chip select register.*

Except for the $\overline{\text{UCS}}$ chip select, which is active on reset as discussed in section 5.5.1, chip selects are not activated until the associated registers have been accessed. (An access is any read or write operation.) For this reason, the chip select registers should not be read by the processor initialization code until after they have been written with valid data. The $\overline{\text{LCS}}$ chip select is activated when the LMCS register is accessed, the $\overline{\text{MCS}}$ chip selects are activated after both the MMCS and MPCS registers have been accessed, and the $\overline{\text{PCS}}$ chip selects are activated after both the PACS and MPCS registers have been accessed.

5.2 CHIP SELECT TIMING

The timing for the $\overline{\text{UCS}}$ and $\overline{\text{LCS}}$ outputs has been modified from the 80C186 and 80C188 microcontrollers. These outputs now assert in conjunction with the demultiplexed address bus (A19–A0) for normal memory timing. To make these outputs available earlier in the bus cycle, the number of programmable memory size selections has been reduced.

The $\overline{\text{MCS}}$3–$\overline{\text{MCS}}$0 and $\overline{\text{PCS}}$ chip selects assert with the AD bus.

5.3 READY AND WAIT-STATE PROGRAMMING

The Am186EM and Am188EM microcontrollers can be programmed to sense a ready signal for each of the peripheral or memory chip select lines. The ready signal can be either the ARDY or SRDY signal. Each chip select control register (UMCS, LMCS, MMCS, PACS, and MPCS) contains a single-bit field, R2, that determines whether the external ready signal is required or ignored. When R2 is set to 1, external ready is ignored. When R2 is set to 0, external ready is required.

The number of wait states to be inserted for each access to a peripheral or memory region is programmable. Zero wait states to 15 wait states can be inserted for the $\overline{\text{PCS}}$3–$\overline{\text{PCS}}$0 peripheral chip selects. Zero wait states to three wait states can be inserted for all other chip selects.

Each of the chip select control registers, other than the PACS register (UMCS, LMCS, MMCS, and MPCS), contains a two-bit field, R1–R0, whose value determines the number of wait states from none to three to be inserted. A value of 00b in this field specifies no inserted wait states. A value of 11b specifies three inserted wait states.

The $\overline{\text{PCS}}$3–$\overline{\text{PCS}}$0 peripheral chip selects can be programmed for up to 15 wait states. The PACS register uses bits R3 and R1–R0 for the additional wait states.

When external ready is required (R2 is set to 0), internally programmed wait states will always complete before external ready can terminate or extend a bus cycle. For example, if the internal wait states are set to insert two wait states (R1–R0 = 10b), the processor samples the external ready pin during the first wait cycle. If external ready is asserted at that time, the access completes after six cycles (four cycles plus two wait states). If external ready is not asserted during the first wait state, the access is extended until ready is asserted, which is followed by one more wait state followed by t_4.

5.4 CHIP SELECT OVERLAP

Although programming the various chip selects on the Am186EM microcontroller so that multiple chip select signals are asserted for the same physical address is not recommended, it may be unavoidable in some systems. In such systems, the chip selects whose assertions overlap must have the same configuration for ready (external ready required or not required) and the number of wait states to be inserted into the cycle by the processor.

The peripheral control block (PCB) is accessed using internal signals. These internal signals function as chip selects configured with zero wait states and no external ready. Therefore, the PCB can be programmed to addresses that overlap external chip select signals if those external chip selects are programmed to zero wait states with no external ready required.

When overlapping an additional chip select with either the LCS or UCS chip selects, it must be noted that setting the Disable Address (DA) bit in the LMCS or UMCS register will disable the address from being driven on the AD bus for all accesses for which the associated chip select is asserted, including any accesses for which multiple chip selects assert.

The MCS and PCS chip select pins can be configured as either chip selects (normal function) or as PIO inputs or outputs. It should be noted; however, that the ready and wait state generation logic for these chip selects is in effect, regardless of their configurations as chip selects or PIOs. This means that if these chip selects are enabled (by a read or write to the MMCS and MPCS registers for the MCS chip selects, or by a read or write to the PACS and MPCS registers for the PCS chip selects), the ready and wait state programming for these signals must agree with the programming for any other chip selects with which their assertion would overlap if they were configured as chip selects.

Although the PCS4 signal is not available on an external pin, the ready and wait state logic for this signal still exists internal to the part. For this reason, the PCS4 address space must follow the rules for overlapping chip selects. The ready and wait-state logic for PCS6–PCS5 is disabled when these signals are configured as address bits A2–A1.

Failure to configure overlapping chip selects with the same ready and wait state requirements may cause the processor to hang with the appearance of waiting for a ready signal. This behavior may occur even in a system in which ready is always asserted (ARDY or SRDY tied High).

Configuring PCS in I/O space with LCS or any other chip select configured for memory address 0 is not considered overlapping of the chip selects. Overlapping chip selects refers to configurations where more than one chip select asserts for the same physical address.

5.5 CHIP SELECT REGISTERS

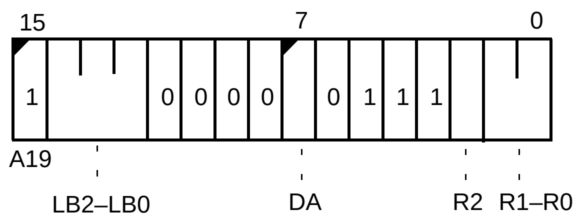
The following sections describe the chip select registers.

5.5.1 Upper Memory Chip Select Register (UMCS, Offset A0h)

The Am186EM and Am188EM microcontrollers provide the $\overline{UCS}$ chip select pin for the top of memory. On reset, the microcontroller begins fetching and executing instructions starting at memory location FFFF0h, so upper memory is usually used as instruction memory. To facilitate this usage, $\overline{UCS}$ defaults to active on reset with a default memory range of 64 Kbytes from F0000h to FFFFFh, external ready required, and three wait states automatically inserted.

The $\overline{UCS}$ memory range always ends at FFFFFh. The lower boundary is programmable. The Upper Memory Chip Select is configured through the UMCS register (Figure 5-1).

Figure 5-1 Upper Memory Chip Select Register (UMCS, offset A0h)



The value of the UMCS register at reset is F03Bh.

Bit 15: Reserved—Set to 1.

Bits 14–12: Lower Boundary (LB2–LB0)—The LB2–LB0 bits define the lower bound of the memory accessed through the $\overline{UCS}$ chip selects. The number of programmable bits has been reduced from eight bits in the 80C186 and 80C188 microcontrollers to three bits in the Am186EM and Am188EM microcontrollers.

The Am186EM and Am188EM microcontrollers provide an additional block size of 512K, which is not available on the 80C186 and 80C188 microcontrollers. Table 5-2 outlines the possible configurations and differences with the 80C186 and 80C188 microcontrollers.

Table 5-2 UMCS Block Size Programming Values

Memory Block Size	Starting Address	LB2–LB0	Comments
64K	F0000h	111b	Default
128K	E0000h	110b	
256K	C0000h	100b	
512K	80000h	000b	Not available on the 80C186 or 80C188 microcontroller

Bits 11–8: Reserved

Bit 7: Disable Address (DA)—The DA bit enables or disables the AD15–AD0 bus during the address phase of a bus cycle when UCS is asserted. If DA is set to 1, AD15–AD0 is not driven during the address phase of a bus cycle when UCS is asserted. If DA is set to 0, AD15–AD0 is driven during the address phase of a bus cycle. Disabling AD15–AD0 reduces power consumption. DA defaults to 0 at power-on reset.

Note: *On the Am188EM microcontroller, the AO15–AO8 address pins are driven during the data phase of the bus cycles, even when the DA bit is set to 1 in either the UMCS or LMCS register.*

If BHE/ADEN (on the Am186EM) or RFSH2/ADEN (on the Am188EM) is held Low on the rising edge of RES, then AD15–AD0 is always driven regardless of the DA setting. This configures AD15–AD0 to be enabled regardless of the setting of DA.

If BHE/ADEN (on the Am186EM) or RFSH2/ADEN (on the Am188EM) is High on the rising edge of RES, then DA in the Upper Memory Chip Select (UMCS) register and DA in the Lower Memory Chip Select (LMCS) register control the AD15–AD0 disabling.

See the descriptions of the BHE/ADEN and RFSH2/ADEN pins in Chapter 3.

Bits 6: Reserved—Set to 0.

Bits 5–3: Reserved—Set to 1.

Bit 2: Ready Mode (R2)—The R2 bit is used to configure the ready mode for the UCS chip select. If R2 is set to 0, external ready is required. If R2 is set to 1, external ready is ignored. In each case, the processor also uses the value of the R1–R0 bits to determine the number of wait states to insert. R2 defaults to 0 at reset.

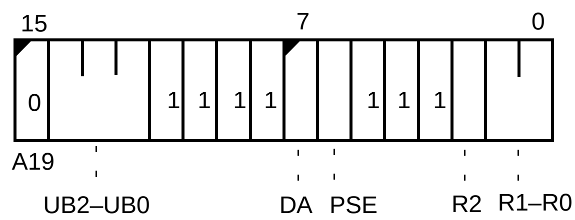
Bits 1–0: Wait-State Value (R1–R0)—The value of R1–R0 determines the number of wait states inserted into an access to the UCS memory area. From zero to three wait states can be inserted (R1–R0 = 00b to 11b). R1–R0 default to 11b at reset.

5.5.2 Low Memory Chip Select Register (LMCS, Offset A2h)

The Am186EM and Am188EM microcontrollers provide the $\overline{LCS}$ chip select pin for the bottom of memory. Since the interrupt vector table is located at 00000h at the bottom of memory, the $\overline{LCS}$ pin has been provided to facilitate this usage. The $\overline{LCS}$ pin is not active on reset, but any read or write access to the LMCS register activates this pin.

The Low Memory Chip Select is configured through the LMCS register (see Figure 5-2).

Figure 5-2 Low Memory Chip Select Register (LMCS, offset A2h)



The value of the LMCS register at reset is undefined.

Bit 15: Reserved—Set to 0.

Bits 14–12: Upper Boundary (UB2–UB0)—The UB2–UB0 bits define the upper bound of the memory accessed through the $\overline{LCS}$ chip select. Because of the timing requirements of the $\overline{LCS}$ output and the nonmultiplexed address bus, the number of programmable memory sizes for the LMCS register is reduced compared to the 80C186 and 80C188 microcontrollers. Consequently, the number of programmable bits has been reduced from eight bits in the 80C186 and 80C188 microcontrollers to three bits in the Am186EM and Am188EM microcontrollers.

The Am186EM and Am188EM microcontrollers have a block size of 512 Kbytes, which is not available on the 80C186 and 80C188 microcontrollers. Table 5-3 outlines the possible configurations and the differences between the 80C186 and 80C188 microcontrollers and the Am186EM and Am188EM microcontrollers.

Table 5-3 LMCS Block Size Programming Values

Memory Block Size	Ending Address	UB2–UB0	Comments
64K	0FFFFh	000b	
128K	1FFFFh	001b	
256K	3FFFFh	011b	
512K	7FFFFh	111b	Not available on the 80C186 and 80C188 microcontrollers

Bits 11–8: Reserved—Set to 1.

Bit 7: Disable Address (DA)—The DA bit enables or disables the AD15–AD0 bus during the address phase of a bus cycle when LCS is asserted. If DA is set to 1, AD15–AD0 is not driven during the address phase of a bus cycle when LCS is asserted. If DA is set to 0, AD15–AD0 is driven during the address phase of a bus cycle. Disabling AD15–AD0 reduces power consumption.

Note: *On the Am188EM microcontroller, the AO15–AO8 address pins are driven during the data phase of the bus cycles, even when the DA bit is set to 1 in either the Upper Memory Chip Select register (UMCS) or the Low Memory Chip Select register (LMCS).*

If BHE/ADEN (on the Am186EM) or RFSH2/ADEN (on the Am188EM) is held Low on the rising edge of RES, then AD15–AD0 is always driven regardless of the DA setting. This configures AD15–AD0 to be enabled regardless of the setting of DA.

If BHE/ADEN (on the Am186EM) or RFSH2/ADEN (on the Am188EM) is High on the rising edge of RES, then the DA bit in the UMCS register and the DA bit in the LMCS register control the AD15–AD0 disabling.

See the descriptions of the BHE/ADEN and RFSH2/ADEN pins in Chapter 3.

Bit 6: PSRAM Mode Enable (PSE)—The PSE bit is used to enable PSRAM support for the LCS chip select memory space. When PSE is set to 1, PSRAM support is enabled. When PSE is set to 0, PSRAM support is disabled. The refresh control unit registers EDRAM, MDRAM, and CDRAM, must be configured for auto refresh before PSRAM support is enabled.

Bits 5–3: Reserved—Set to 1.

Bit 2: Ready Mode (R2)—The R2 bit is used to configure the ready mode for the LCS chip select. If R2 is set to 0, external ready is required. If R2 is set to 1, external ready is ignored. In each case, the processor also uses the value of the R1–R0 bits to determine the number of wait states to insert.

Bits 1–0: Wait-State Value (R1–R0)—The value of R1–R0 determines the number of wait states inserted into an access to the LCS memory area. From zero to three wait states can be inserted (R1–R0 =00b to 11b).

5.5.3 Midrange Memory Chip Select Register (MMCS, Offset A6h)

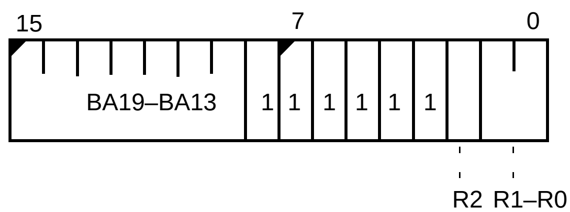
The Am186EM and Am188EM microcontrollers provide four chip select pins, $\overline{\text{MCS}}$3–$\overline{\text{MCS}}$0, for use within a user-locatable memory block. The base address of the memory block can be located anywhere within the 1-Mbyte memory address space, exclusive of the areas associated with the $\overline{\text{UCS}}$ and $\overline{\text{LCS}}$ chip selects (and, if they are mapped to memory, the address range of the Peripheral Chip Selects, $\overline{\text{PCS}}$6–$\overline{\text{PCS}}$5 and $\overline{\text{PCS}}$3–$\overline{\text{PCS}}$0). The $\overline{\text{MCS}}$ address range can overlap the $\overline{\text{PCS}}$ address range if the $\overline{\text{PCS}}$ chip selects are mapped to I/O space.

The Midrange Memory Chip Selects are programmed through two registers. The Midrange Memory Chip Select (MMCS) register (see Figure 5-3) determines the base address and the ready condition and wait states of the memory block accessed through the $\overline{\text{MCS}}$ pins. The $\overline{\text{PCS}}$ and $\overline{\text{MCS}}$ Auxiliary (MPCS) register is used to configure the block size. The $\overline{\text{MCS}}$3–$\overline{\text{MCS}}$0 pins are not active on reset. Both the MMCS and MPCS registers must be accessed with a read or write to activate these chip selects.

Unlike the $\overline{\text{UCS}}$ and $\overline{\text{LCS}}$ chip selects, the $\overline{\text{MCS}}$3–$\overline{\text{MCS}}$0 outputs assert with the multiplexed AD address bus (AD15–AD0 or AO15–AO8 and AD7–AD0) rather than the earlier timing of the A19–A0 bus. The A19–A0 bus can still be used for address selection, but the timing is delayed for a half cycle later than that for $\overline{\text{UCS}}$ and $\overline{\text{LCS}}$.

The Midrange Memory Chip Selects are configured by the MMCS register (Figure 5-3).

Figure 5-3 Midrange Memory Chip Select Register (MMCS, offset A6h)



The value of the MMCS register at reset is undefined.

Bits 15–9: Base Address (BA19–BA13)—The base address of the memory block that is addressed by the $\overline{\text{MCS}}$ chip select pins is determined by the value of BA19–BA13. These bits correspond to bits A19–A13 of the 20-bit memory address. Bits A12–A0 of the base address are always 0.

The base address can be set to any integer multiple of the size of the memory block size selected in the MPCS register. For example, if the midrange block is 32 Kbytes, the block could be located at 10000h or 18000h but not at 14000h.

The base address of the midrange chip selects can be set to 00000h only if the $\overline{\text{LCS}}$ chip select is not active. This is due to the fact that the $\overline{\text{LCS}}$ base address is defined to be address 00000h and chip select address ranges are not allowed to overlap. Because of the additional restriction that the base address must be a multiple of the block size, a 512K MMCS block size can only be used when located at address 00000h, and the $\overline{\text{LCS}}$ chip selects must not be active in this case. Use of the $\overline{\text{MCS}}$ chip selects to access low memory allows the timing of these accesses to follow the AD address bus rather than the A address bus. Locating a 512K MMCS block at 80000h always conflicts with the range of the $\overline{\text{UCS}}$ chip select and is not allowed.

Bits 8–3: Reserved—Set to 1.

Bit 2: Ready Mode (R2)—The R2 bit is used to configure the ready mode for the $\overline{MCS}$ chip selects. If R2 is set to 0, external ready is required. If R2 is set to 1, external ready is ignored. In each case, the processor also uses the value of the R1–R0 bits to determine the number of wait states to insert.

Bits 1–0: Wait-State Value (R1–R0)—The value of R1–R0 determines the number of wait states inserted into an access to the $\overline{MCS}$ memory area. From zero to three wait states can be inserted (R1–R0 = 00b to 11b).

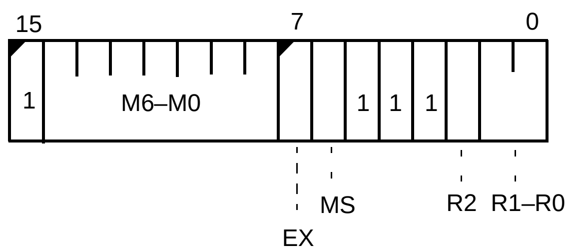
5.5.4 PCS and MCS Auxiliary Register (MPCS, Offset A8h)

The PCS and MCS Auxiliary (MPCS) register (see Figure 5-4) differs from the other chip select control registers in that it contains fields that pertain to more than one type of chip select. The MPCS register fields provide program information for MCS3–MCS0 as well as PCS6–PCS5 and PCS3–PCS0.

In addition to its function as a chip select control register, the MPCS register contains a field that configures the PCS6–PCS5 pins as either chip selects or as alternate sources for the A2 and A1 address bits. When programmed to provide address bits A1 and A2, PCS6–PCS5 cannot be used as peripheral chip selects. These outputs can be used to provide latched address bits for A2 and A1.

On reset, PCS6–PCS5 are not active. If PCS6–PCS5 are configured as address pins, an access to the MPCS register causes the pins to activate. No corresponding access to the PACS register is required to activate the PCS6–PCS5 pins as addresses.

Figure 5-4 PCS and MCS Auxiliary Register (MPCS, offset A8h)



The value of the MPCS register at reset is undefined.

Bit 15: Reserved—Set to 1.

Bits 14–8: MCS Block Size (M6–M0)—This field determines the total block size for the MCS3–MCS0 chip selects. Each individual chip select is active for one quarter of the total block size. The size of the memory block defined is shown in Table 5-4.

Only one of the M6–M0 bits can be set at any time. If more than one of the M6–M0 bits is set, unpredictable operation of the MCS lines occurs.

Table 5-4 MCS Block Size Programming

Total Block Size	Individual Select Size	M6–M0
8K	2K	0000001b
16K	4K	0000010b
32K	8K	0000100b
64K	16K	0001000b
128K	32K	0010000b
256K	64K	0100000b
512K	128K	1000000b

Bit 7: Pin Selector (EX)—This bit determines whether the $\overline{PCS}$6–$\overline{PCS}$5 pins are configured as chip selects or as alternate outputs for A2–A1. When this bit is set to 1, $\overline{PCS}$6–$\overline{PCS}$5 are configured as peripheral chip select pins. When EX is set to 0, $\overline{PCS}$5 becomes address bit A1 and $\overline{PCS}$6 becomes address bit A2.

Bit 6: Memory/ I/O Space Selector (MS)—This bit determines whether the $\overline{PCS}$ pins are active during memory bus cycles or I/O bus cycles. When MS is set to 1, the $\overline{PCS}$ outputs are active for memory bus cycles. When MS is set to 0, the $\overline{PCS}$ outputs are active for I/O bus cycles.

Bits 5–3: Reserved—Set to 1.

Bit 2: Ready Mode (R2)—This bit applies only to the $\overline{PCS}$6–$\overline{PCS}$5 chip selects. If R2 is set to 0, external ready is required. If R2 is set to 1, external ready is ignored. In each case, the processor also uses the value of the R1–R0 bits to determine the number of wait states to insert.

Bits 1–0: Wait-State Value (R1–R0)—These bits apply only to the $\overline{PCS}$6–$\overline{PCS}$5 chip selects. The value of R1–R0 determines the number of wait states inserted into an access to the $\overline{PCS}$ memory or I/O area. From zero to three wait states can be inserted (R1–R0 = 00b to 11b).

5.5.5 Peripheral Chip Select Register (PACS, Offset A4h)

Unlike the $\overline{UCS}$ and $\overline{LCS}$ chip selects, the $\overline{PCS}$ outputs assert with the same timing as the multiplexed AD address bus. Also, each peripheral chip select asserts over a 256-byte address range, which is twice the address range covered by peripheral chip selects in the 80C186 and 80C188 microcontrollers.

The Am186EM and Am188EM microcontrollers provide six chip selects, $\overline{PCS}$6–$\overline{PCS}$5 and $\overline{PCS}$3–$\overline{PCS}$0, for use within a user-locatable memory or I/O block. ($\overline{PCS}$4 is not implemented on the Am186EM and Am 188EM microcontrollers.) The base address of the memory block can be located anywhere within the 1-Mbyte memory address space, exclusive of the areas associated with the $\overline{UCS}$, $\overline{LCS}$, and $\overline{MCS}$ chip selects, or they can be configured to access the 64-Kbyte I/O space.

The Peripheral Chip Selects are programmed through two registers—the Peripheral Chip Select (PACS) register and the $\overline{PCS}$ and $\overline{MCS}$ Auxiliary (MPCS) register. The Peripheral Chip Select (PACS) register (Figure 5-5) determines the base address, the ready condition, and the wait states for the $\overline{PCS}$3–$\overline{PCS}$0 outputs.

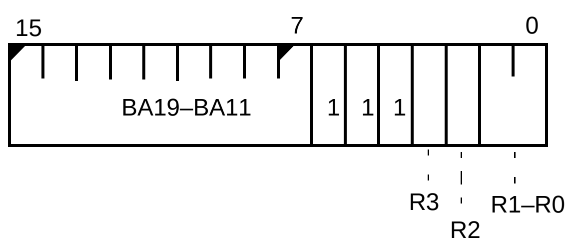
The $\overline{PCS}$ and $\overline{MCS}$ Auxiliary (MPCS) register (see Figure 5-4) contains bits that configure the $\overline{PCS}$6–$\overline{PCS}$5 pins as either chip selects or address pins A1 and A2. When the $\overline{PCS}$6–$\overline{PCS}$5 pins are chip selects, the MPCS register also determines whether $\overline{PCS}$ chip selects are active during memory or I/O bus cycles and specifies the ready and wait states for the $\overline{PCS}$6–$\overline{PCS}$5 outputs.

The $\overline{PCS}$ pins are not active on reset. Both the PACS and MPCS registers must be accessed with a read or write to activate the PCS pins as chip selects.

$\overline{PCS}$6–$\overline{PCS}$5 can be configured and activated as address pins by writing only the MPCS register. No corresponding access to the PACS register is required in this case.

$\overline{PCS}$3–$\overline{PCS}$0 can be configured for zero wait states to 15 wait states. $\overline{PCS}$6–$\overline{PCS}$5 can be configured for zero wait states to three wait states.

Figure 5-5 Peripheral Chip Select Register (PACS, offset A4h)



The value of the PACS register at reset is undefined.

Bits 15–7: Base Address (BA19–BA11)—The base address of the peripheral chip select block is defined by BA19–BA11 of the PACS register. BA19–BA11 correspond to bits 19–11 of the 20-bit programmable base address of the peripheral chip select block. Bit 6 of the PACS register corresponds to bit 10 of the base address in the 80C186 and 80C188 microcontrollers, and is not implemented. Thus, code previously written for the 80C186 microcontroller in which bit 6 was set with a meaningful value would not produce the address expected on the Am186EM.

When the $\overline{PCS}$ chip selects are mapped to I/O space, BA19–16 must be programmed to 0000b because the I/O address bus is only 16-bits wide.

Table 5-5 $\overline{\text{PCS}}$ Address Ranges

$\overline{\text{PCS}}$ Line	Range	
	Low	High
$\overline{\text{PCS}}$0	Base Address	Base Address+255
$\overline{\text{PCS}}$1	Base Address+256	Base Address+511
$\overline{\text{PCS}}$2	Base Address+512	Base Address+767
$\overline{\text{PCS}}$3	Base Address+768	Base Address+1023
Reserved	N/A	N/A
$\overline{\text{PCS}}$5	Base Address+1280	Base Address+1535
$\overline{\text{PCS}}$6	Base Address+1536	Base Address+1791

Bits 6–4: Reserved—Set to 1.

Bit 3: Wait-State Value (R3)—If this bit is set to 0, the number of wait states from zero to three is encoded in the R1–R0 bits. In this case, R1–R0 encodes from zero (00b) to three (11b) wait states.

When R3 is set to 1, the four possible values of R1–R0 encode four additional wait-state values as follows: 00b = 5 wait states, 01b = 7 wait states, 10b = 9 wait states, and 11b = 15 wait states. Table 5-6 shows the wait-state encoding.

Table 5-6 $\overline{\text{PCS}}$3–$\overline{\text{PCS}}$0 Wait-State Encoding

R3	R1	R0	Wait States
0	0	0	0
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	5
1	0	1	7
1	1	0	9
1	1	1	15

Bit 2: Ready Mode (R2)—The R2 bit is used to configure the ready mode for the $\overline{\text{PCS}}$3–$\overline{\text{PCS}}$0 chip selects. If R2 is set to 0, external ready is required. External ready is ignored when R2 is set to 1. In each case, the processor also uses the value of the R3 and R1–R0 bits to determine the number of wait states to insert. The ready mode for $\overline{\text{PCS}}$6–$\overline{\text{PCS}}$5 is configured through the MPCS register.

Bits 1–0: Wait-State Value (R1–R0)—The value of R3 and R1–R0 determines the number of wait states inserted into a $\overline{\text{PCS}}$3–$\overline{\text{PCS}}$0 access. Up to 15 wait states can be inserted.

See the discussion of bit 3 (R3) for the wait-state encoding of R1–R0.

From zero to three wait states for the $\overline{\text{PCS}}$6–$\overline{\text{PCS}}$5 outputs are programmed through the R1–R0 bits in the MPCS register.

CHAPTER

6 REFRESH CONTROL UNIT

6.1 OVERVIEW

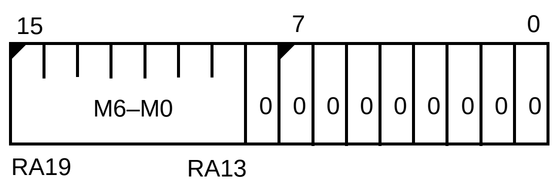
The Refresh Control Unit (RCU) automatically generates refresh bus cycles. After a programmable period of time, the RCU generates a memory read request to the bus interface unit. The RCU is fixed to three wait states for the PSRAM auto refresh mode.

The Refresh Control Unit operates off the processor internal clock. If the power-save mode is in effect, the Refresh Control Unit must be reprogrammed to reflect the new clock rate.

If the HLDA pin is active when a refresh request is generated (indicating a bus hold condition), then the microcontroller deactivates the HLDA pin in order to perform a refresh cycle. The circuit external bus master must remove the HOLD signal for at least one clock to allow the refresh cycle to execute.

6.1.1 Memory Partition Register (MDRAM, Offset E0h)

Figure 6-1 Memory Partition Register (MDRAM, offset E0h)



The MDRAM register is set to 0000h on reset.

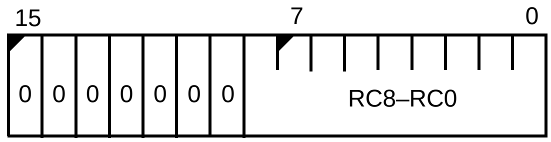
Bits 15–9: Refresh Base (M6–M0)—Upper bits corresponding to address bits A19–A13 of the 20-bit memory refresh address. Since these bits are available only on the AD bus, the AD bit must not be set in the LMCS register if the refresh control unit is used. When using PSRAM mode, M6–M0 must be programmed to 0000000b.

These bits are cleared to 0 at reset.

Bits 8–0: Reserved—Read back as 0.

6.1.2 Clock Prescaler Register (CDRAM, Offset E2h)

Figure 6-2 Clock Prescaler Register (CDRAM, offset E2h)



The CDRAM register is undefined on reset.

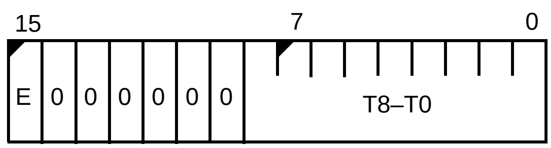
Bits 15–9: Reserved—Read back as 0.

Bits 8–0: Refresh Counter Reload Value (RC8–RC0)—Contains the value of the desired clock count interval between refresh cycles. The counter value should not be set to less than 18 (12h), otherwise there would never be sufficient bus cycles available for the processor to execute code.

In power-save mode, the refresh counter value must be adjusted to take into account the reduced processor clock rate.

6.1.3 Enable RCU Register (EDRAM, Offset E4h)

Figure 6-3 Enable RCU Register (EDRAM, offset E4h)



The EDRAM register is set to 0000h on reset.

Bit 15: Enable RCU (E)—Enables the refresh counter unit when set to 1. Clearing the E bit at any time clears the refresh counter and stops refresh requests, but it does not reset the refresh address. Set to 0 on reset.

Bits 14–9: Reserved—Read back as 0.

Bits 8–0: Refresh Count (T8–T0)—This read-only field contains the present value of the down counter which triggers refresh requests.

CHAPTER

7 INTERRUPT CONTROL UNIT

7.1 OVERVIEW

The Am186EM and Am188EM microcontrollers can receive interrupt requests from a variety of sources, both internal and external. The internal interrupt controller arranges these requests by priority and presents them one at a time to the CPU.

There are six external interrupt sources on the Am186EM and Am188EM microcontrollers—five maskable interrupt pins (INT4–INT0) and the non-maskable interrupt (NMI) pin. There are six internal interrupt sources that are not connected to external pins—three timers, two DMA channels, and the asynchronous serial port.

The Am186EM and Am188EM microcontrollers provide three interrupts that are not present on the 80C186 and 80C188 microcontrollers:

- INT4, an additional external interrupt pin that operates like the INT3–INT0 pins
- An internal watchdog timer interrupt
- An internal interrupt from the serial port

The INT4–INT0 interrupt request pins can be used as direct interrupt requests. If more inputs are needed, INT3–INT0 can also be cascaded with an 82C59A-compatible external interrupt control device. An external interrupt controller can be used as the system master by programming the internal interrupt controller to operate in slave mode. In all cases, nesting can be enabled that allows high priority interrupts to interrupt lower-priority interrupt service routines.

7.1.1 Definitions of Interrupt Terms

The following definitions cover some of the terminology that is used in describing the functionality of the interrupt controller. Table 7-1 contains information regarding the reserved interrupts.

7.1.1.1 Interrupt Type

An 8-bit interrupt type identifies each of the 256 possible interrupts.

Software exceptions, internal peripherals, and non-cascaded external interrupts supply the interrupt type through the internal interrupt controller.

Cascaded external interrupts and slave-mode external interrupts get the interrupt type from the external interrupt controller by means of interrupt acknowledge cycles on the bus.

7.1.1.2 Interrupt Vector Table

The interrupt vector table is a memory area of 1 Kbyte beginning at address 00000h that holds up to 256 four-byte address pointers containing the address for the interrupt service routine for each possible interrupt type. For each interrupt, an 8-bit interrupt type identifies the appropriate interrupt vector table entry.

Interrupts 00h to 1Fh are reserved. See Table 7-1.

The processor calculates the index to the interrupt vector table by shifting the interrupt type left 2 bits (multiplying by 4).

7.1.1.3 Maskable and Non-Maskable Interrupts

Interrupt types 08h through 1Fh are maskable. Of these, only 08h through 14h are actually in use (see Table 7-1). The maskable interrupts are enabled and disabled by the interrupt enable flag (IF) in the processor status flags, but the INT command can execute any interrupt regardless of the setting of IF.

Interrupt types 00h through 07h and all software interrupts (the INT instruction) are non-maskable. The non-maskable interrupts are not affected by the setting of the IF flag.

The Am186EM and Am188EM microcontrollers provide two methods for masking and unmasking the maskable interrupt sources. Each interrupt source has an interrupt control register that contains a mask bit specific to that interrupt. In addition, the Interrupt Mask register is provided as a single source to access all of the mask bits.

If the Interrupt Mask register is written while interrupts are enabled, it is possible that an interrupt could occur while the register is in an undefined state. This can cause interrupts to be accepted even though they were masked both before and after the write to the Interrupt Mask register. Therefore, the Interrupt Mask register should only be written when interrupts are disabled. Mask bits in the individual interrupt control registers can be written while interrupts are enabled, and there will be no erroneous interrupt operation.

7.1.1.4 Interrupt Enable Flag (IF)

The interrupt enable flag (IF) is part of the processor status flags (see section 2.1.1 on page 2-2). If IF is set to 1, maskable interrupts are enabled and can cause processor interrupts. (Individual maskable interrupts can still be disabled by means of the mask bit in each control register.)

If IF is set to 0, all maskable interrupts are disabled.

The IF flag does not affect the NMI or software exception interrupts (interrupt types 00h to 07h), and it does not affect the execution of any interrupt through the INT instruction.

7.1.1.5 Interrupt Mask Bit

Each of the interrupt control registers for the maskable interrupts contains a mask bit (MSK). If MSK is set to 1 for a particular interrupt, that interrupt is disabled regardless of the IF setting.

7.1.1.6 Interrupt Priority

The column titled *Overall Priority* in Table 7-1 shows the fundamental priority breakdown for the interrupts at power-on reset. The non-maskable interrupts 00h through 07h are always prioritized ahead of the maskable interrupts.

The maskable interrupts can be reprioritized by reconfiguring the PR2–PR0 bits in the interrupt control registers. The PR2–PR0 bits in all the maskable interrupts are set to priority level 7 at power-on reset.

7.1.1.7 Software Interrupts

Software interrupts can be initiated by the INT instruction. Any of the 256 possible interrupts can be initiated by the INT instruction. INT 21h causes an interrupt to the vector located at 00084h in the interrupt vector table. INT FFh causes an interrupt to the vector located at 003FCh in the interrupt vector table. Software interrupts are not maskable and are not affected by the setting of the IF flag.

7.1.1.8 Software Exceptions

A software exception interrupt occurs when an instruction causes an interrupt due to some condition in the processor. Interrupt types 00h, 01h, 03h, 04h, 05h, 06h, and 07h are software exception interrupts. Software exceptions are not maskable and are not affected by the setting of the IF flag.

Table 7-1 Am186EM and Am188EM Microcontroller Interrupt Types

Interrupt Name	Interrupt Type	Vector Table Address	EOI Type	Overall Priority	Related Instructions	Notes
Divide Error Exception	00h	00h	N/A	1	DIV, IDIV	1
Trace Interrupt	01h	04h	N/A	1A	All	2
Non-Maskable Interrupt (NMI)	02h	08h	N/A	1B		
Breakpoint Interrupt	03h	0Ch	N/A	1	INT 3	1
INTO Detected Overflow Exception	04h	10h	N/A	1	INTO	1
Array Bounds Exception	05h	14h	N/A	1	BOUND	1
Unused Opcode Exception	06h	18h	N/A	1	Undefined Opcodes	1
ESC Opcode Exception	07h	1Ch	N/A	1	ESC Opcodes	1, 3
Timer 0 Interrupt	08h	20h	08	2A		4, 5
Timer 1 Interrupt	12h	48h	08	2B		4, 5
Timer 2 Interrupt	13h	4Ch	08	2C		4, 5
Reserved for AMD Use	09h					
DMA 0 Interrupt	0Ah	28h	0A	3		5
DMA 1 Interrupt	0Bh	2Ch	0B	4		5
INT0 Interrupt	0Ch	30h	0C	5		
INT1 Interrupt	0Dh	34h	0D	6		
INT2 Interrupt	0Eh	38h	0E	7		
INT3 Interrupt	0Fh	3Ch	0F	8		
INT4 Interrupt	10h	40h	10	9		6
Watchdog Timer Interrupt	11h	44h	11	9		6
Asynchronous Serial Port Interrupt	14h	50h	14	9		6
Reserved for AMD Use	15h–1Fh					

Notes:

1. *Interrupts generated as a result of an instruction execution.*
2. *Trace is performed in the same manner as 80C186 and 80C188.*
3. *An ESC opcode causes a trap. This is part of the 80C186 and 80C188 co-processor interface, which is not supported on the Am186EM.*
4. *All three timers constitute one source of request to the interrupt controller. As such, they share the same priority level with respect to other interrupt sources. However, the timers have a defined priority order among themselves (2A>2B>2C).*
5. *The interrupt types of these sources are programmable in slave mode.*
6. *Not available in slave mode.*

7.1.2 Interrupt Conditions and Sequence

Interrupts are generally serviced as follows.

7.1.2.1 Non-Maskable Interrupts

Non-maskable interrupts—the trace interrupt, the NMI interrupt, and software interrupts [both user-defined (INT) and software exceptions]—are serviced regardless of the setting of the interrupt enable flag (IF) in the processor status flags.

7.1.2.2 Maskable Hardware Interrupts

In order for maskable hardware interrupt requests to be serviced, the IF flag must be set by the STI instruction, and the mask bit associated with each interrupt must be reset.

7.1.2.3 The Interrupt Request

When an interrupt is requested, the internal interrupt controller verifies that the interrupt is enabled and that there are no higher priority interrupt requests being serviced or pending. If the interrupt request is granted, the interrupt controller uses the interrupt type (see Table 7-1) to access a vector from the interrupt vector table.

Each interrupt type has a four-byte vector available in the interrupt vector table. The interrupt vector table is located in the 1024 bytes from 00000h to 003FFh. Each four-byte vector consists of a 16-bit offset (IP) value and a 16-bit segment (CS) value. The 8-bit interrupt type is shifted left 2 bit positions (multiplied by 4) to generate the index into the interrupt vector table.

7.1.2.4 Interrupt Servicing

A valid interrupt transfers execution to a new program location based on the vector in the interrupt vector table. The next instruction address (CS:IP) and the processor status flags are pushed onto the stack.

The interrupt enable flag (IF) is cleared after the processor status flags are pushed on the stack, disabling maskable interrupts during the interrupt service routine (ISR).

The segment:offset values from the interrupt vector table are loaded into the code segment (CS) and the instruction pointer (IP), and execution of the ISR begins.

7.1.2.5 Returning from the Interrupt

The interrupt return (IRET) instruction pops the processor status flags and the return address off the stack. Program execution resumes at the point where the interrupt occurred.

The interrupt enable flag (IF) is restored by the IRET instruction along with the rest of the processor status flags. If the IF flag was set before the interrupt was serviced, interrupts are re-enabled when the IRET is executed. If there are valid interrupts pending when the IRET is executed, the instruction at the return address is not executed. Instead, the new interrupt is serviced immediately.

If an ISR intends to permanently modify the value of any of the saved flags, it must modify the copy of the Processor Status Flags register that was pushed onto the stack.

7.1.3 Interrupt Priority

Table 7-1 shows the predefined types and overall priority structure for the Am186EM and Am188EM microcontrollers. Non-maskable interrupts (interrupt types 0–7) are always higher priority than maskable interrupts. Maskable interrupts have a programmable priority that can override the default priorities relative to one another.

The levels of interrupt priority are as follows:

- Interrupt priority for non-maskable interrupts and software interrupts
- Interrupt priority for maskable hardware interrupts

7.1.3.1 Non-Maskable Interrupts and Software Interrupt Priority

The non-maskable interrupts from 00h to 07h and software interrupts (INT instruction) always take priority over the maskable hardware interrupts. Within the non-maskable and software interrupts, the trace interrupt has the highest priority, followed by the NMI interrupt, followed by the remaining non-maskable and software interrupts.

After the trace interrupt and the NMI interrupt, the remaining software exceptions are mutually exclusive and can only occur one at a time, so there is no further priority breakdown.

7.1.3.2 Maskable Hardware Interrupt Priority

Beginning with interrupt type 8 (the Timer 0 interrupt), the maskable hardware interrupts have both an overall priority (see Table 7-1) and a programmable priority. The programmable priority is the primary priority for maskable hardware interrupts. The overall priority is the secondary priority for maskable hardware interrupts.

Since all maskable interrupts are set to a programmable priority of seven on reset, the overall priority of the interrupts determines the priority in which each interrupt is granted by the interrupt controller until programmable priorities are changed by reconfiguring the control registers.

The overall priority levels shown in Table 7-1 are not the same as the programmable priority level that is associated with each maskable hardware interrupt. Each of the maskable hardware interrupts has a programmable priority from zero to seven, with zero being the highest priority (see Table 7-3 on page 7-14).

For example, if the INT4–INT0 interrupts are all changed to programmable priority six and no other programmable priorities are changed from the reset value of seven, then the INT4–INT0 interrupts take precedence over all other maskable interrupts. (Within INT4–INT0, INT0 takes precedence over INT1, and INT1 takes precedence over INT2, etc., because of the underlying hierarchy of the overall priority.)

7.1.4 Software Exceptions, Traps, and NMI

The following predefined interrupts cannot be masked by programming.

7.1.4.1 Divide Error Exception (Interrupt Type 00h)

Generated when a DIV or IDIV instruction quotient cannot be expressed in the number of destination bits.

7.1.4.2 Trace Interrupt (Interrupt Type 01h)

If the trace flag (TF) in the Processor Status flags register is set, the trace interrupt is generated after most instructions. This interrupt allows programs to execute in single-step mode. The interrupt is not generated after prefix instructions like REP, instructions that modify segment registers like POP DS, or the WAIT instruction.

Taking the trace interrupt clears the TF bit after the processor status flags are pushed onto the stack. The IRET instruction at the end of the single step interrupt service routine restores the processor status flags (and the TF bit) and transfers control to the next instruction to be traced.

Trace mode is initiated by pushing the processor status flags onto the stack, setting the TF flag on the stack, and then popping the flags.

7.1.4.3 Non-Maskable Interrupt—NMI (Interrupt Type 02h)

The NMI pin provides an external interrupt source that is serviced regardless of the state of the IF (interrupt enable flag) bit. No external interrupt acknowledge sequence is performed for an NMI interrupt (see section 7.1.5). A typical use of NMI is to activate a power failure routine.

7.1.4.4 Breakpoint Interrupt (Interrupt Type 03h)

An interrupt caused by the 1-byte version of the INT instruction (INT3).

7.1.4.5 INTO Detected Overflow Exception (Interrupt Type 04h)

Generated by an INTO instruction if the OF bit is set in the Processor Status Flags (FLAGS) register.

7.1.4.6 Array BOUNDS Exception (Interrupt Type 05h)

Generated by a BOUND instruction if the array index is outside the array bounds. The array bounds are located in memory at a location indicated by one of the instruction operands. The other operand indicates the value of the index to be checked.

7.1.4.7 Unused Opcode Exception (Interrupt Type 06h)

Generated if execution is attempted on undefined opcodes.

7.1.4.8 ESC Opcode Exception (Interrupt Type 07h)

Generated if execution of ESC opcodes (D8h–DFh) is attempted. The microcontrollers do not check the escape opcode trap bit. The return address of this exception points to the ESC instruction that caused the exception. If a segment override prefix preceded the ESC instruction, the return address points to the segment override prefix.

Note: *All numeric coprocessor opcodes cause a trap. The Am186EM and Am188EM microcontrollers do not support the numeric coprocessor interface.*

7.1.5 Interrupt Acknowledge

Interrupts can be acknowledged in two different ways—the internal interrupt controller can provide the interrupt type or an external interrupt controller can provide the interrupt type. The processor requires the interrupt type as an index into the interrupt vector table.

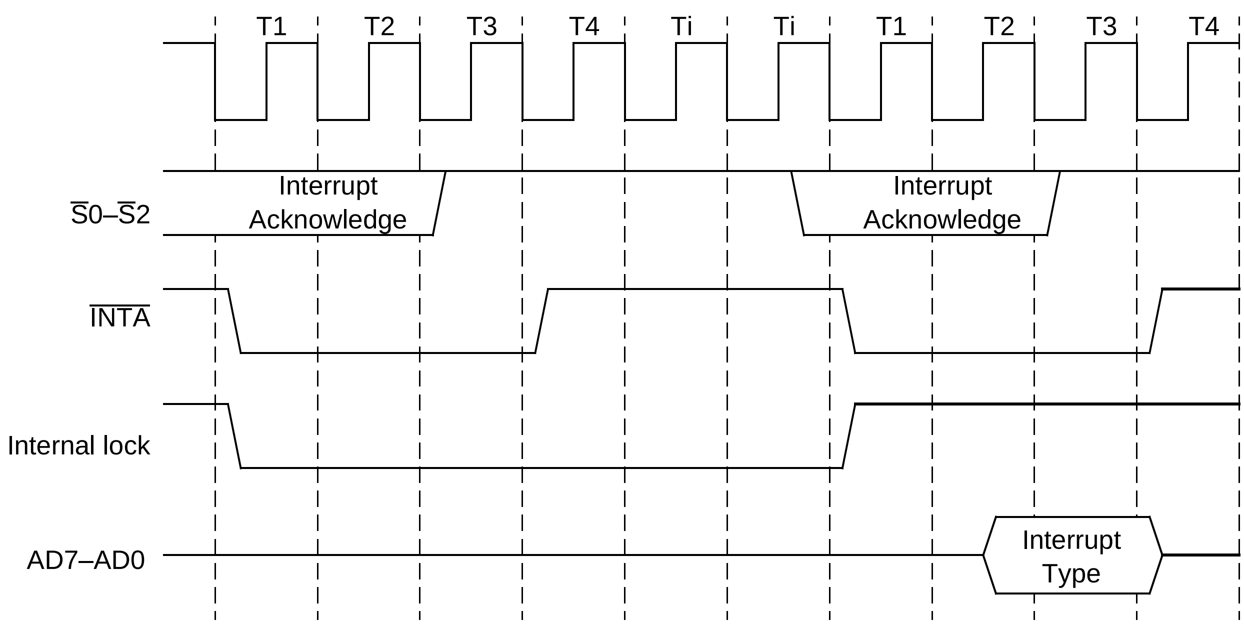
When the internal interrupt controller is supplying the interrupt type, no bus cycles are generated. The only external indication that an interrupt is being serviced is the processor reading the interrupt vector table.

When an external interrupt controller is supplying the interrupt type, the processor generates two interrupt acknowledge bus cycles (see Figure 7-1). The interrupt type is written to the AD7–AD0 lines by the external interrupt controller during the second bus cycle.

Interrupt acknowledge bus cycles have the following characteristics:

- The two interrupt acknowledge cycles are internally locked. (There is no $\overline{\text{LOCK}}$ pin on the Am186EM and Am188EM microcontrollers.)
- Two idle states are always inserted between the two cycles.
- Wait states are inserted if READY is not returned to the processor.

Figure 7-1 External Interrupt Acknowledge Bus Cycles



Notes:

1. *ALE is active for each INTA cycle.*
2. *$\overline{\text{RD}}$ is inactive.*

7.1.6 Interrupt Controller Reset Conditions

On reset, the interrupt controller performs the following nine actions:

1. All special fully nested mode (SFNM) bits are reset, implying fully nested mode.
2. All priority (PR) bits in the various control registers are set to 1. This places all sources at the lowest priority (level 7).
3. All level-triggered mode (LTM) bits are reset to 0, resulting in edge-triggered mode.
4. All interrupt in-service bits are reset to 0.
5. All interrupt request bits are reset to 0.
6. All mask (MSK) bits are set to 1. All interrupts are masked.
7. All cascade (C) bits are reset to 0 (non-cascade).
8. The interrupt priority mask is set to 7, allowing interrupts of all priorities.
9. The interrupt controller is initialized to master mode.

7.2 MASTER MODE OPERATION

This section describes master mode operation of the internal interrupt controller. See section 7.4 on page 7-28 for a description of slave mode operation.

Six pins are provided for external interrupt sources. One of these pins is NMI, the non-maskable interrupt. NMI is generally used for unusual events like power failure. The other five pins can be configured in any of the following ways:

- Fully nested mode—five interrupt lines with internally-generated interrupt types
- Cascade mode one—an interrupt line and interrupt acknowledge line pair with externally-generated interrupt types, plus three interrupt input lines with internally-generated types
- Cascade mode two—two pairs of interrupt and interrupt acknowledge lines with externally-generated interrupt types, and one interrupt input line (INT4) with internally-generated type

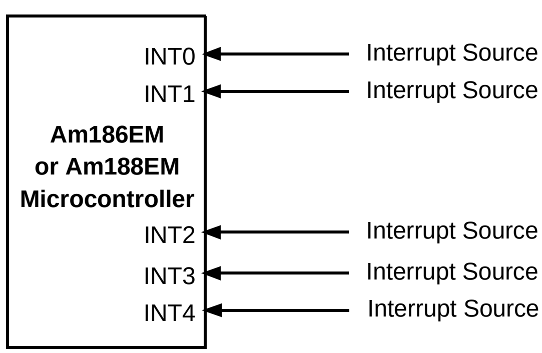
The basic modes of operation of the interrupt controller in master mode are similar to the 82C59A. The interrupt controller responds identically to internal interrupts in all three modes, the difference is only in the interpretation of function of the five external interrupt pins. The interrupt controller is set into one of these modes by programming the correct bits in the INT0 and INT1 control registers. The modes of interrupt controller operation are fully nested mode, cascade mode, special fully nested mode, and polled mode.

7.2.1 Fully Nested Mode

In fully nested mode, five pins are used as direct interrupt requests as in Figure 7-2. The interrupt types for these five inputs are generated internally. An in-service bit is provided for every interrupt source. If a lower-priority device requests an interrupt while the in-service bit (IS) is set for a higher priority interrupt, no interrupt is generated by the interrupt controller. In addition, if another interrupt request occurs from the same interrupt source while the in-service bit is set, no interrupt is generated by the interrupt controller. This allows interrupt service routines operating with interrupts enabled to be suspended only by interrupts of equal or higher priority than the in-service interrupt.

When an interrupt service routine is completed, the proper IS bit must be reset by writing the interrupt type to the EOI register. This is required to allow subsequent interrupts from this interrupt source and to allow servicing of lower-priority interrupts. A write to the EOI register should be executed at the end of the interrupt service routine just before the return from interrupt instruction.

Figure 7-2 Fully Nested (Direct) Mode Interrupt Controller Connections



7.2.2 Cascade Mode

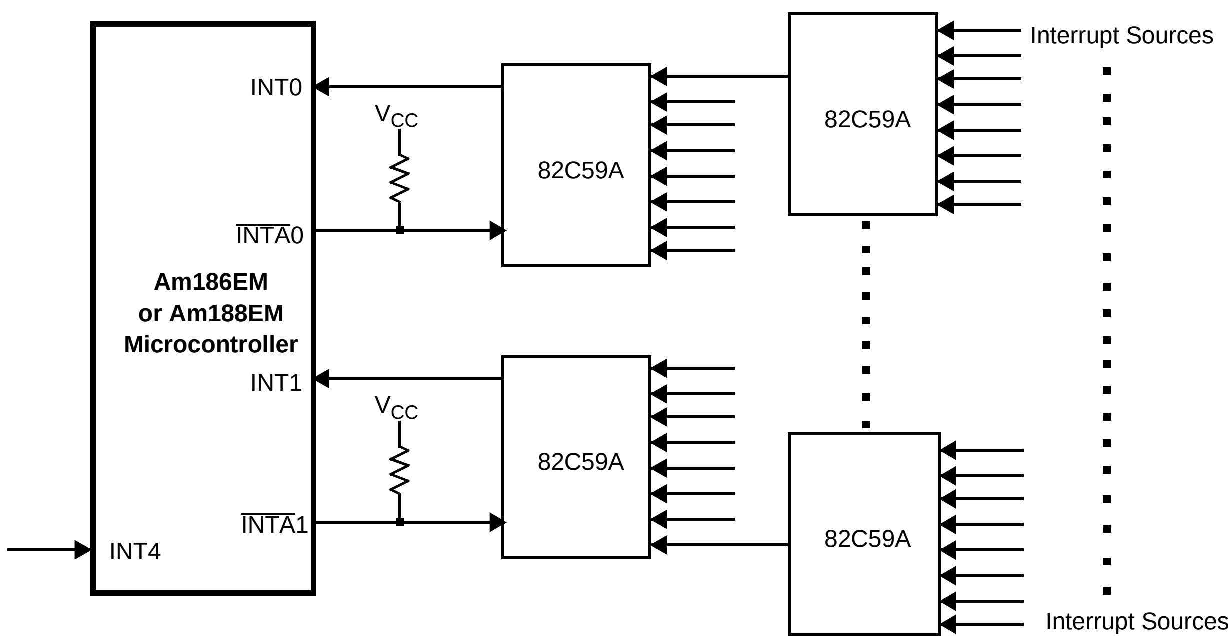
The Am186EM and Am188EM microcontrollers have five interrupt pins, two of which (INT2 and INT3) have dual functions. In fully nested mode, the five pins are used as direct interrupt inputs and the corresponding interrupt types are generated internally. In cascade mode, four of the five pins can be configured into interrupt input and dedicated acknowledge signal pairs. INT0 can be configured with interrupt acknowledge $\overline{\text{INTA}}$0 (INT2). INT1 can be configured with interrupt acknowledge $\overline{\text{INTA}}$1 (INT3).

External sources in cascade mode use externally generated interrupt types. When an interrupt is acknowledged, two $\overline{\text{INTA}}$ cycles are initiated and the type is read into the microcontroller on the second cycle (see section 7.1.5 on page 7-7). The capability to interface to one or two external 82C59A programmable interrupt controllers is provided when the inputs are configured in cascade mode.

Figure 7-3 shows the interconnection for cascade mode. INT0 is an interrupt input interfaced to one 82C59A, and INT2/$\overline{\text{INTA}}$0 serves as the dedicated interrupt acknowledge signal to that peripheral. INT1 and INT3/$\overline{\text{INTA}}$1 are also interfaced to an 82C59A. Each interrupt and acknowledge pair can be selectively placed in the cascade or non-cascade mode by programming the proper value into the INT0 and INT1 control registers. The dedicated acknowledge signals eliminate the need for external logic to generate $\overline{\text{INTA}}$ and device select signals.

Cascade mode provides the capability to serve up to 128 external interrupt sources through the use of external master and slave 82C59As. Three levels of priority are created, requiring priority resolution in the microcontroller interrupt controller, the master 82C59As, and the slave 82C59As. If an external interrupt is serviced, one IS bit is set at each of these levels. When the interrupt service routine is completed, up to three end-of-interrupt (EOI) register writes must be issued by the program.

Figure 7-3 Cascade Mode Interrupt Controller Connections



7.2.3 Special Fully Nested Mode

Specially fully nested mode is entered by setting the SFNM bit in the INT0 or INT1 control registers. (See section 7.3.1 on page 7-13.) It enables complete nesting with external 82C59A masters or multiple interrupts from the same external interrupt pin when not in cascade mode. In this case, the ISRs must be re-entrant.

In fully nested mode, an interrupt request from an interrupt source is not recognized when the in-service bit for that source is set. In this case, if more than one interrupt source is connected to an external interrupt controller, all of the interrupts go through the same Am186EM or Am188EM microcontroller interrupt request pin. As a result, if the external interrupt controller receives a higher-priority interrupt, its interrupt is not recognized by the microcontroller until the in-service bit is reset.

In special fully nested mode, the microcontroller's interrupt controller allows the processor to take interrupts from an external pin regardless of the state of the in-service bit for an interrupt source in order to allow multiple interrupts from a single pin. An in-service bit continues to be set, however, to inhibit interrupts from other lower-priority Am186EM or Am188EM microcontroller interrupt sources.

In special fully nested mode with cascade mode, when a write is issued to the EOI register at the end of the interrupt service routine, software polling of the IS register in the external master 82C59A must determine if there is more than one IS bit set. If so, the IS bit in the microcontroller remains active and the next ISR is entered.

7.2.4 Operation in a Polled Environment

To allow reading of the Poll register information without setting the indicated in-service bit, the Am186EM and Am188EM microcontrollers provide a Poll Status register (Figure 7-15) in addition to the Poll register. Poll register information is duplicated in the Poll Status register, but the Poll Status register can be read without setting the associated in-service bit. These registers are located in two adjacent memory locations in the peripheral control block.

The interrupt controller can be used in polled mode if interrupts are not desired. When polling, interrupts are disabled and software polls the interrupt controller as required. The interrupt controller is polled by reading the Poll Status register (Figure 7-15). Bit 15 in the Poll Status register indicates to the processor that an interrupt of high enough priority is requesting service. Bits 4–0 indicate to the processor the interrupt type of the highest priority source requesting service. After determining that an interrupt is pending, software reads the Poll register (rather than the Poll Status register), which causes the in-service bit of the highest priority source to be set.

7.2.5 End-of-Interrupt Write to the EOI Register

A program must write to the EOI register to reset the in-service (IS) bit when an interrupt service routine is completed. There are two types of writes to the EOI register—specific EOI and non-specific EOI (see section 7.3.14 on page 7-27).

Non-specific EOI does not specify which IS bit is to be reset. Instead, the interrupt controller automatically resets the IS bit of the highest priority source with an active service routine.

Specific EOI requires the program to send the interrupt type to the interrupt controller to indicate the source IS bit that is to be reset. Specific reset is applicable when interrupt nesting is possible or when the highest priority IS bit that was set does not belong to the service routine in progress.

7.3 MASTER MODE INTERRUPT CONTROLLER REGISTERS

The interrupt controller registers for master mode are shown in Table 7-2. All the registers can be read and written unless otherwise specified.

Registers can be redefined in slave mode. See section 7.4 on page 7-28 for detailed information regarding slave mode register usage. On reset, the microcontroller is in master mode. Bit 14 of the relocation register (see Figure 4-2) must be set to initiate slave mode operation.

Table 7-2 Interrupt Controller Registers in Master Mode

Offset	Register Mnemonic	Register Name	Associated Pins	Comments
3Ah	I1CON	INT1 Control	INT1	
38h	I0CON	INT0 Control	INT0	
3Eh	I3CON	INT3 Control	INT3	
3Ch	I2CON	INT2 Control	INT2	
40h	I4CON	INT4 Control	INT4	
36h	DMA1CON	DMA1 Interrupt Control	DRQ1	
34h	DMA0CON	DMA0 Interrupt Control	DRQ0	
32h	TCUCON	Timer Interrupt Control	TMRIN1 TMRIN0 TMROUT1 TMROUT0	
42h	WDCON	Watchdog Timer Interrupt Control		
44h	SPICON	Serial Port Interrupt Control	TXD, RXD	
30h	INTSTS	Interrupt Status		
2Eh	REQST	Interrupt Request	INT4–INT0 DRQ1–DRQ0	Read-only register
2Ch	INSERV	In-Service	INT4–INT0 DRQ1–DRQ0	
2Ah	PRIMSK	Priority Mask		
28h	IMASK	Interrupt Mask	INT4–INT0 DRQ1–DRQ0	
26h	POLLST	Poll Status		Read-only register
24h	POLL	Poll		Read-only register
22h	EOI	End of Interrupt		Write-only register

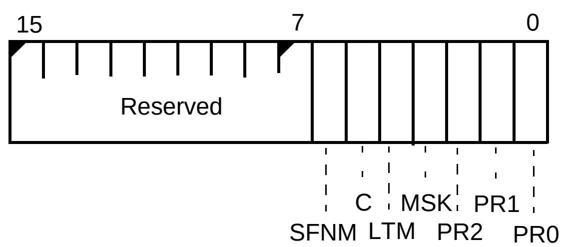
7.3.1 INT0 and INT1 Control Registers (I0CON, Offset 38h, I1CON, Offset 3Ah) (Master Mode)

The INT0 interrupt is assigned to interrupt type 0Ch. The INT1 interrupt is assigned to interrupt type 0Dh.

When cascade mode is enabled for INT0 by setting the C bit of I0CON to 1, the INT2 pin becomes $\overline{\text{INTA}}$0, the interrupt acknowledge for INT0.

When cascade mode is enabled for INT1 by setting the C bit of I1CON to 1, the INT3 pin becomes $\overline{\text{INTA}}$1, the interrupt acknowledge for INT1.

Figure 7-4 INT0 and INT1 Control Registers (I0CON, I1CON, offsets 38h and 3Ah)



The value of I0CON and I1CON at reset is 000Fh.

Bits 15–7: Reserved—Set to 0.

Bit 6: Special Fully Nested Mode (SFNM)—When set to 1, enables special fully nested mode.

Bit 5: Cascade Mode (C)—When set to 1, this bit enables cascade mode.

Bit 4: Level-Triggered Mode (LTM)—This bit determines whether the microcontroller interprets an INT0 or INT1 interrupt request as edge- or level-sensitive. A 1 in this bit configures INT0 or INT1 as an active High, level-sensitive interrupt. A 0 in this bit configures INT0 or INT1 as a Low-to-High, edge-triggered interrupt. In either case, INT0 or INT1 must remain High until they are acknowledged.

Bit 3: Mask (MSK)—This bit determines whether the INT0 or INT1 signal can cause an interrupt. A 1 in this bit masks this interrupt source, preventing INT0 or INT1 from causing an interrupt. A 0 in this bit enables INT0 or INT1 interrupts.

This bit is duplicated in the Interrupt Mask register. See the Interrupt Mask register in section 7.3.11 on page 7-24.

Bits 2–0: Priority Level (PR2–PR0)—This field determines the priority of INT0 or INT1 relative to the other interrupt signals, as shown in Table 7-3 on page 7-14.

Table 7-3 Priority Level

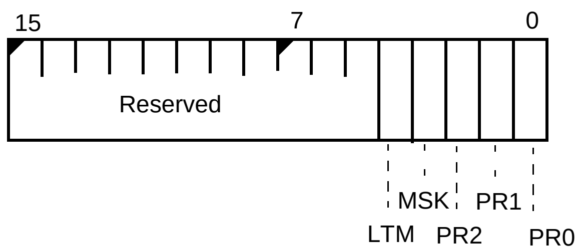
Priority	PR2–PR0
(High) 0	0 0 0b
1	0 0 1b
2	0 1 0b
3	0 1 1b
4	1 0 0b
5	1 0 1b
6	1 1 0b
(Low) 7	1 1 1b

7.3.2 INT2 and INT3 Control Registers (I2CON, Offset 3Ch, I3CON, Offset 3Eh) (Master Mode)

The INT2 interrupt is assigned to interrupt type OEh. The INT3 interrupt is assigned to interrupt type 0Fh.

The INT2 and INT3 pins can be configured as interrupt acknowledge pins $\overline{\text{INTA}}$0 and $\overline{\text{INTA}}$1 when cascade mode is implemented.

Figure 7-5 INT2 and INT3 Control Registers (I2CON, I3CON, offsets 3Ch and 3Eh)



The value of I2CON and I3CON at reset is 000Fh.

Bits 15–5: Reserved—Set to 0.

Bit 4: Level-Triggered Mode (LTM)—This bit determines whether the microcontroller interprets an INT2 or INT3 interrupt request as edge- or level-sensitive. A 1 in this bit configures INT2 or INT3 as an active High, level-sensitive interrupt. A 0 in this bit configures INT2 or INT3 as a Low-to-High, edge-triggered interrupt. In either case, INT2 or INT3 must remain High until they are acknowledged.

Bit 3: Mask (MSK)—This bit determines whether the INT2 or INT3 signal can cause an interrupt. A 1 in this bit masks this interrupt source, preventing INT2 or INT3 from causing an interrupt. A 0 in this bit enables INT2 or INT3 interrupts.

This bit is duplicated in the Interrupt Mask register. See the Interrupt Mask register in section 7.3.11 on page 7-24.

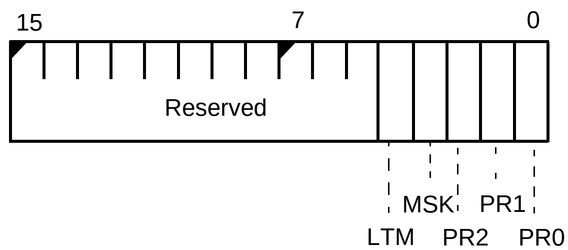
Bits 2–0: Priority Level (PR2–PR0)—This field determines the priority of INT2 or INT3 relative to the other interrupt signals, as shown in Table 7-3 on page 7-14.

7.3.3 INT4 Control Register (I4CON, Offset 40h) (Master Mode)

The Am186EM and Am188EM microcontrollers provide INT4, an additional external interrupt pin. This input behaves like INT3–INT0 on the 80C186/188 microcontroller with the exception that INT4 is only intended for use as a nested-mode interrupt source.

This interrupt is assigned to interrupt type 10h. The Interrupt 4 Control register (see Figure 7-6) controls the operation of the INT4 signal.

Figure 7-6 INT4 Control Register (I4CON, offset 40h)



The value of I4CON at reset is 000Fh.

Bits 15–5: Reserved—Set to 0.

Bit 4: Level-Triggered Mode (LTM)—This bit determines whether the microcontroller interprets an INT4 interrupt request as edge- or level-sensitive. A 1 in this bit configures INT4 as an active High, level-sensitive interrupt. A 0 in this bit configures INT4 as a Low-to-High, edge-triggered interrupt. In either case, INT4 must remain High until it is acknowledged.

Bit 3: Mask (MSK)—This bit determines whether the INT4 signal can cause an interrupt. A 1 in this bit masks this interrupt source, preventing INT4 from causing an interrupt. A 0 in this bit enables INT4 interrupts.

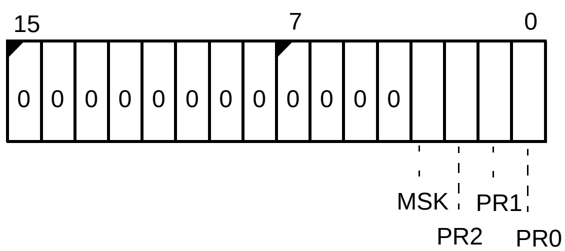
This bit is duplicated in the Interrupt Mask register. See the Interrupt Mask register in section 7.3.11 on page 7-24.

Bits 2–0: Priority (PR)—This field determines the priority of INT4 relative to the other interrupt signals, as shown in Table 7-3 on page 7-14.

7.3.4 Timer and DMA Interrupt Control Registers (TCUCON, Offset 32h, DMA0CON, Offset 34h, DMA1CON, Offset 36h) (Master Mode)

The three timer interrupts are assigned to interrupt type 08h, 12h, and 13h. All three timer interrupts are configured through TCUCON, offset 32h. The DMA0 interrupt is assigned to interrupt type 0Ah. The DMA1 interrupt is assigned to interrupt type 0Bh.

Figure 7-7 Timer/DMA Interrupt Control Registers (TCUCON, DMA0CON, DMA1CON, offsets 32h, 34h, and 36h)



The value of TCUCON, DMA0CON, and DMA1CON at reset is 000Fh.

Bits 15–4: Reserved—Set to 0.

Bit 3: Interrupt Mask (MSK)—This bit determines whether the corresponding signal can generate an interrupt. A 1 masks this interrupt source. A 0 enables the corresponding interrupt.

This bit is duplicated in the Interrupt Mask register. See the Interrupt Mask register in section 7.3.11 on page 7-24.

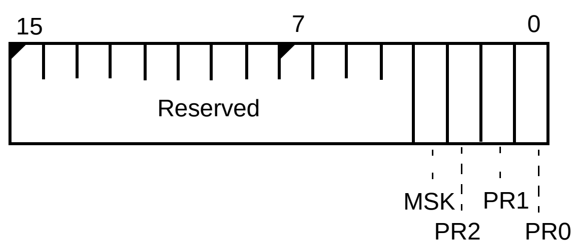
Bits 2–0: Priority Level (PR2–PR0)—Sets the priority level for its corresponding source. See Table 7-3 on page 7-14.

7.3.5 Watchdog Timer Interrupt Control Register (WDCON, Offset 42h) (Master Mode)

The Am186EM and Am188EM microcontrollers provide an additional on-chip interrupt source, the watchdog timer. This timer is constructed from existing 80C186 microcontroller pins. It is implemented by connecting the TMROUT1 output to an additional internal interrupt to create the watchdog timer interrupt. This interrupt is assigned to interrupt type 11h. The control register format is shown in Figure 7-8.

The systems programmer should program the timer (see section 8.2.2 on page 8-3) and then program the interrupt pin.

Figure 7-8 Watchdog Timer Interrupt Control Register (WDCON, offset 42h)



The value of WDCON at reset is 000Fh.

Bits 15–5: Reserved—Set to 0.

Bit 4: Reserved—*Must* be set to 0 to ensure proper operation of the Am186EM and Am188EM microcontrollers.

Bit 3: Mask (MSK)—This bit determines whether the watchdog timer can cause an interrupt. A 1 in this bit masks this interrupt source, preventing the watchdog timer from causing an interrupt. A 0 in this bit enables watchdog timer interrupts.

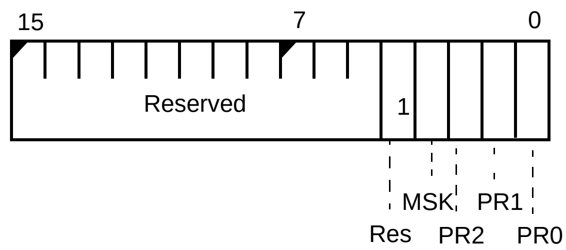
This bit is duplicated in the Interrupt Mask register. See the Interrupt Mask register in section 7.3.11 on page 7-24.

Bits 2–0: Priority (PR)—This field determines the priority of the watchdog timer relative to the other interrupt signals, as shown in Table 7-3 on page 7-14.

7.3.6 Serial Port Interrupt Control Register (SPICON, Offset 44h) (Master Mode)

The Serial Port Interrupt Control register controls the operation of the asynchronous serial port interrupt source (SPI, bit 10 in the Interrupt Request register). This interrupt is assigned to interrupt type 14h. The control register format is shown in Figure 7-9.

Figure 7-9 Serial Port Interrupt Control Register (SPICON, offset 44h)



The value of SPICON at reset is 001Fh.

Bits 15–5: Reserved—Set to 0.

Bit 4: Reserved—Set to 1.

Bit 3: Mask (MSK)—This bit determines whether the serial port can cause an interrupt. A 1 in this bit masks this interrupt source, preventing the serial port from causing an interrupt. A 0 in this bit enables serial port interrupts.

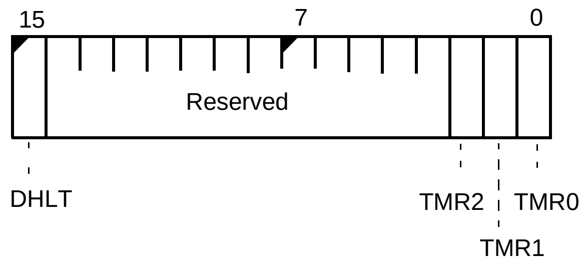
This bit is duplicated in the Interrupt Mask register. See the Interrupt Mask register in section 7.3.11 on page 7-24.

Bits 2–0: Priority (PR2–PR0)—This field determines the priority of the serial port relative to the other interrupt signals. After a reset, the priority is 7. See Table 7-3 on page 7-14.

7.3.7 Interrupt Status Register (INTSTS, Offset 30h) (Master Mode)

The Interrupt Status (INTSTS) register indicates the interrupt request status of the three timers.

Figure 7-10 Interrupt Status Register (INTSTS, offset 30h)



Bit 15: DMA Halt (DHLT)—When set to 1, halts any DMA activity. This pin is automatically set to 1 when non-maskable interrupts occur and is reset when an IRET instruction is executed. Time-critical software, such as interrupt handlers, can modify this bit directly to inhibit DMA transfers. Because of the function of this register as an interrupt request register for the timers, the DHLT bit should not be modified by software when timer interrupts are enabled.

Bits 14–3: Reserved

Bits 2–0: Timer Interrupt Request (TMR2–TMR0)—When set to 1, these bits indicate that the corresponding timer has an interrupt request pending. (Note that the timer TMR bit in the REQST register is the OR of these timer interrupt requests.)

7.3.8 Interrupt Request Register (REQST, Offset 2Eh) (Master Mode)

The hardware interrupt sources have interrupt request bits inside the interrupt controller. A read from this register yields the status of these bits. The Interrupt Request register is a read-only register. The format of the REQST register is shown in Figure 7-11.

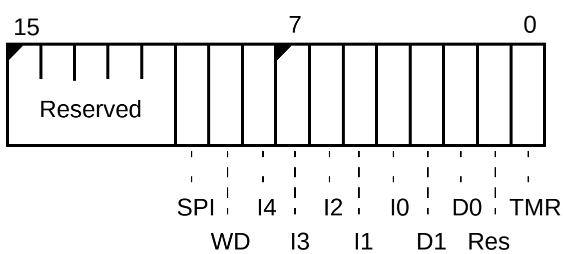
The Am186EM and Am188EM microcontrollers define three new bits to report the state of INT4, the Watchdog Timer, and the asynchronous serial port.

For internal interrupts (SPI, WD, D1, D0, and TMR), the corresponding bit is set to 1 when the device requests an interrupt. The bit is reset during the internally generated interrupt acknowledge.

For INT4–INT0 external interrupts, the corresponding bit (I4–I0) reflects the current value of the external signal. The device must hold this signal High until the interrupt is serviced.

Generally the interrupt service routine signals the external device to remove the interrupt request.

Figure 7-11 Interrupt Request Register (REQST, offset 2Eh)



The REQST register is undefined on reset.

Bits 15–11: Reserved

Bit 10: Serial Port Interrupt Request (SPI)—This bit indicates the interrupt state of the serial port. If enabled, the SPI bit is the logical OR of all possible serial port interrupt sources (THRE, RDR, BRKI, FER, PER, and OER status bits).

Bit 9: Watchdog Timer Interrupt Request (WD)—When this bit is set to 1, the Watchdog Timer has an interrupt pending.

Bits 8–4: Interrupt Requests (I4–I0)—When set to 1, the corresponding INT pin has an interrupt pending (i.e., when INT0 is pending, I0 is set). These bits reflect the status of the external pin.

Bits 3–2: DMA Channel Interrupt Request (D1–D0)—When set to 1, the corresponding DMA channel has an interrupt pending.

Bit 1: Reserved

Bit 0: Timer Interrupt Request (TMR)—This bit indicates the state of the timer interrupts. This bit is the logical OR of the timer interrupt requests. When set to a 1, this bit indicates that the timer control unit has an interrupt pending.

The Interrupt Status register indicates the specific timer that is requesting an interrupt. See section 7.3.7.

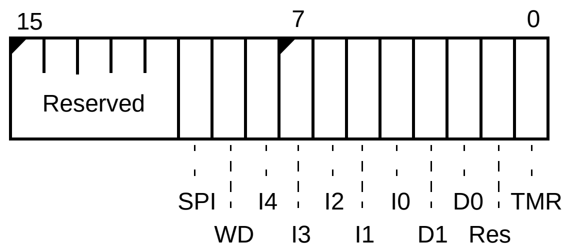
7.3.9 In-Service Register (INSERV, Offset 2Ch) (Master Mode)

The Am186EM and Am188EM microcontrollers define three new bits to report the in-service state of INT4, the Virtual Watchdog Timer, and the asynchronous serial port. The format of the modified In-Service register is shown in Figure 7-12.

The bits in the INSERV register are set by the interrupt controller when the interrupt is taken. Each bit in the register is cleared by writing the corresponding interrupt type to the End-of-Interrupt (EOI) register. See Table 7-1 on page 7-3.

When an in-service bit is set, the microcontroller will not generate an interrupt request for the associated source, preventing an interrupt from interrupting itself if interrupts are enabled in the ISR. Special fully nested mode allows the INT1–INT0 requests to circumvent this restriction for the INT0 and INT1 sources.

Figure 7-12 In-Service Register (INSERV, offset 2Ch)



The INSERV register is set to 0000h on reset.

Bits 15–11: Reserved

Bit 10: Serial Port Interrupt In-Service (SPI)—This bit indicates the in-service state of the asynchronous serial port.

Bit 9: Watchdog Timer Interrupt In-Service (WD)—This bit indicates the in-service state of the Watchdog Timer.

Bits 8–4: Interrupt In-Service (I4–I0)—These bits indicate the in-service state of the corresponding INT pin.

Bits 3–2: DMA Channel Interrupt In-Service (D1–D0)—These bits indicate the in-service state of the corresponding DMA channel.

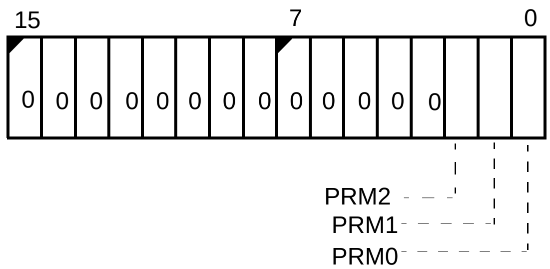
Bit 1: Reserved

Bit 0: Timer Interrupt In-Service (TMR)—This bit indicates the state of the in-service timer interrupts. This bit is the logical OR of all the timer interrupt status bits. When set to a 1, this bit indicates that the corresponding timer interrupt status bit is in-service.

7.3.10 Priority Mask Register (PRIMSK, Offset 2Ah) (Master Mode)

The Priority Mask (PRIMSK) register provides the value that determines the minimum priority level at which maskable interrupts can generate an interrupt.

Figure 7-13 Priority Mask Register (PRIMSK, offset 2Ah)



The value of PRIMSK at reset is 0007h.

Bits 15–3: Reserved—Set to 0.

Bits 2–0: Priority Field Mask (PRM2–PRM0)—This field determines the minimum priority that is required in order for a maskable interrupt source to generate an interrupt. Maskable interrupts with programmable priority values that are numerically higher than this field are masked. The possible values are zero (000b) to seven (111b).

A value of seven (111b) allows all interrupt sources that are not masked to generate interrupts. A value of five (101b) allows only unmasked interrupt sources with a programmable priority of zero to five (000b to 101b) to generate interrupts.

Table 7-4 Priority Level

Priority	PR2–PR0
(High) 0	0 0 0b
1	0 0 1b
2	0 1 0b
3	0 1 1b
4	1 0 0b
5	1 0 1b
6	1 1 0b
(Low) 7	1 1 1b

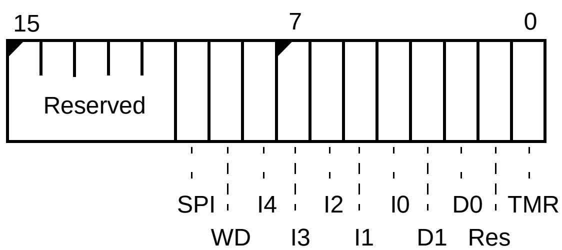
7.3.11 Interrupt Mask Register (IMASK, Offset 28h) (Master Mode)

The Am186EM and Am188EM microcontrollers define three new bits to report the mask state of the INT4 Control, Watchdog Timer Interrupt Control, and Serial Port Interrupt Control registers.

The Interrupt Mask (IMASK) register is a read/write register. Programming a bit in the IMASK register has the effect of programming the MSK bit in the associated control register. The format of the IMASK register is shown in Figure 7-14.

Do not write to the interrupt mask register while interrupts are enabled. To modify mask bits while interrupts are enabled, use the individual interrupt control registers.

Figure 7-14 Interrupt Mask Register (IMASK, offset 28h)



The IMASK register is set to 07FDh on reset.

Bits 15–11: Reserved

Bit 10: Serial Port Interrupt Mask (SPI)— When set to 1, this bit indicates that the asynchronous serial port interrupt is masked.

Bit 9: Virtual Watchdog Timer Interrupt Mask (WD)—When set to 1, this bit indicates that the Watchdog Timer interrupt is masked.

Bits 8–4: Interrupt Mask (I4–I0)—When set to 1, an I4–I0 bit indicates that the corresponding interrupt is masked.

Bits 3–2: DMA Channel Interrupt Masks (D1–D0)—When set to 1, a D1–D0 bit indicates that the corresponding DMA channel interrupt is masked.

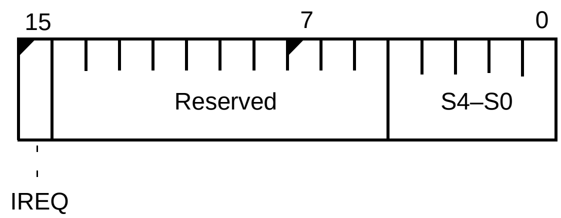
Bit 1: Reserved

Bit 0: Timer Interrupt Mask (TMR)—When set to 1, this bit indicates that interrupt requests from the timer control unit are masked.

7.3.12 Poll Status Register (POLLST, Offset 26h) (Master Mode)

The Poll Status (POLLST) register mirrors the current state of the Poll register. The POLLST register can be read without affecting the current interrupt request. But when the Poll register is read, the current interrupt is acknowledged and the next interrupt takes its place in the Poll register.

Figure 7-15 Poll Status Register (POLLST, offset 26h)



Bit 15: Interrupt Request (IREQ)—Set to 1 if an interrupt is pending. When this bit is set to 1, the S4–S0 field contains valid data.

Bits 14–5: Reserved—Set to 0.

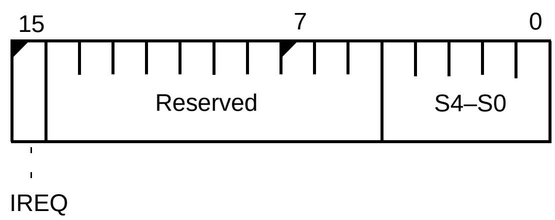
Bits 4–0: Poll Status (S4–S0)—Indicates the interrupt type of the highest priority pending interrupt.

7.3.13 Poll Register (POLL, Offset 24h) (Master Mode)

When the Poll register is read, the current interrupt is acknowledged and the next interrupt takes its place in the Poll register.

The Poll Status register mirrors the current state of the Poll register, but the Poll Status register can be read without affecting the current interrupt request.

Figure 7-16 Poll Register (POLL, offset 24h)



Bit 15: Interrupt Request (IREQ)—Set to 1 if an interrupt is pending. When this bit is set to 1, the S4–S0 field contains valid data.

Bits 14–5: Reserved—Set to 0.

Bits 4–0: Poll Status (S4–S0)—Indicates the interrupt type of the highest priority pending interrupt. Reading the Poll register acknowledges the highest priority pending interrupt and enables the next interrupt to advance into the register.

Although the IS bit is set, the interrupt service routine does not begin execution automatically. The application software must execute the appropriate ISR.

7.3.14 End-of-Interrupt Register (EOI, Offset 22h) (Master Mode)

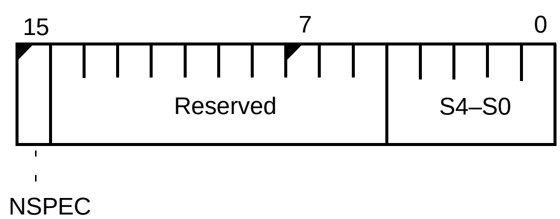
The End-of-Interrupt (EOI) register is a write-only register. The in-service flags in the In-Service register (see section 7.3.9 on page 7-22) are reset by writing to the EOI register. Before executing the IRET instruction that ends an interrupt service routine (ISR), the ISR should write to the EOI register to reset the IS bit for the interrupt.

The specific EOI reset is the most secure method to use for resetting IS bits. Figure 7-17 shows example code for a specific EOI reset. See Table 7-1 on page 7-3 for specific EOI values.

Figure 7-17 Example EOI Assembly Code

```
            ...
            ...                     ;ISR code
            ...
            mov dx, EOI_ADDR
exit:       mov ax,int_type         ;load the interrupt type in ax
            out dx,ax               ;write the interrupt type to EOI
            popa
            iret                    ;return from interrupt
```

Figure 7-18 End-of-Interrupt Register (EOI, offset 22h)



Bit 15: Non-Specific EOI (NSPEC)—The NSPEC bit determines the type of EOI command. When written as a 1, NSPEC indicates non-specific EOI. When written as a 0, NSPEC indicates the specific EOI interrupt type in S4–S0.

Bits 14–5: Reserved

Bits 4–0: Source EOI Type (S4–S0)—Specifies the EOI type of the interrupt that is currently being processed. See Table 7-1 on page 7-3.

7.4 SLAVE MODE OPERATION

When slave mode is used, the microcontroller's internal interrupt controller is used as a slave controller to an external master interrupt controller. The internal interrupts are monitored by the internal interrupt controller, while the external controller functions as the system master interrupt controller.

On reset, the microcontroller is in master mode. To activate slave mode operation, bit 14 of the relocation register must be set (see Figure 4-2 on page 4-4).

Because of pin limitations caused by the need to interface to an external 82C59A master, the internal interrupt controller does not accept external inputs. However, there are enough interrupt controller inputs (internally) to dedicate one to each timer. In slave mode, each timer interrupt source has its own mask bit, IS bit, and control word.

The INT4, watchdog timer, and serial port interrupts are not available in slave mode. In slave mode, each peripheral must be assigned a unique priority to ensure proper interrupt controller operation. The programmer must assign correct priorities and initialize interrupt control registers before enabling interrupts.

7.4.1 Slave Mode Interrupt Nesting

Slave mode operation allows nesting of interrupt requests. When an interrupt is acknowledged, the priority logic masks off all priority levels except those with equal or higher priority.

7.4.2 Slave Mode Interrupt Controller Registers

The Interrupt Controller Registers for slave mode are shown in Table 7-5. All registers can be read and written, unless specified otherwise.

Table 7-5 Interrupt Controller Registers in Slave Mode

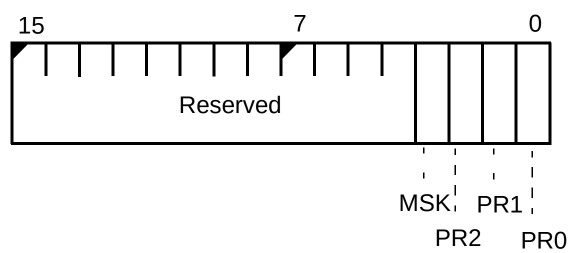
Offset	Register Mnemonic	Register Name	Affected Pins	Comments
3Ah	T2INTCON	Timer 2 Interrupt Control		Interrupt Type XXXXX101
38h	T1INTCON	Timer 1 Interrupt Control	TMRIN1 TMROUT1	Interrupt Type XXXXX100
36h	DMA1CON	DMA 1 Interrupt Control		Interrupt Type XXXXX011
34h	DMA0CON	DMA 0 Interrupt Control		Interrupt Type XXXXX010
32h	T0INTCON	Timer 0 Interrupt Control	TMRIN0 TMROUT0	Interrupt Type XXXXX000
30h	INTSTS	Interrupt Status		
2Eh	REQST	Interrupt Request		Read Only
2Ch	INSERV	In-Service		Read Only
2Ah	PRIMSK	Priority Mask		
28h	IMASK	Interrupt Mask		
22h	EOI	Specific EOI		Write Only
20h	INTVEC	Interrupt Vector		

7.4.3 Timer and DMA Interrupt Control Registers (T0INTCON, Offset 32h, T1INTCON, Offset 38h, T2INTCON, Offset 3Ah, DMA0CON, Offset 34h, DMA1CON, Offset 36h) (Slave Mode)

In slave mode, there are three separate registers for the three timers. In master mode, all three timers are masked and prioritized in one register, TCUCON.

In slave mode, the two DMA control registers retain their functionality and addressing from master mode.

Figure 7-19 Timer and DMA Interrupt Control Registers (T0INTCON, T1INTCON, T2INTCON, DMA0CON, DMA1CON, offsets 32h, 38h, 3Ah, 34h, and 36h)



These registers are set to 000Fh on reset.

Bits 15–4: Reserved—Set to 0.

Bit 3: Mask (MSK)—This bit determines whether the interrupt source can cause an interrupt. A 1 in this bit masks the interrupt source, preventing the source from causing an interrupt. A 0 in this bit enables interrupts from the source.

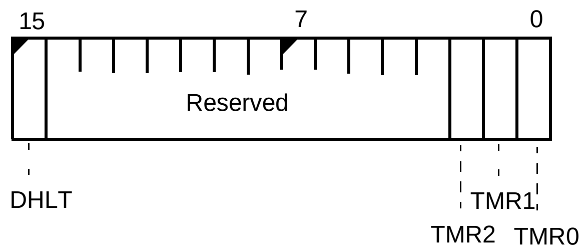
This bit is duplicated in the Interrupt Mask register. See the Interrupt Mask register in section 7.4.8 on page 7-34.

Bits 2–0: Priority Level (PR2–PR0)—This field determines the priority of the interrupt source relative to the other interrupt signals, as shown in Table 7-3 on page 7-14.

7.4.4 Interrupt Status Register (INTSTS, Offset 30h) (Slave Mode)

The Interrupt Status register controls DMA activity when non-maskable interrupts occur and indicates the current interrupt status of the three timers.

Figure 7-20 Interrupt Status Register (INTSTS, offset 30h)



The INTSTS register is set to 0000h on reset.

Bit 15: DMA Halt (DHLT)—When set to 1, halts any DMA activity. Automatically set to 1 when non-maskable interrupts occur and reset when an IRET instruction is executed.

Bits 14–3: Reserved

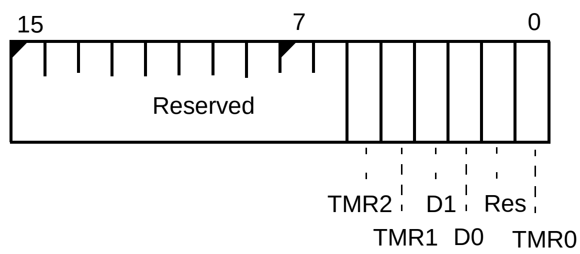
Bits 2–0: Timer Interrupt Request (TMR2–TMR0)—When set to 1, indicates the corresponding timer has an interrupt request pending.

7.4.5 Interrupt Request Register (REQST, Offset 2Eh) (Slave Mode)

The internal interrupt sources have interrupt request bits inside the interrupt controller. A read from this register yields the status of these bits. The Interrupt Request register is a read-only register. The format of the Interrupt Request register is shown in Figure 7-21.

For internal interrupts (D1, D0, TMR2, TMR1, and TMR0), the corresponding bit is set to 1 when the device requests an interrupt. The bit is reset during the internally generated interrupt acknowledge.

Figure 7-21 Interrupt Request Register (REQST, offset 2Eh)



The REQST register is set to 0000h on reset.

Bits 15–6: Reserved

Bits 5–4: Timer 2/Timer 1 Interrupt Request (TMR2–TMR1)—When set to 1, these bits indicate the state of any interrupt requests from the associated timer.

Bits 3–2: DMA Channel Interrupt Request (D1–D0)—When set to 1, D1–D0 indicate that the corresponding DMA channel has an interrupt pending.

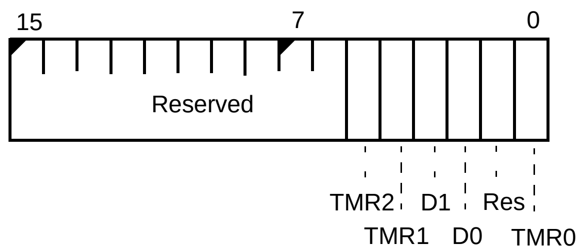
Bit 1: Reserved

Bit 0: Timer 0 Interrupt Request (TMR0)—When set to 1, this bit indicates the state of an interrupt request from Timer 0.

7.4.6 In-Service Register (INSERV, Offset 2Ch) (Slave Mode)

The format of the In-Service register is shown in Figure 7-22. The bits in the In-Service register are set by the interrupt controller when the interrupt is taken. The in-service bits are cleared by writing to the End-of-Interrupt (EOI) register.

Figure 7-22 In-Service Register (INSERV, offset 2Ch)



The INSERV register is set to 0000h on reset.

Bits 15–6: Reserved

Bits 5–4: Timer 2/Timer 1 Interrupt In-Service (TMR2–TMR1)—When set to 1, these bits indicate that the corresponding timer interrupt is currently being serviced.

Bits 3–2: DMA Channel Interrupt In-Service (D1–D0)—When set to 1, the corresponding DMA channel is currently being serviced.

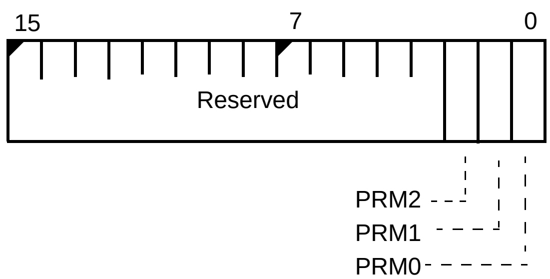
Bit 1: Reserved

Bit 0: Timer 0 Interrupt In-Service (TMR0)—When set to 1, this bit indicates Timer 0 is currently being serviced.

7.4.7 Priority Mask Register (PRIMSK, Offset 2Ah) (Slave Mode)

The format of the Priority Mask register is shown in Figure 7-23. The Priority Mask register provides the value that determines the minimum priority level at which maskable interrupts can generate an interrupt.

Figure 7-23 Priority Mask Register (PRIMSK, offset 2Ah)



The value of the PRIMSK register at reset is 0007h.

Bits 15–3: Reserved

Bits 2–0: Priority Field Mask (PRM2–PRM0)—This field determines the minimum priority which is required in order for a maskable interrupt source to generate an interrupt.

A value of seven (111b) allows all interrupt sources that are not masked to generate interrupts. A value of five (101b) allows only unmasked interrupt sources with a programmable priority of zero to five (000b to 101b) to generate interrupts.

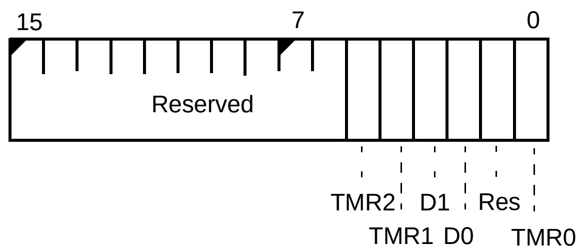
Table 7-6 Priority Level

Priority	PR2–PR0
(High) 0	0 0 0b
1	0 0 1b
2	0 1 0b
3	0 1 1b
4	1 0 0b
5	1 0 1b
6	1 1 0b
(Low) 7	1 1 1b

7.4.8 Interrupt Mask Register (IMASK, Offset 28h) (Slave Mode)

The format of the Interrupt Mask register is shown in Figure 7-24. The Interrupt Mask register is a read/write register. Programming a bit in the Interrupt Mask register has the effect of programming the MSK bit in the associated control register.

Figure 7-24 Interrupt Mask Register (IMASK, offset 28h)



The IMASK register is set to 003Dh on reset.

Bits 15–6: Reserved

Bits 5–4: Timer 2/Timer 1 Interrupt Mask (TMR2–TMR1)—These bits indicate the state of the mask bit of the Timer Interrupt Control register and when set to a 1, indicate which source has its interrupt requests masked.

Bits 3–2: DMA Channel Interrupt Mask (D1–D0)—These bits indicate the state of the mask bits of the corresponding DMA control register.

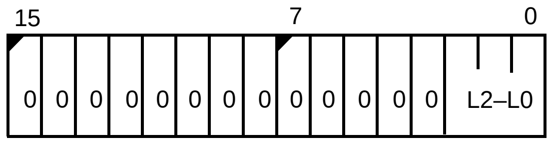
Bit 1: Reserved

Bit 0: Timer 0 Interrupt Mask (TMR0)—This bit indicates the state of the mask bit of the Timer Interrupt Control register and when set to a 1, indicates Timer 0 has its interrupt request masked.

7.4.9 Specific End-of-Interrupt Register (EOI, Offset 22h) (Slave Mode)

In slave mode, a write to the EOI register resets an in-service bit of a specific priority. The user supplies a three-bit priority-level value that points to an in-service bit to be reset. The command is executed by writing the correct value in the Specific EOI register at offset 22h.

Figure 7-25 Specific End-of-Interrupt Register (EOI, offset 22h)



The EOI register is undefined on reset.

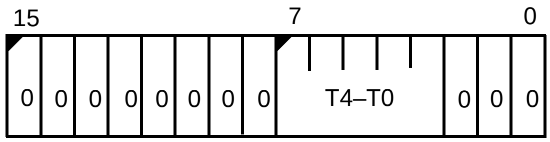
Bits 15–3: Reserved—Write as 0.

Bits 2–0: Interrupt Type (L2–L0)—Encoded value indicating the priority of the IS (interrupt service) bit to be reset. Writes to these bits cause an EOI to be issued for the interrupt type in slave mode. Write-only register.

7.4.10 Interrupt Vector Register (INTVEC, Offset 20h) (Slave Mode)

Vector generation in slave mode is exactly like that of an 8259A or 82C59A slave. The interrupt controller generates an 8-bit interrupt type that the CPU shifts left two bits (multiplies by four) to generate an offset into the interrupt vector table.

Figure 7-26 Interrupt Vector Register (INTVEC, offset 20h)



The INTVEC register is undefined on reset.

Bits 15–8: Reserved—Read as 0.

Bits 7–3: Interrupt Type (T4–T0)—Sets the five most significant bits of the interrupt types for the internal interrupt type. The interrupt controller itself provides the lower three bits of the interrupt type, as determined by the priority level of the interrupt request. See Table 7-5 on page 7-15.

Bits 2–0: Reserved—Read as 0.

CHAPTER

8 TIMER CONTROL UNIT

8.1 OVERVIEW

There are three 16-bit programmable timers in the Am186EM and Am188EM microcontrollers. Timers 0 and 1 are highly versatile and are each connected to two external pins (each one has an input and an output). These two timers can be used to count or time external events, or they can be used to generate nonrepetitive or variable-duty-cycle waveforms. Timer 1 can also be configured as a watchdog timer.

The watchdog timer provides a mechanism for detecting software crashes or hangs. The TMROUT1 output is internally connected to the watchdog timer interrupt. Software developers must first program the TIMER1 Mode/Control, Count, and Max Count registers, and then program the Watchdog Timer Interrupt Control register (see Figure 7-8 on page 7-18). The TIMER1 Count register must be reloaded at intervals less than the TIMER1 max count to assure the watchdog interrupt is not taken. If the code crashes or hangs, the TIMER1 countdown can cause a watchdog interrupt.

Timer 2 is not connected to any external pins. It can be used for real-time coding and time-delay applications. It can also be used as a prescale to timer 0 and timer 1 or as a DMA request source.

8.2 PROGRAMMABLE REGISTERS

The timers are controlled by eleven 16-bit registers (see Table 8-1) that are located in the peripheral control block.

Table 8-1 Timer Control Unit Register Summary

Offset from PCB	Register Mnemonic	Register Name
56h	T0CON	Timer 0 Mode/Control
5Eh	T1CON	Timer 1 Mode/Control
66h	T2CON	Timer 2 Mode/Control
50h	T0CNT	Timer 0 Count
58h	T1CNT	Timer 1 Count
60h	T2CNT	Timer 2 Count
52h	T0CMPA	Timer 0 Maxcount Compare A
54h	T0CMPB	Timer 0 Maxcount Compare B
5Ah	T1CMPA	Timer 1 Maxcount Compare A
5Ch	T0CMPB	Timer 1 Maxcount Compare B
62h	T2CMPA	Timer 2 Maxcount Compare A

The timer-count registers contain the current value of a timer. The timer-count registers can be read or written at any time, regardless of whether the corresponding timer is running. The microcontroller increments the value of a timer-count register each time a timer event occurs.

Each timer also has a corresponding maximum-count register that defines the maximum value for the timer. When the timer reaches the maximum value, it resets to 0 during the same clock cycle. (The value in the timer-count register never equals the maximum-count register.) In addition, timers 0 and 1 have a secondary maximum-count register. Using both the primary and secondary maximum-count registers lets the timer alternate between two maximum values.

If the timer is programmed to use only the primary maximum-count register, the timer output pin switches Low for one clock cycle, the clock cycle after the maximum value is reached. If the timer is programmed to use both of its maximum-count registers, the output pin creates a waveform by indicating which maximum-count register is currently in control. The duty cycle and frequency of the waveform depend on the values in the alternating maximum-count registers.

8.2.1 Timer Operating Frequency

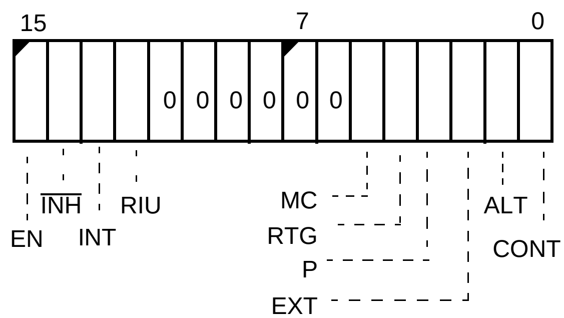
Each timer is serviced on every fourth clock cycle. Therefore, a timer can operate at a maximum speed of one-quarter of the internal clock frequency. A timer can be clocked externally at the same maximum frequency of one-fourth of the internal clock frequency. However, because of internal synchronization and pipelining of the timer circuitry, the timer output takes up to six clock cycles to respond to the clock or gate input.

The timers are run by the processor's internal clock. If power-save mode is in effect, the timers operate at the reduced power-save clock rate.

8.2.2 Timer 0 and Timer 1 Mode and Control Registers (T0CON, Offset 56h, T1CON, Offset 5Eh)

These registers control the functionality of timer 0 and timer 1. See Figure 8-1.

Figure 8-1 Timer 0 and Timer 1 Mode and Control Registers (T0CON, T1CON, offsets 56h and 5Eh)



The value of T0CON and T1CON at reset is 0000h.

Bit 15: Enable Bit (EN)—When set to 1, the timer is enabled. When set to 0, the timer is inhibited from counting. This bit can only be written with the $\overline{\text{INH}}$ bit set at the same time.

Bit 14: Inhibit Bit ($\overline{\text{INH}}$)—Allows selective updating of enable (EN) bit. When set to 1 during a write, EN can also be modified. When set to 0 during a write, writes to EN are ignored. This bit is not stored and is always read as 0.

Bit 13: Interrupt Bit (INT)—When set to 1, an interrupt request is generated when the count register equals a maximum count. If the timer is configured in dual maxcount mode, an interrupt is generated each time the count reaches maxcount A or maxcount B. When INT is set to 0, the timer will not issue interrupt requests. If the enable bit is cleared after an interrupt request has been generated but before the pending interrupt is serviced, the interrupt request will still be present.

Bit 12: Register in Use Bit (RIU)—When the Maxcount Compare A register is being used for comparison to the timer count value, this bit is set to 0. When the Maxcount Compare B register is being used, this bit is set to 1.

Bits 11–6: Reserved—Set to 0.

Bit 5: Maximum Count Bit (MC)—The MC bit is set to 1 when the timer reaches a maximum count. In dual maxcount mode, the bit is set each time either Maxcount Compare A or B register is reached. This bit is set regardless of the timer interrupt-enable bit. The MC bit can be used to monitor timer status through software polling instead of through interrupts.

Bit 4: Retrigger Bit (RTG)—Determines the control function provided by the timer input pin. When set to 1, a 0 to 1 edge transition on TMRIN0 or TMRIN1 resets the count. When set to 0, a High input enables counting and a Low input holds the timer value. This bit is ignored when external clocking (EXT=1) is selected.

Bit 3: Prescaler Bit (P)—When set to 1, the timer is prescaled by timer 2. When set to 0, the timer counts up every fourth CLKOUT period. This bit is ignored when external clocking is enabled (EXT=1).

Bit 2: External Clock Bit (EXT)—When set to 1, an external clock is used. When set to 0, the internal clock is used.

Bit 1: Alternate Compare Bit (ALT)—When set to 1, the timer counts to maxcount compare A, then resets the count register to 0. Then the timer counts to maxcount compare B, resets the count register to zero, and starts over with maxcount compare A.

If ALT is clear, the timer counts to maxcount compare A and then resets the count register to zero and starts counting again against maxcount compare A. In this case, maxcount compare B is not used.

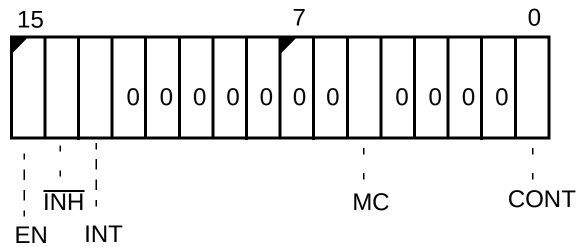
Bit 0: Continuous Mode Bit (CONT)—When set to 1, CONT causes the associated timer to run in the normal continuous mode.

When CONT is set to 0, EN is cleared after each timer count sequence and the timer clears and then halts on reaching the maximum count. If CONT=0 and ALT=1, the timer counts to the maxcount compare A register value and resets, then it counts to the B register value and resets and halts.

8.2.3 Timer 2 Mode and Control Register (T2CON, Offset 66h)

This register controls the functionality of timer 2. See Figure 8-2.

Figure 8-2 Timer 2 Mode and Control Register (T2CON, offset 66h)



The value of T2CON at reset is 0000h.

Bit 15: Enable Bit (EN)—When EN is set to 1, the timer is enabled. When set to 0, the timer is inhibited from counting. Do not write to this bit unless the $\overline{INH}$ bit is set to 1 during the same write.

Bit 14: Inhibit Bit ($\overline{INH}$)—Allows selective updating of enable (EN) bit. When INH is set to 1 during a write, EN can be modified on the same write. When INH is set to 0 during a write, writes to EN are ignored. This bit is not stored and is always read as 0.

Bit 13: Interrupt Bit (INT)—When INT is set to 1, an interrupt request is generated when the count register equals a maximum count. When INT is set to 0, the timer will not issue interrupt requests. If the EN enable bit is cleared after an interrupt request has been generated, but before the pending interrupt is serviced, the interrupt request remains active.

Bits 12–6: Reserved—Set to 0.

Bit 5: Maximum Count Bit (MC)—The MC bit is set to 1 when the timer reaches its maximum count. This bit is set regardless of the timer interrupt-enable bit. The MC bit can be used to monitor timer status through software polling instead of through interrupts.

Bits 4–1: Reserved—Set to 0.

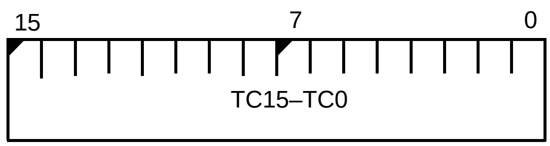
Bit 0: Continuous Mode Bit (CONT)—When CONT is set to 1, it causes the associated timer to run continuously. When set to 0, EN is cleared after each timer count sequence and the timer halts on reaching the maximum count.

8.2.4 Timer Count Registers (T0CNT, Offset 50h, T1CNT, Offset 58h, T2CNT, Offset 60h)

These registers can be incremented by one every four internal processor clocks. Timer 0 and timer 1 can also be configured to increment based on the TMRIN0 and TMRIN1 external signals, or they can be prescaled by timer 2. See Figure 8-3.

The count registers are compared to maximum count registers and various actions are triggered based on reaching a maximum count.

Figure 8-3 Timer Count Registers (T0CNT, T1CNT, T2CNT, offsets 50h, 58h, and 60h)



The value of these registers at reset is undefined.

Bits 15–0: Timer Count Value (TC15–TC0)—This register contains the current count of the associated timer. The count is incremented every fourth processor clock in internal clocked mode, or each time the timer 2 maxcount is reached if prescaled by timer 2. Timer 0 and timer 1 can be configured for external clocking based on the TMRIN0 and TMRIN1 signals.

8.2.5 Timer Maxcount Compare Registers (T0CMPA, Offset 52h, T0CMPB, Offset 54h, T1CMPA, Offset 5Ah, T1CMPB, Offset 5Ch, T2CMPA, Offset 62h)

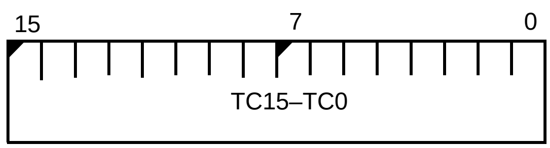
These registers serve as comparators for their associated count registers. Timer 0 and timer 1 each have two maximum count compare registers. See Figure 8-4.

Timer 0 and timer 1 can be configured to count and compare to register A and then count and compare to register B. Using this method, the TMROUT0 or TMROUT1 signals can be used to generate waveforms of various duty cycles.

Timer 2 has one compare register, T2CMPA.

If a maximum count compare register is set to 0000h, the timer associated with that compare register will count from 0000h to FFFFh before requesting an interrupt. With a 40-MHz clock, a timer configured this way interrupts every 6.5536 ms.

Figure 8-4 Timer Maxcount Compare Registers (T0CMPA, T0CMPB, T1CMPA, T1CMPB, T2CMPA, offsets 52h, 54h, 5Ah, 5Ch, and 62h)



The value of these registers at reset is undefined.

Bits 15–0: Timer Compare Value (TC15–TC0)—This register contains the maximum value a timer will count to before resetting its count register to 0.

CHAPTER

9 DMA CONTROLLER

9.1 OVERVIEW

Direct memory access (DMA) permits transfer of data between memory and peripherals without CPU involvement. The DMA unit in the Am186EM and Am188EM microcontrollers provides two high-speed DMA channels. Data transfers can occur between memory and I/O spaces (e.g., memory to I/O) or within the same space (e.g., memory-to-memory or I/O-to-I/O). Either bytes or words can be transferred to or from even or odd addresses on the Am186EM. (The Am188EM microcontroller does not support word transfers.) Two bus cycles (a minimum of eight clocks) are necessary for each data transfer.

Each channel accepts a DMA request from one of two sources: the channel request pin (DRQ1–DRQ0) or Timer 2. The two DMA channels can be programmed with different priorities to resolve simultaneous DMA requests, and transfers on one channel can interrupt the other channel.

9.2 DMA OPERATION

The format of the DMA control block is shown in Table 9-1. Six registers in the peripheral control block define the operation of each channel. The DMA registers consist of a 20-bit source address (2 registers), a 20-bit destination address (2 registers), a 16-bit transfer count register, and a 16-bit control register.

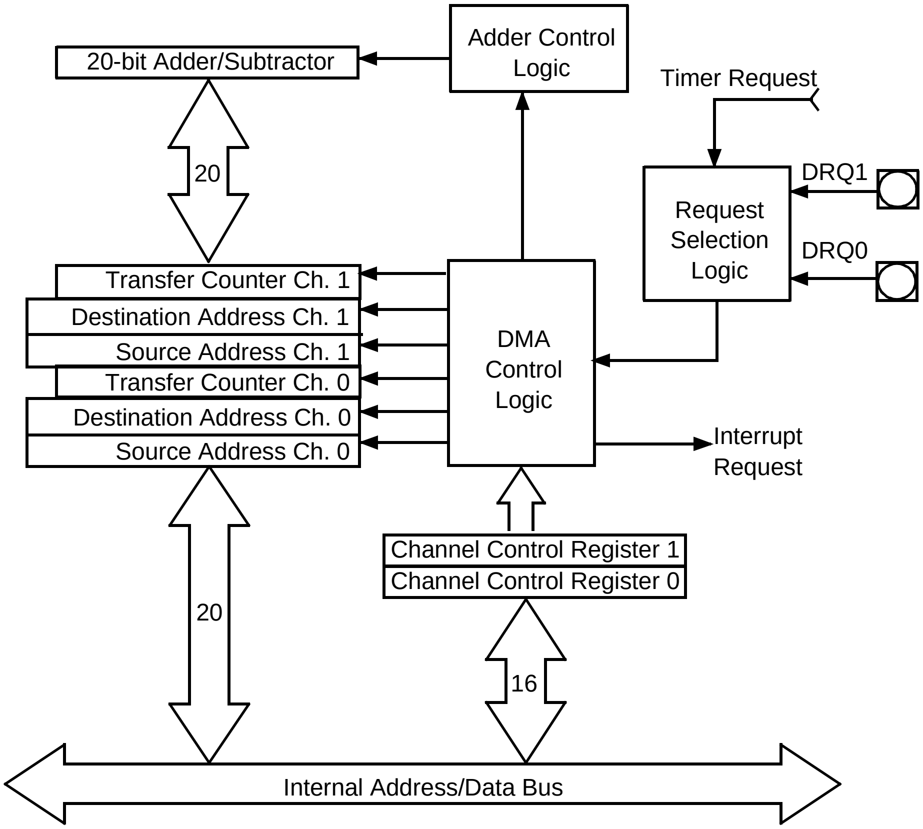
Table 9-1 DMA Controller Register Summary

Offset from PCB	Register Mnemonic	Register Name
CAh	D0CON	DMA 0 Control
DAh	D1CON	DMA 1 Control
C8h	D0TC	DMA 0 Transfer Count
D8h	D1TC	DMA 1 Transfer Count
C6h	D0DSTH	DMA 0 Destination Address High
D6h	D1DSTH	DMA 1 Destination Address High
C4h	D0DSTL	DMA 0 Destination Address Low
D4h	D1DSTL	DMA 1 Destination Address Low
C2h	D0SRCH	DMA 0 Source Address High
D2h	D1SRCH	DMA 1 Source Address High
C0h	D0SRCL	DMA 0 Source Address Low
D0h	D1SRCL	DMA 1 Source Address Low

The DMA transfer count register (DTC) specifies the number of DMA transfers to be performed. Up to 64 Kbytes or 64 Kwords can be transferred with automatic termination.

The DMA control registers define the channel operations (see Figure 9-1). All registers can be modified or altered during any DMA activity. Any changes made to these registers are reflected immediately in DMA operation.

Figure 9-1 DMA Unit Block Diagram



9.3 PROGRAMMABLE DMA REGISTERS

The sections on the following pages describe the control registers that are used to configure and operate the two DMA channels.

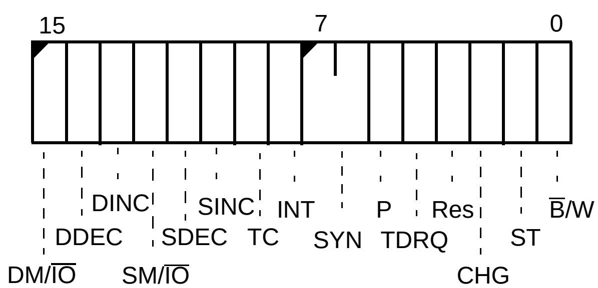
9.3.1 DMA Control Registers (D0CON, Offset CAh, D1CON, Offset DAh)

The DMA control registers (see Figure 9-2) determine the mode of operation for the DMA channels. These registers specify the following options:

- Whether the destination address is memory or I/O space
- Whether the destination address is incremented, decremented, or maintained constant after each transfer
- Whether the source address is memory or I/O space
- Whether the source address is incremented, decremented, or maintained constant after each transfer
- If DMA activity ceases after a programmed number of DMA cycles
- If an interrupt is generated after the last transfer
- The mode of synchronization
- The relative priority of one DMA channel with respect to the other DMA channel
- Whether timer 2 DMA requests are enabled or disabled
- Whether bytes or words are transferred

The DMA channel control registers can be changed while the channel is operating. Any changes made during DMA operations affect the current DMA transfer.

Figure 9-2 DMA Control Registers (D0CON, D1CON, offsets CAh and DAh)



The value of D0CON and D1CON at reset is FFF9h.

Bit 15: Destination Address Space Select (DM/IO)—Selects memory or I/O space for the destination address. When DM/IO is set to 1, the destination address is in memory space. When set to 0, the destination address is in I/O space.

Bit 14: Destination Decrement (DDEC)—When DDEC is set to 1, the destination address is automatically decremented after each transfer. The address decrements by 1 or 2, depending on the byte/word bit (B/W, bit 0). The address remains constant if the increment and decrement bits are set to the same value (00b or 11b).

Bit 13: Destination Increment (DINC)—When DINC is set to 1, the destination address is automatically incremented after each transfer. The address increments by 1 or 2, depending on the byte/word bit (B/W, bit 0). The address remains constant if the increment and decrement bits are set to the same value (00b or 11b).

Bit 12: Source Address Space Select (SM/IO)—When SM/IO is set to 1, the source address is in memory space. When set to 0, the source address is in I/O space.

Bit 11: Source Decrement (SDEC)—When SDEC is set to 1, the source address is automatically decremented after each transfer. The address decrements by 1 or 2 depending on the byte/word bit ($\overline{B}$/W, bit 0). The address remains constant if the increment and decrement bits are set to the same value (00b or 11b).

Bit 10: Source Increment (SINC)—When SINC is set to 1, the source address is automatically incremented after each transfer. The address increments by 1 or 2 depending on the byte/word bit ($\overline{B}$/W, bit 0). The address remains constant if the increment and decrement bits are set to the same value (00b or 11b).

Bit 9: Terminal Count (TC)—The DMA decrements the transfer count for each DMA transfer. When TC is set to 1, source or destination synchronized DMA transfers terminate when the count reaches 0. When TC is set to 0, source or destination synchronized DMA transfers do not terminate when the count reaches 0. Unsynchronized DMA transfers always terminate when the count reaches 0, regardless of the setting of this bit.

Bit 8: Interrupt (INT)—When INT is set to 1, the DMA channel generates an interrupt request on completion of the transfer count. The TC bit must also be set to generate an interrupt.

Bits 7–6: Synchronization Type (SYN1–SYN0)—The SYN1–SYN0 bits select channel synchronization as shown in Table 9-2. For more information on DMA synchronization, see section 9.4 on page 9-10.

Table 9-2 Synchronization Type

SYN1	SYN0	Sync Type
0	0	Unsynchronized
0	1	Source Synch
1	0	Destination Synch
1	1	Reserved

Bit 5: Relative Priority (P)—When P is set to 1, it selects high priority for this channel relative to the other channel during simultaneous transfers.

Bit 4: Timer Enable/Disable Request (TDRQ)—When TDRQ is set to 1, it enables DMA requests from timer 2. When set to 0, TDRQ disables DMA requests from timer 2.

Bit 3: Reserved

Bit 2: Change Start Bit (CHG)—This bit must be set to 1 during a write to allow modification of the ST bit. When CHG is set to 0 during a write, ST is not altered when writing the control word.

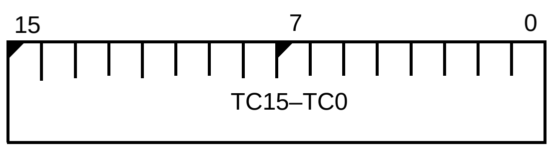
Bit 1: Start/Stop DMA Channel (ST)—The DMA channel is started when the start bit is set to 1. This bit can be modified only when the CHG bit is set to a 1 during the same register write.

Bit 0: Byte/Word Select ($\overline{B}$/W)—On the Am186EM microcontroller, when $\overline{B}$/W is set to 1, word transfers are selected. When $\overline{B}$/W is set to 0, byte transfers are selected. Word transfers are not supported on the Am188EM microcontroller.

9.3.2 DMA Transfer Count Registers (D0TC, Offset C8h, D1TC, Offset D8h)

Each DMA channel maintains a 16-bit DMA Transfer Count register (DTC). This register is decremented after every DMA cycle, regardless of the state of the TC bit in the DMA Control register. However, if the TC bit in the DMA control word is set or if unsynchronized transfers are programmed, DMA activity terminates when the Transfer Count register reaches 0.

Figure 9-3 DMA Transfer Count Registers (D0TC, D1TC, offsets C8h and D8h)



The value of D0TC and D1TC at reset is undefined.

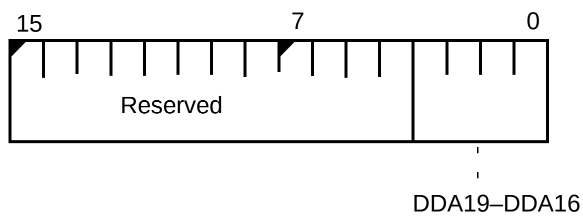
Bits 15–0: DMA Transfer Count (TC15–TC0)—Contains the transfer count for a DMA channel. Value is decremented by 1 after each transfer.

9.3.3 DMA Destination Address High Register (High Order Bits) (D0DSTH, Offset C6h, D1DSTH, Offset D6h)

Each DMA channel maintains a 20-bit destination and a 20-bit source register. Each register takes up two full 16-bit registers (the high register and the low register) in the peripheral control block. For each DMA channel to be used, all four registers must be initialized. These registers can be individually incremented or decremented after each transfer. If word transfers are performed, the address is incremented or decremented by 2 after each transfer. If byte transfers are performed, the address is incremented or decremented by 1.

Each register can point into either memory or I/O space. The user must program the upper four bits to 0000b in order to address the normal 64K I/O space. Since the DMA channels can perform transfers to or from odd addresses, there is no restriction on values for the destination and source address registers. Higher transfer rates can be achieved on the Am186EM microcontroller if all word transfers are performed to or from even addresses so that accesses occur in single, 16-bit bus cycles.

Figure 9-4 DMA Destination Address High Register (D0DSTH, D1DSTH, offsets C6h and D6h)



The value of D0DSTH and D1DSTH at reset is undefined.

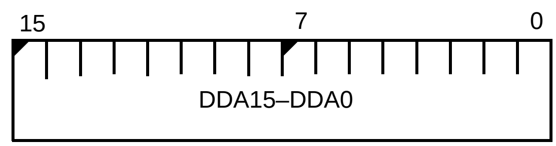
Bits 15–4: Reserved

Bits 3–0: DMA Destination Address High (DDA19–DDA16)—These bits are driven onto A19–A16 during the write phase of a DMA transfer.

9.3.4 DMA Destination Address Low Register (Low Order Bits) (D0DSTL, Offset C4h, D1DSTL, Offset D4h)

Figure 9-5 shows the DMA Destination Address Low register. The sixteen bits of this register are combined with the four bits of the DMA Destination Address High register (see Figure 9-4) to produce a 20-bit destination address.

Figure 9-5 DMA Destination Address Low Register (D0DSTL, D1DSTL, offsets C4h and D4h)



The value of D0DSTL and D1DSTL at reset is undefined.

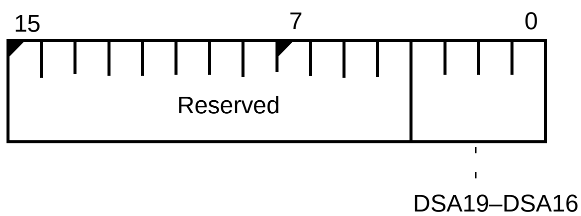
Bits 15–0: DMA Destination Address Low (DDA15–DDA0)—These bits are driven onto A15–A0 during the write phase of a DMA transfer.

9.3.5 DMA Source Address High Register (High Order Bits) (D0SRCH, Offset C2h, D1SRCH, Offset D2h)

Each DMA channel maintains a 20-bit destination and a 20-bit source register. Each register takes up two full 16-bit registers (the high register and the low register) in the peripheral control block. For each DMA channel to be used, all four registers must be initialized. These registers can be individually incremented or decremented after each transfer. If word transfers are performed, the address is incremented or decremented by 2 after each transfer. If byte transfers are performed, the address is incremented or decremented by 1.

Each register can point into either memory or I/O space. The user must program the upper four bits to 0000b in order to address the normal 64K I/O space. Since the DMA channels can perform transfers to or from odd addresses, there is no restriction on values for the destination and source address registers. Higher transfer rates can be achieved on the Am186EM microcontroller if all word transfers are performed to or from even addresses so that accesses occur in single, 16-bit bus cycles.

Figure 9-6 DMA Source Address High Register (D0SRCH, D1SRCH, offsets C2h and D2h)



The value of D0SRCH and D1SRCH at reset is undefined.

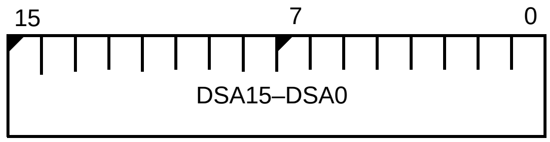
Bits 15–4: Reserved

Bits 3–0: DMA Source Address High (DSA19–DSA16)—These bits are driven onto A19–A16 during the read phase of a DMA transfer.

9.3.6 DMA Source Address Low Register (Low Order Bits) (D0SRCL, Offset C0h, D1SRCL, Offset D0h)

Figure 9-7 shows the DMA Source Address Low register. The sixteen bits of this register are combined with the four bits of the DMA Source Address High register (see Figure 9-6) to produce a 20-bit source address.

Figure 9-7 DMA Source Address Low Register (D0SRCL, D1SRCL, offsets C0h and D0h)



The value of D0SRCL and D1SRCL at reset is undefined.

Bits 15–0: DMA Source Address Low (DSA15–DSA0)—These bits are driven onto A15–A0 during the read phase of a DMA transfer.

9.4 DMA REQUESTS

Data transfers can be either source or destination synchronized—either the source of the data or the destination of the data can request the data transfer. DMA transfers can also be unsynchronized (i.e., the transfer takes place continually until the correct number of transfers has occurred).

During source synchronized or unsynchronized transfers, the DMA channel can begin a transfer immediately after the end of the previous DMA transfer, and a complete transfer can occur every two bus cycles or eight clock cycles (assuming no wait states).

When destination synchronization is performed, data is not fetched from the source address until the destination device signals that it is ready to receive it. When destination synchronized transfers are requested, the DMA controller relinquishes control of the bus after every transfer. If no other bus activity is initiated, another DMA cycle begins after two processor clocks. This allows the destination device time to remove its request if another transfer is not desired.

When the DMA controller relinquishes the bus during destination synchronized transfers, the CPU can initiate a bus cycle. As a result, a complete bus cycle is often inserted between destination-synchronized transfers. Table 9-3 shows the maximum DMA transfer rates based on the different synchronization strategies.

Table 9-3 Maximum DMA Transfer Rates

	Maximum DMA Transfer Rate (Mbytes/sec)			
Synchronization Type	**40 MHz**	**33 MHz**	**25 MHz**	**20 MHz**
Unsynchronized	10	8.25	6.25	5
Source Synch	10	8.25	6.25	5
Destination Synchronized (CPU needs bus)	6.6	5.5	4.16	3.3
Destination Synchronized (CPU does not need bus)	8	6.6	5	4

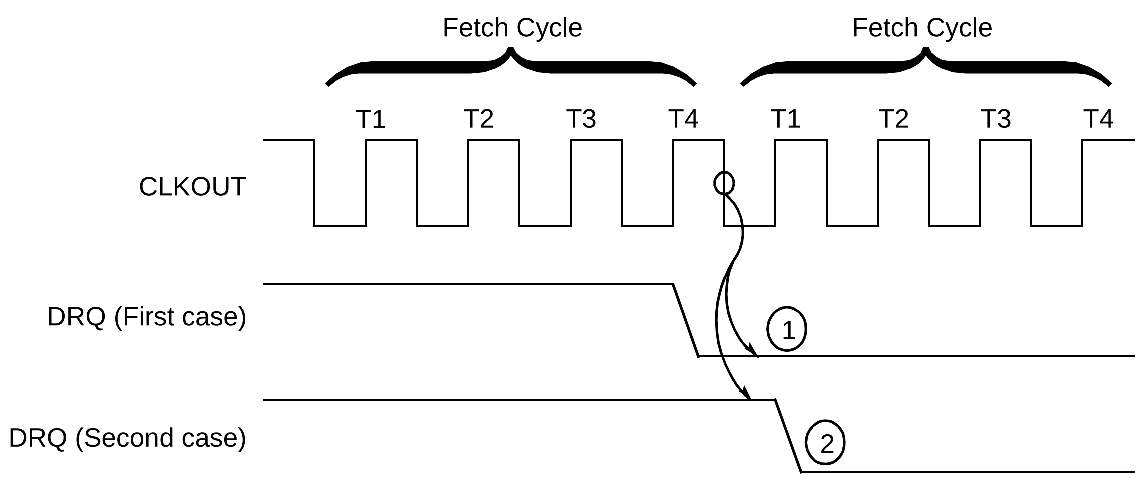
9.4.1 Synchronization Timing

DRQ1 or DRQ0 must be deasserted before the end of the DMA transfer to prevent another DMA cycle from occurring. The timing for the required deassertion depends on whether the transfer is source-synchronized or destination-synchronized.

9.4.1.1 Source Synchronization Timing

Figure 9-8 shows a typical source-synchronized DMA transfer. The DRQ signal must be deasserted at least four clocks before the end of the transfer (at T1 of the deposit phase). If more transfers are not required, a source-synchronized transfer allows the source device at least three clock cycles from the time it is acknowledged to deassert its DRQ line.

Figure 9-8 Source-Synchronized DMA Transfers



Notes:

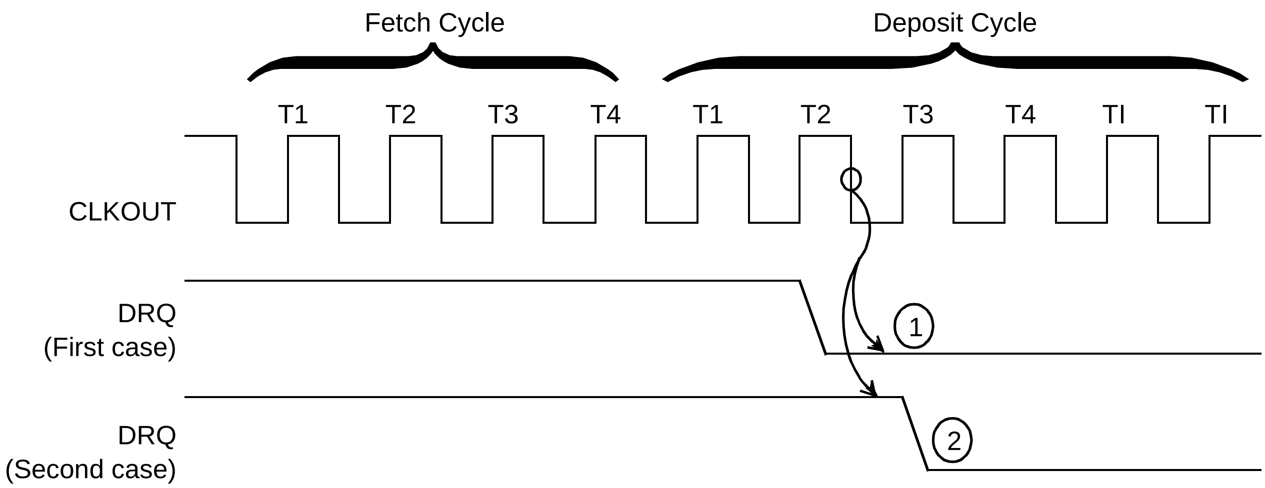
1. *This source-synchronized transfer is not followed immediately by another DMA transfer.*
2. *This source-synchronized transfer is immediately followed by another DMA transfer because DRQ is not deasserted soon enough.*

9.4.1.2 Destination Synchronization Timing

Figure 9-9 shows a typical destination-synchronized DMA transfer. A destination-synchronized transfer differs from a source-synchronized transfer in that two idle states are added to the end of the deposit cycle. The two idle states allow the destination device to deassert its DRQ signal four clocks before the end of the cycle. Without the two idle states, the destination device would not have time to deassert its DRQ signal.

Because of the two extra idle states, a destination-synchronized DMA channel allows other bus masters to take the bus during the idle states. The CPU, the refresh control unit, and another DMA channel can all access the bus during the idle states.

Figure 9-9 Destination Synchronized DMA Transfers



Notes:

1. *This destination-synchronized transfer is not followed immediately by another DMA transfer.*
2. *This destination-synchronized transfer is immediately followed by another DMA transfer because DRQ is not deasserted soon enough.*

9.4.2 DMA Acknowledge

No explicit DMA acknowledge signal is provided. Since both source and destination registers are maintained, a read from a requesting source or a write to a requesting destination should be used as the DMA acknowledge signal. Since the chip-select lines can be programmed to be active for a given block of memory or I/O space, and the DMA source and destination address registers can be programmed to point to the same given block, a chip-select line could be used to indicate a DMA acknowledge.

9.4.3 DMA Priority

The DMA channels can be programmed so that one channel is always given priority over the other, or they can be programmed to alternate cycles when both have DMA requests pending (see section 9.3.1, bit 5, the P bit). DMA cycles always have priority over internal CPU cycles except between internally locked memory accesses or word accesses to odd memory locations. However, an external bus hold takes priority over an internal DMA cycle.

Because an interrupt request cannot suspend a DMA operation and the CPU cannot access memory during a DMA cycle, interrupt latency time suffers during sequences of continuous DMA cycles. An NMI request, however, causes all internal DMA activity to halt. This allows the CPU to respond quickly to the NMI request.

9.4.4 DMA Programming

DMA cycles occur whenever the ST bit of the control register is set. If synchronized transfers are programmed, a DRQ must also be generated. Therefore, the source and destination transfer address registers and the transfer count register (if used) must be programmed before the ST bit is set.

Each DMA register can be modified while the channel is operating. If the CHG bit is set to 0 when the control register is written, the ST bit of the control register will not be modified by the write. If multiple channel registers are modified, an internally LOCKed string transfer should be used to prevent a DMA transfer from occurring between updates to the channel registers.

9.4.5 DMA Channels on Reset

On reset, the state of the DMA channels is as follows:

- The ST bit for each channel is reset.
- Any transfer in progress is aborted.
- The values of the transfer count registers, source address registers, and destination address registers are undefined.

CHAPTER

10 ASYNCHRONOUS SERIAL PORT

10.1 OVERVIEW

The Am186EM and Am188EM microcontrollers provide an asynchronous serial port. The asynchronous serial port is a two-pin interface that permits full-duplex bidirectional data transfer. The asynchronous serial port supports the following features:

- Full-duplex operation
- 7-bit or 8-bit data transfers
- Odd parity, even parity, or no parity
- 1 or 2 stop bits

If additional RS-232 signals are required, they can be created with available PIO pins (see section 12.1 on page 12-1). The asynchronous serial port transmit and receive sections are double-buffered. Break character recognition, framing, parity, and overrun error detection are provided. Exception interrupt generation is programmed by the user.

The transmit/receive clock is based on the internal processor clock internally divided down to the serial port operating frequency. If power-save mode is in effect, the divide factor must be reprogrammed. The serial port permits 7-bit and 8-bit data transfers. DMA transfers through the serial port are not supported.

The serial port generates one interrupt for all serial port events (transmit complete, data received, or error). The Serial Port Status register contains the reason for the serial port interrupt. The interrupt type assigned to the serial port is 14h.

The serial port can be used in power-save mode, but the transfer rate must be adjusted to correctly reflect the new internal operating frequency and the serial port must not receive any information until the frequency is changed.

10.2 PROGRAMMABLE REGISTERS

The asynchronous serial port is programmed through the use of five, 16-bit peripheral registers. See Table 10-1.

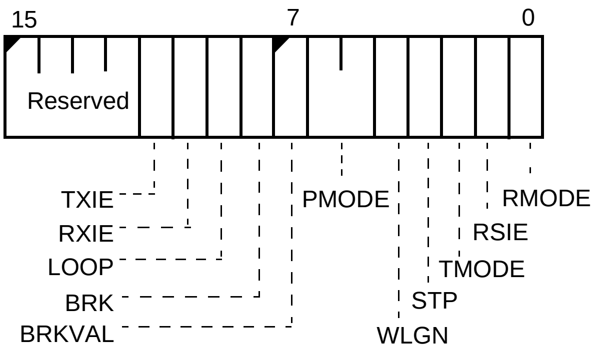
Table 10-1 Asynchronous Serial Port Register Summary

Offset from PCB	Register Mnemonic	Register Name
80h	SPCT	Serial Port Control
82h	SPSTS	Serial Port Status
84h	SPTD	Serial Port Transmit Data
86h	SPRD	Serial Port Receive Data
88h	SPBAUD	Serial Port Baud Rate Divisor

10.2.1 Serial Port Control Register (SPCT, Offset 80h)

The Serial Port Control register controls both the transmit and receive sections of the serial port. The format of the Serial Port Control register is shown in Figure 10-1.

Figure 10-1 Serial Port Control Register (SPCT, offset 80h)



The value of SPCT at reset is 0000h.

Bits 15–12: Reserved—Set to 0.

Bit 11: Transmit Holding Register Empty Interrupt Enable (TXIE)—This bit enables the serial port to generate an interrupt for the transmit holding register empty condition, indicating that the serial port is ready to accept a new character for transmission. If this bit is 1 and the Serial Port Transmit Holding register does not contain valid data, the serial port generates an interrupt request. The value of TXIE after power-on reset is 0.

Bit 10: Receive Data Ready Interrupt Enable (RXIE)—This bit enables the serial port to generate an interrupt for the receive data ready condition. If this bit is 1 and the Serial Port Receive Buffer register contains data that has been received on the serial port, the serial port generates an interrupt request. The value of RXIE after power-on reset is 0.

Bit 9: Loopback (LOOP)—Setting this bit to 1 places the serial port in the loopback mode. In this mode, the TXD output is set High and the transmit shift register is connected to the receive shift register. Data transmitted by the transmit section is immediately received by the receive section. The loopback mode is provided for testing the serial port. The value of LOOP after power-on reset is 0.

Bit 8: Send Break (BRK)—Setting this bit to 1 causes the serial port to send a continuous level on the TXD output. A break is a continuous Low on the TXD output for a duration of more than one frame transmission time. The level driven on the TXD output is determined by the BRKVAL bit.

To use the transmitter to time the frame, set the BRK bit when the transmitter is empty (indicated by the TEMT bit of the Serial Port Status register), write the serial port transmit holding register, then wait until the TEMT bit is again set before resetting the BRK bit. Since the TXD output is held constant while BRK is set, the data written to the transmit holding register will not appear on the pin. The value of BRK after power-on reset is 0.

Bit 7: Break Value (BRKVAL)—This bit determines the output value transmitted on the TXD pin during a send break operation. If BRKVAL is 1, a continuous High level is driven on the TXD output. If BRKVAL is 0, a continuous Low level is driven on the TXD output. Only a continuous Low value (BRKVAL=0) will result in a break being detected by the receiver. The value of BRKVAL after power-on reset is 0.

Bits 6–5: Parity Mode (PMODE)—This field specifies how parity generation and checking are performed during transmission and reception, as shown in Table 10-2.

Table 10-2 Parity Mode Bit Settings

Parity	PMODE
None (No parity bit in frame)	0 X
Odd (Odd number of 1s in frame)	1 0
Even (Even number of 1s in frame)	1 1

If parity checking and generation is selected, a parity bit is received or sent in addition to the specified number of data bits.

The value of PMODE after power-on reset is 00b.

Bit 4: Word Length (WLGN)—This bit determines the number of bits transmitted or received in a frame. If WLGN is 0, the serial port sends and receives 7 bits of data per frame. If WLGN is 1, the serial port sends and receives 8 bits of data per frame. The value of WLGN after power-on reset is 0.

Bit 3: Stop Bits (STP)—A 0 in the STP bit specifies that one stop bit is used to signify the end of a frame. A 1 in this bit specifies that two stop bits are used to signify the end of a frame. The value of STP after power-on reset is 0.

Bit 2: Transmit Mode (TMODE)—The TMODE bit enables data transmission and controls the operational mode of the serial port for the transmission of data. If TMODE is 0, the transmit section and transmit interrupts of the serial port are disabled. If TMODE is 1, the transmit section of the serial port is enabled. The value of TMODE after power-on reset is 0.

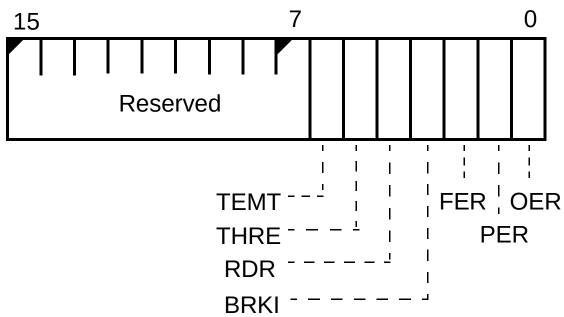
Bit 1: Receive Status Interrupt Enable (RSIE)—This bit enables the serial port to generate an interrupt because of an exception during reception. If this bit is 1 and the serial port receives a break, or experiences a framing error, parity error, or overrun error, the serial port generates a serial port interrupt. The value of RSIE after power-on reset is 0.

Bit 0: Receive Mode (RMODE)—This field enables data reception and controls the operational mode of the serial port for the reception of data. If RMODE is 0, the receive section and receive interrupts of the serial port are disabled. If RMODE is 1, the receive section of the serial port is enabled. The value of RMODE after power-on reset is 0.

10.2.2 Serial Port Status Register (SPSTS, Offset 82h)

The Serial Port Status register indicates the status of the transmit and receive sections of the serial port. The format of the Serial Port Status register is shown in Figure 10-2.

Figure 10-2 Serial Port Status Register (SPSTS, offset 82h)



Bits 15–7: Reserved—Set to 0.

Bit 6: Transmitter Empty (TEMT)—The TEMT bit is 1 when the transmitter has no data to transmit and the transmit shift register is empty. This indicates to software that it is safe to disable the transmit section. This bit is read-only.

Bit 5: Transmit Holding Register Empty (THRE)—When the THRE bit is 1, the transmit holding register contains invalid data and can be written with data to be transmitted. When the THRE bit is 0, the transmit holding register cannot be written because it contains valid data that has not yet been copied to the transmit shift register for transmission.

If transmit interrupts are enabled by the TMODE and TXIE fields, a serial port interrupt request is generated when the THRE bit is 1. The THRE bit is reset automatically by writing the transmit holding register. This bit is read-only, allowing other bits of the Serial Port Status register to be written (i.e., resetting the BRKI bit) without interfering with the current data request.

Bit 4: Receive Data Ready (RDR)—When the RDR bit is 1, the receive buffer register contains data that can be read. When the RDR bit is 0, the receive buffer register does not contain valid data. This bit is read-only.

If receive interrupts are enabled by the RMODE and RXIE fields, a serial port interrupt request is generated when the THRE bit is 1. Reading the receive buffer register resets the RDR bit.

Bit 3: Break Interrupt (BRKI)—The BRKI bit is set to indicate that a break has been received. If the RSIE bit is 1, the BRKI bit being set causes a serial port interrupt request. The BRKI bit should be reset by software.

Bit 2: Framing Error (FER)—The FER bit is set to indicate that a framing error occurred during reception of data. If the RSIE bit is 1, the FER bit being set causes a serial port interrupt request. The FER bit should be reset by software.

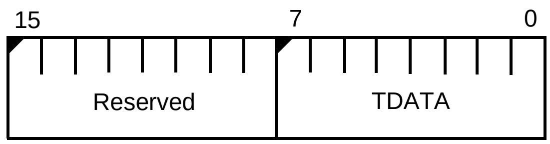
Bit 1: Parity Error (PER)—The PER bit is set to indicate that a parity error occurred during reception of data. If the RSIE bit is 1, the PER bit being set causes a serial port interrupt request. The PER bit should be reset by software.

Bit 0: Overrun Error (OER)—The OER bit is set when an overrun error occurs during reception of data. If the RSIE bit is 1, the OER bit being set causes a serial port interrupt request. The OER bit should be reset by software.

10.2.3 Serial Port Transmit Data Register (SPTD, Offset 84h)

Software writes this register (Figure 10-4) with data to be transmitted on the serial port. The transmitter is double-buffered, and the transmit section copies data from the transmit data register to the transmit shift register (which is not accessible to software) before transmitting the data.

Figure 10-3 Serial Port Transmit Data Register (SPTD, offset 84h)



The value of SPTD at reset is undefined.

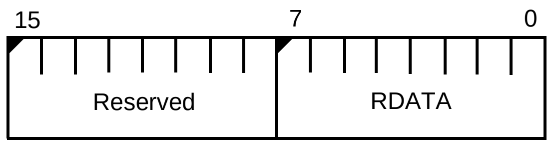
Bits 15–8: Reserved

Bit 7–0: Transmit Data (TDATA)—This field is written with data to be transmitted on the serial port. The THRE bit in the Serial Port Status register indicates whether there is valid data in the SPTD register. To avoid overwriting data in the SPTD register, the THRE bit should be read as a 1 before writing this register. Writing this register causes the THRE bit to be reset.

10.2.4 Serial Port Receive Data Register (SPRD, Offset 86h)

This register (Figure 10-4) contains data received over the serial port. The receiver is double-buffered, and the receive section can be receiving a subsequent frame of data in the receive shift register (which is not accessible to software) while the receive data register is being read by software.

Figure 10-4 Serial Port Receive Data Register (SPRD, offset 86h)



The value of SPRD at reset is undefined.

Bits 15–8: Reserved

Bits 7–0: Receive Data (RDATA)—This field contains data received on the serial port. The RDR bit of the Serial Port Status register indicates valid data in the SPRD register. To avoid reading invalid data, the RDR bit should be read as a 1 before the SPRD register is read. Reading this register causes the RDR bit to be reset.

10.2.5 Serial Port Baud Rate Divisor Register (SPBAUD, Offset 88h)

This register (Figure 10-5) specifies a clock divisor for the generation of the serial clock that controls the serial port. The serial clock rate is 16 times the baud rate of transmission or reception of data. The SPBAUD register specifies the number of internal processor cycles in one *phase* (half period) of the 16x serial clock.

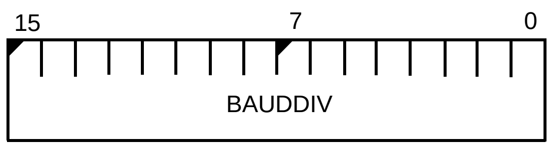
If power-save mode is in effect, the baud rate divisor must be reprogrammed to reflect the new processor clock frequency.

A general formula for the baud rate divisor is:

BAUDDIV=(Processor Frequency÷(32 · Baud Rate))–1

The maximum baud rate is 1/32 of the internal processor clock and is achieved by setting BAUDDIV=0000h. For a 40-MHz clock, a baud rate of 9600 can be achieved with BAUDDIV=129 (81h). A 1% error applies.

Figure 10-5 Serial Port Baud Rate Divisor Register (SPBAUD, offset 88h)



The value of SPBAUD at reset is undefined.

Bits 15–0: Baud Rate Divisor (BAUDDIV)—This field specifies the divisor for the internal processor clock that generates one phase (half period) of the serial clock. The serial clock operates at 16 times the data transmission or reception baud rate.

Table 10-3 shows baud rate divisors for a range of common baud rates and processor clock rates.

Table 10-3 Serial Port Baud Rate Table

Baud Rate	Divisor Based on CPU Clock Rate			
	20 MHz	**25 MHz**	**33 MHz**	**40 MHz**
300	2082	2603	3471	4165
600	1040	1301	1735	2082
1200	519	650	867	1040
2400	259	324	433	519
4800	129	161	216	259
9600	64	80	107	129
14,400	42	53	71	85
19,200	31	39	53	64
625 Kbaud	0	N/A	N/A	1
781.25 Kbaud	N/A	0	N/A	N/A
1.041 Mbaud	N/A	N/A	0	N/A
1.25 Mbaud	N/A	N/A	N/A	0

CHAPTER

11 SYNCHRONOUS SERIAL INTERFACE

11.1 OVERVIEW

The synchronous serial interface lets the Am186EM and Am188EM microcontrollers communicate with application-specific integrated circuits (ASICs) that require programmability but are short on pins. The four-pin interface permits half-duplex, bidirectional data transfer at speeds of up to 20 Mbit/s with a 40-MHz CPU clock.

Unlike the asynchronous serial port, the SSI operates in a master/slave configuration. The Am186EM and Am188EM microcontrollers operate as the master port.

The SSI interface provides four pins for communicating with system components: two enables (SDEN0 and SDEN1), a clock (SCLK), and a data pin (SDATA). Five registers (see Table 11-1) are used to control and monitor the interface.

- The Synchronous Serial Status register (SSS) reports the current port status.
- The Synchronous Serial Control register (SSC) sets the port clock rate and controls the enable signals.
- There are two data transmit registers—the Synchronous Serial Transmit 0 register (SSD0) and the Synchronous Serial Transmit 1 register (SSD1)—but data is transmitted and received over a single pin (SDATA).
- The Synchronous Serial Receive Register (SSR) holds data received over the SSI.

Table 11-1 Synchronous Serial Interface Register Summary

Offset from PCB	Register Mnemonic	Register Name
10h	SSS	Synchronous Serial Status
12h	SSC	Synchronous Serial Control
14h	SSD1	Synchronous Serial Transmit 1
16h	SSD0	Synchronous Serial Transmit 0
18h	SSR	Synchronous Serial Receive

11.1.1 Four-Pin Interface

The SDEN1–SDEN0 enable pins can be enabled for up to two peripheral devices.

Transmit and receive operations are synchronized between the master (Am186EM or Am188EM microcontroller) and slave (peripheral) by means of the SCLK output. SCLK is derived from the processor internal clock divided by 2, 4, 8, or 16, as specified by the SSC register. SCLK is only driven during data transmit or receive operations. The inactive state of SCLK is High.

If power-save mode is in effect, the SCLK frequency is affected by the reduced processor clock frequency.

Data is transferred across the SDATA input/output pin. Data is driven on the falling edge of SCLK and latched on the rising edge of SCLK. The least-significant bit of the data is shifted first for both transmit and receive operations. During write operations, the processor holds data for one-half of an SCLK period following the transfer of the last data bit. SDATA has a weak keeper that holds the last value of SDATA on the pin.

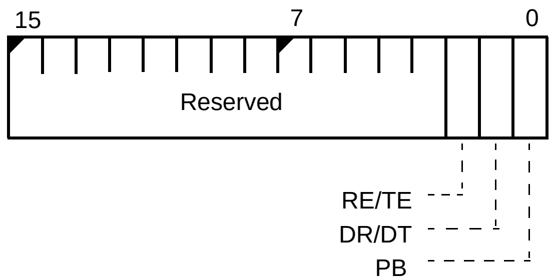
11.2 PROGRAMMABLE REGISTERS

The registers documented on the following pages are accessible to the system programmer.

11.2.1 Synchronous Serial Status Register (SSS, Offset 10h)

This read-only register indicates the state of the SSI port. The format of the Synchronous Serial Status register is shown in Figure 11-1.

Figure 11-1 Synchronous Serial Status Register (SSS, offset 10h)



The value of the SSS register at reset is 0000h.

Bits 15–3: Reserved—Set to 0.

Bit 2: Receive/Transmit Error Detect (RE/TE)—This bit is set when the SSI detects either a read of the Synchronous Serial Receive register or a write to one of the transmit registers while the SSI is busy (PB=1). This bit is reset when the SDEN output is inactive (bits DE1–DE0 in the SSC register are both 0).

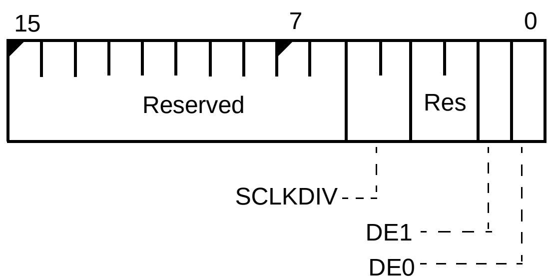
Bit 1: Data Receive/Transmit Complete (DR/DT)—The DR/DT bit is set at the end of the transfer of data bit 7 (SCLK rising edge) during a transmit or receive operation. This bit is reset when the SSR register is read, when one of the SSD0 or SSD1 registers is written, when the SSS register is read (unless the SSI completes an operation and sets the bit in the same cycle), or when both SDEN0 and SDEN1 become inactive.

Bit 0: SSI Port Busy (PB)—When the PB bit is set, a transmit or receive operation is in progress. When PB is reset, the port is ready to transmit or receive data.

11.2.2 Synchronous Serial Control Register (SSC, Offset 12h)

This read/write register controls the operation of the SDEN0–SDEN1 outputs and the transfer rate of the SSI port. The SDEN0 and SDEN1 outputs are asserted when a 1 is written to the corresponding bit. However, in the case when both DE0 and DE1 are set, only SDEN0 will be asserted. The format of the Synchronous Serial Control register is shown in Figure 11-2.

Figure 11-2 Synchronous Serial Control Register (SSC, offset 12h)



The value of the SSC register at reset is 0000h.

Bits 15–6: Reserved—Set to 1.

Bits 5–4: SCLK Divide (SCLKDIV)—These bits determine the SCLK frequency. SCLK is derived from the internal processor clock by dividing by 2, 4, 8, or 16. Table 11-2 shows the processor clock frequency divider values for the possible SCLKDIV settings.

If power-save mode is in effect, the SCLK frequency is affected by the reduced processor clock frequency.

Table 11-2 SCLK Divider Values

SCLKDIV	SCLK Frequency Divider
00b	Processor clock / 2
01b	Processor clock / 4
10b	Processor clock / 8
11b	Processor clock / 16

Bits 3–2: Reserved—Set to 0.

Bit 1: SDEN1 Enable (DE1)—When this bit is set to 1, the SDEN1 pin is held High. When DE1 is set to 0, the SDEN1 pin is Low.

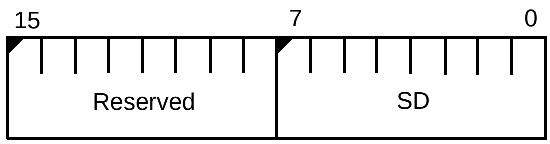
Bit 0: SDEN0 Enable (DE0)—When this bit is set to 1, the SDEN0 pin is held High. When DE0 is set to 0, the SDEN0 pin is Low.

11.2.3 Synchronous Serial Transmit 1 Register (SSD1, Offset 14h) Synchronous Serial Transmit 0 Register (SSD0, Offset 16h)

The Synchronous Serial Transmit 1 and 0 registers contain data to be transferred from the processor to the peripheral on a write operation. Only the least-significant 8 bits of the register are used. The format of SSD1 and SSD0 is shown in Figure 11-3.

Writes to SSD1 or SSD0 cause the PB bit in the SSS register to be set and a transmission sequence to begin as shown in Figure 11-5 on page 11-8. A write to either SSD1 or SSD0 while the port is busy sets the RE/TE (Receive/Transmit Error) bit in the SSS register and does not generate additional data transfers.

Figure 11-3 Synchronous Serial Transmit Register (SSD1, SSD0, offsets 14h and 16h)



The value of these registers at reset is undefined.

Bits 15–8: Reserved—Set to 0.

Bits 7–0: Send Data (SD)—Data to transmit over the SDATA pin. Bit 0 is transmitted first, bit 7 is transmitted last.

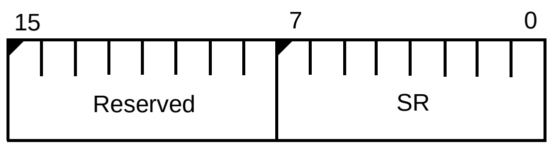
11.2.4 Synchronous Serial Receive Register (SSR, Offset 18h)

The Synchronous Serial Receive (SSR) register contains the data transferred from the peripheral to the processor on a read operation. Only the least-significant 8 bits of the register are used. The format of the SSR register is shown in Figure 11-4.

A receive data transmission is initiated by reading the SSR register while the port is not busy (PB bit in SSS register is 0) and one or both of the enable bits (DE1–DE0 in the SSC register) is set. A receive transmission is not initiated by reading the SSR register when neither of the enable bits is set (DE1–DE0 = 00b). This allows the software to read the received data without initiating another receive transmission.

A read of the Synchronous Serial Receive register while the port is busy (PB bit is set in the SSS register) sets the RE/TE (Receive/Transmit Error) bit in the SSS register and returns an indeterminate value. Such a read does not generate additional data transfers.

Figure 11-4 Synchronous Serial Receive Register (SSR, offset 18h)



The value of this register at reset is undefined.

Bits 15–8: Reserved—Set to 0.

Bits 7–0: Receive Data (SR)—Data received over the SDATA pin. Bit 0 is transmitted first, bit 7 is transmitted last.

11.3 SSI PROGRAMMING

The SSI interface allows for a variety of software and hardware protocols.

- **Signaling a read/write**—In general, software uses the first write to the SSI to transmit an address or count to the peripheral. This value can include a read/write flag in the case where the device supports both reads and writes.
- **Using SSD1 as an address register**—The SSD1 register can be an address register that holds the value of the last address accessed, and the SSD0 register can be the data transmit register. In this case, the current value in the SSD1 register can be used by software to generate the next address or to determine if the last transaction was a read or a write.
- **Using SSD1 and SSD0 as transmit registers for two peripheral devices**—In some systems, it may clarify the code and aid in debugging to view the two data transmit registers as unique to different peripheral devices. This allows the last value transmitted to each device to be examined by debug code.

Figure 11-5 Synchronous Serial Interface Multiple Write

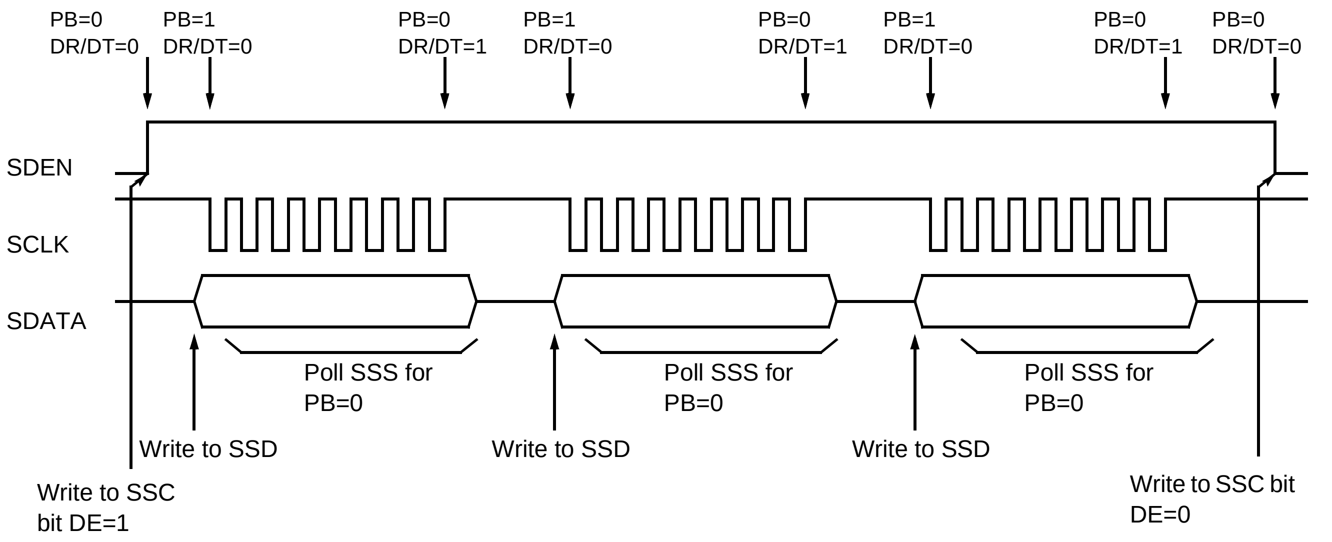
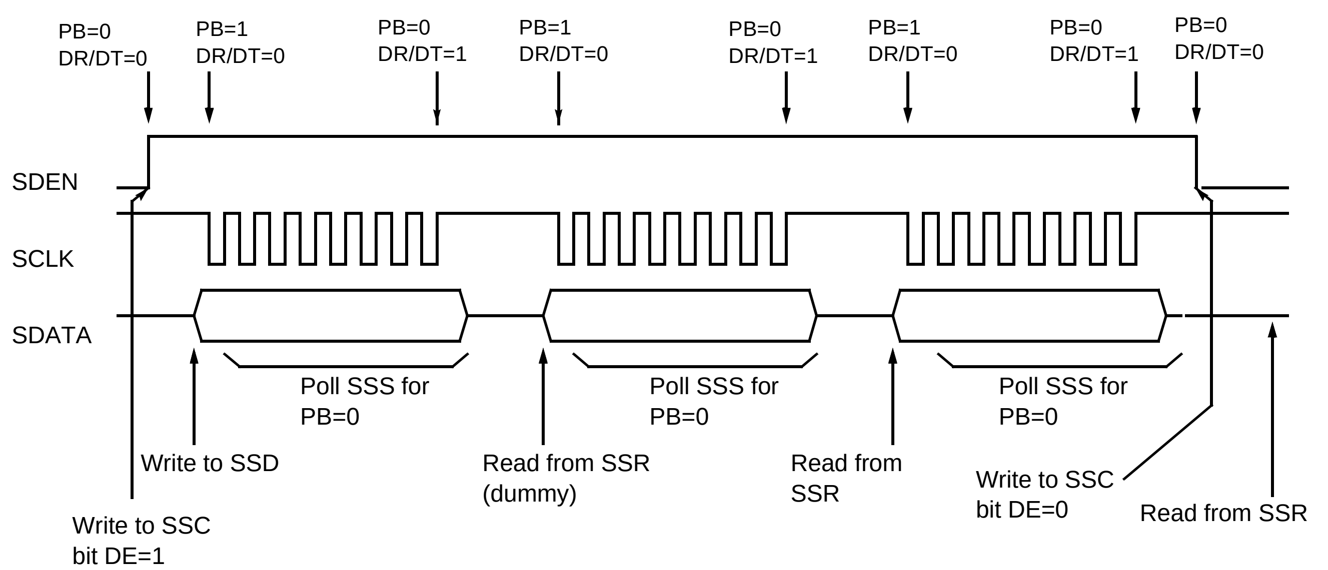


Figure 11-6 Synchronous Serial Interface Multiple Read



CHAPTER

12 PROGRAMMABLE I/O PINS

12.1 OVERVIEW

Thirty-two pins on the Am186EM and Am188EM microcontrollers are available as user-programmable I/O signals (PIOs). Each of these pins can be used as a PIO if the normal function of the pin is not needed. If a pin is enabled to function as a PIO signal, the normal function is disabled and does not affect the pin. A PIO signal can be configured to operate as an input or output with or without internal pullup or pulldown resistors, or as an open-drain output.

After power-on reset, the PIO pins default to various configurations. The column titled *Power-On Reset State* in Table 12-1 lists the defaults for the PIOs. The system initialization code must reconfigure PIOs as required.

The A19–A17 address pins default to normal operation on power-on reset, allowing the processor to correctly begin fetching instructions at the boot address FFFF0h. The DT/$\overline{\text{R}}$, $\overline{\text{DEN}}$, and SRDY pins also default to normal operation on power-on reset.

Figure 12-1 Programmable I/O Pin Operation

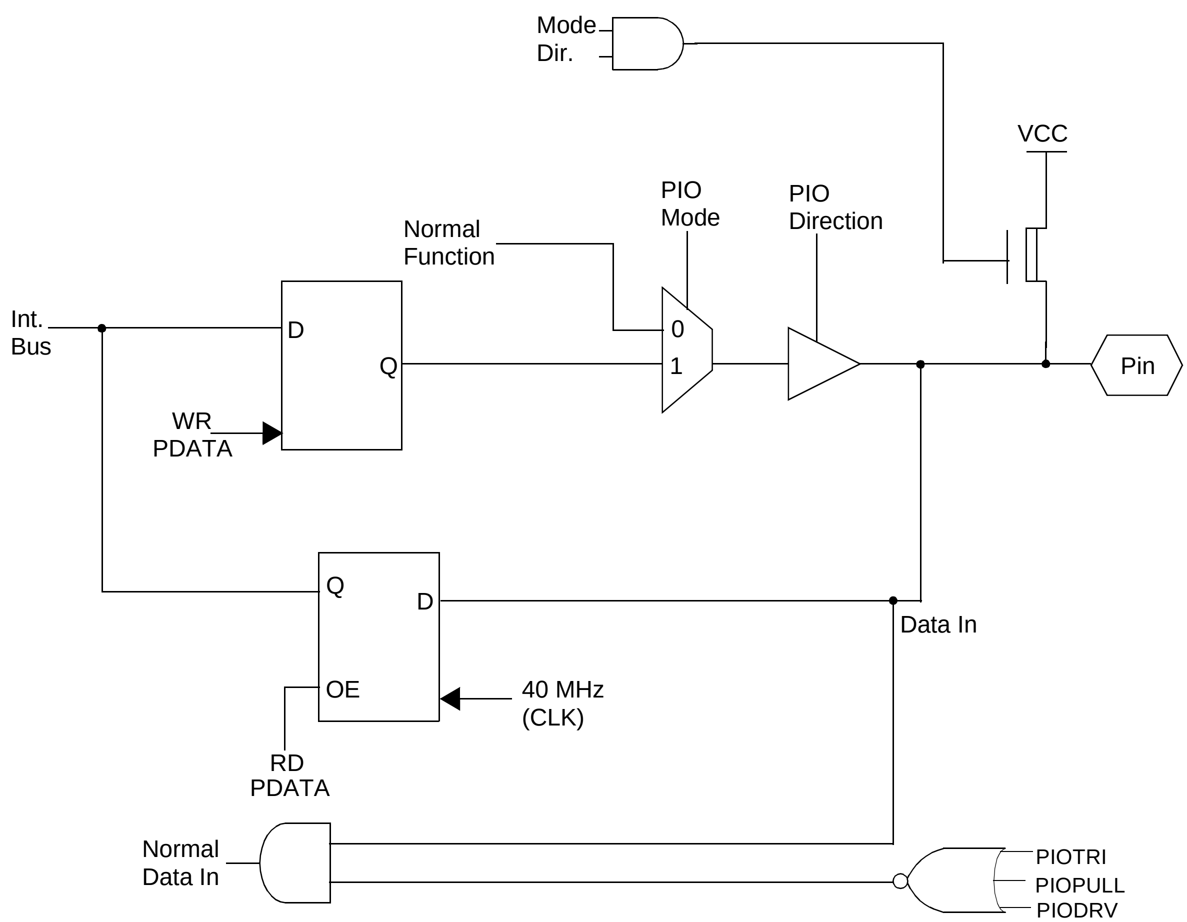


Table 12-1 PIO Pin Assignments

PIO No.	Associated Pin	Power-On Reset Status
0	TMRIN1	Input with pullup
1	TMROUT1	Input with pulldown
2	$\overline{PCS}$6/A2	Input with pullup
3	$\overline{PCS}$5/A1	Input with pullup
4	DT/$\overline{R}$	Normal operation[3]
5	DEN	Normal operation[3]
6	SRDY	Normal operation[4]
7[1]	A17	Normal operation[3]
8[1]	A18	Normal operation[3]
9[1]	A19	Normal operation[3]
10	TMROUT0	Input with pulldown
11	TMRIN0	Input with pullup
12	DRQ0	Input with pullup
13	DRQ1	Input with pullup
14	$\overline{MCS}$0	Input with pullup
15	$\overline{MCS}$1	Input with pullup
16	$\overline{PCS}$0	Input with pullup
17	$\overline{PCS}$1	Input with pullup
18	$\overline{PCS}$2	Input with pullup
19	$\overline{PCS}$3	Input with pullup
20	SCLK	Input with pullup
21	SDATA	Input with pullup
22	SDEN0	Input with pulldown
23	SDEN1	Input with pulldown
24	$\overline{MCS}$2	Input with pullup
25	$\overline{MCS3}$/$\overline{RFSH}$	Input with pullup
26[1,2]	$\overline{UZI}$	Input with pullup
27	TXD	Input with pullup
28	RXD	Input with pullup
29[1,2]	S6/$\overline{CLKDIV}$2	Input with pullup
30	INT4	Input with pullup
31	INT2	Input with pullup

Notes:

1. *These pins are used by emulators. (Emulators also use* $\overline{S}$2–$\overline{S}$0, $\overline{RES}$, *NMI, CLKOUTA,* $\overline{BHE}$, *ALE, AD15–AD0, and A16–A0.)*
2. *These pins revert to normal operation if* $\overline{BHE}$/$\overline{ADEN}$ *(Am186EM) or* $\overline{RFSH}$2/$\overline{ADEN}$ *(Am188EM) is held Low during power-on reset.*
3. *When used as a PIO, input with pullup option available.*
4. *When used as a PIO, input with pulldown option available.*

12.2 PIO MODE REGISTERS

Table 12-2 shows the possible settings for the PIO Mode and PIO Direction bits. The Am186EM and Am188EM microcontrollers default the 32 PIO pins to either 00b (normal operation) or 01b (PIO input with weak internal pullup or pulldown enabled).

Pins that default to active High outputs at reset are pulled down. All other pins are pulled up or are normal operation. See Table 12-2. The column titled *Power-On Reset State* in Table 12-1 lists the defaults for the PIOs.

The internal pullup resistor has a value of approximately 10 Kohms. The internal pulldown resistor has a value of approximately 10 Kohms.

Table 12-2 PIO Mode and PIO Direction Settings

PIO Mode	PIO Direction	Pin Function
0	0	Normal operation
0	1	PIO input with pullup/pulldown
1	0	PIO output
1	1	PIO input without pullup/pulldown

Figure 12-2 PIO Mode 1 Register (PIOMODE1, offset 76h)

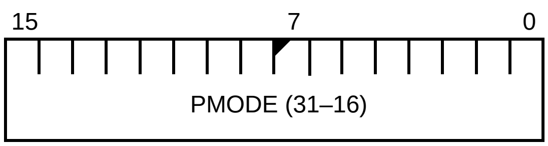
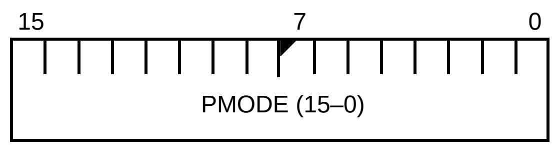


Figure 12-3 PIO Mode 0 Register (PIOMODE0, offset 70h)



12.2.1 PIO Mode 1 Register (PIOMODE1, Offset 76h)

The value of PIOMODE1 at reset is 0000h.

Bits 15–0: PIO Mode Bits (PMODE31–PMODE16)—This field with the PIO direction registers determines whether each PIO pin performs its pre-assigned function or is enabled as a custom PIO signal. The most significant bit of the PMODE field determines whether PIO31 is enabled, the next bit determines whether PIO30 is enabled, and so on. Table 12-2 shows the values that the PIO mode bits and the PIO direction bits can encode.

12.2.2 PIO Mode 0 Register (PIOMODE0, Offset 70h)

The value of PIOMODE0 at reset is 0000h.

Bits 15–0: PIO Mode Bits (PMODE15–PMODE0)—This field is a continuation of the PMODE field in the PIO Mode 1 register.

12.3 PIO DIRECTION REGISTERS

Each PIO is individually programmed as an input or output by a bit in one of the PIO Direction registers (see Figure 12-4 and Figure 12-5). Table 12-2 on page 12-3 shows the values that the PIO mode bits and the PIO direction bits can encode. The column titled *Power-On Reset State* in Table 12-1 lists the reset default values for the PIOs. Bits in the PIO Direction registers have the same correspondence to pins as bits in the PIO Mode registers.

Figure 12-4 PIO Direction 1 Register (PDIR1, offset 78h)

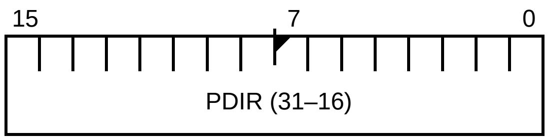
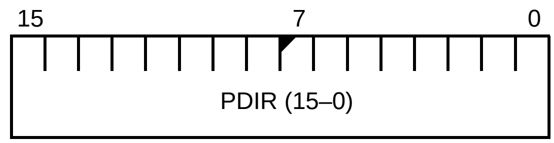


Figure 12-5 PIO Direction 0 Register (PDIR0, offset 72h)



12.3.1 PIO Direction 1 Register (PDIR1, Offset 78h)

The value of PDIR1 at reset is FFFFh.

Bits 15–0: PIO Direction Bits (PDIR31–PDIR16)—This field determines whether each PIO pin acts as an input or an output. The most significant bit of the PDIR field determines the direction of PIO31, the next bit determines the direction of PIO30, and so on. A 1 in the bit configures the PIO signal as an input, and a 0 in the bit configures it as an output or as normal pin function.

12.3.2 PIO Direction 0 Register (PDIR0, Offset 72h)

The value of PDIR0 at reset is FC0Fh.

Bits 15–0: PIO Direction Bits (PDIR15–PDIR0)—This field is a continuation of the PDIR field in the PIO Direction 1 register.

12.4 PIO DATA REGISTERS

If a PIO pin is enabled as an output, the value in the corresponding bit in one of the PIO Data registers (see Figure 12-6 and Figure 12-7) is driven on the pin with no inversion (Low=0, High=1). If a PIO pin is enabled as an input, the value on the PIO pin is reflected in the value of the corresponding bit in the PIO Data register, with no inversion. Bits in the PIO Data registers have the same correspondence to pins as bits in the PIO Mode registers and PIO Direction registers.

Figure 12-6 PIO Data 1 Register (PDATA1, offset 7Ah)

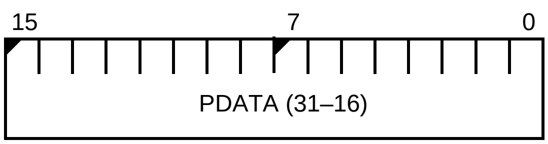
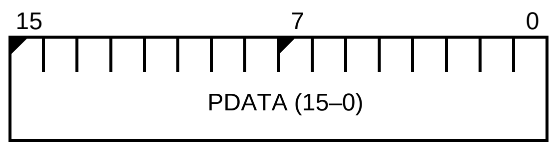


Figure 12-7 PIO Data 0 Register (PDATA0, offset 74h)



12.4.1 PIO Data Register 1 (PDATA1, Offset 7Ah)

Bits 7–0: PIO Data Bits (PDATA31–PDATA16)—This field determines the level driven on each PIO pin or reflects the external level of the pin, depending upon whether the pin is configured as an output or an input in the PIO Direction registers. The most significant bit of the PDATA field indicates the level of PIO31, the next bit indicates the level of PIO30, and so on.

The value of PDATA1 at reset is undefined.

12.4.2 PIO Data Register 0 (PDATA0, Offset 74h)

Bits 15–0: PIO Data Bits (PDATA15–PDATA0)—This field is a continuation of the PDATA field in the PIO Data 1 register.

The value of PDATA0 at reset is undefined.

12.5 OPEN-DRAIN OUTPUTS

The PIO Data registers permit the PIO signals to be operated as open-drain outputs. This is accomplished by keeping the appropriate PDATA bits constant in the PIO Data register and writing the data value into its associated bit position in the PIO Direction register, so the output is either driving Low or is disabled, depending on the data.

APPENDIX

A REGISTER SUMMARY

This appendix summarizes the peripheral control block registers. Table A-1 lists all the registers. Figure A-1 shows the layout of each of the internal registers.

The column titled *Comment* in Table A-1 is used to identify the specific use of interrupt registers when there is a mix of master mode and slave mode usage. The registers that are marked as *Slave & master* can have different configurations for the different modes.

Table A-1 Internal Register Summary

Hex Offset	Mnemonic	Register Description	Comment
FE	RELREG	Peripheral control block relocation register	
F6	RESCON	Reset configuration register	
F4	PRL	Processor release level register	
F0	PDCON	Power-save control register	
E4	EDRAM	Enable RCU register	
E2	CDRAM	Clock prescaler register	
E0	MDRAM	Memory partition register	
D8	D1TC	DMA 1 transfer count register	
D6	D1DSTH	DMA 1 destination address high register	
D4	D1DSTL	DMA 1 destination address low register	
D2	D1SRCH	DMA 1 source address high register	
D0	D1SRCL	DMA 1 source address low register	
CA	D0CON	DMA 0 control register	
C8	D0TC	DMA 0 transfer count register	
C6	D0DSTH	DMA 0 destination address high register	
C4	D0DSTL	DMA 0 destination address low register	
C2	D0SRCH	DMA 0 source address high register	
C0	D0SRCL	DMA 0 source address low register	
A8	MPCS	$\overline{\text{PCS}}$ and $\overline{\text{MCS}}$ auxiliary register	
A6	MMCS	Midrange memory chip select register	
A4	PACS	Peripheral chip select register	
A2	LMCS	Low memory chip select register	
A0	UMCS	Upper memory chip select register	
88	SPBAUD	Serial port baud rate divisor register	
86	SPRD	Serial port receive data register	
84	SPTD	Serial port transmit data register	
82	SPSTS	Serial port status register	
80	SPCT	Serial port control register	
7A	PDATA1	PIO data 1 register	
78	PDIR1	PIO direction 1 register	
76	PIOMODE1	PIO mode 1 register	
74	PDATA0	PIO data 0 register	
72	PDIR0	PIO direction 0 register	
70	PIOMODE0	PIO mode 0 register	
66	T2CON	Timer 2 mode/control register	
62	T2CMPA	Timer 2 maxcount compare A register	
60	T2CNT	Timer 2 count register	
5E	T1CON	Timer 1 mode/control register	

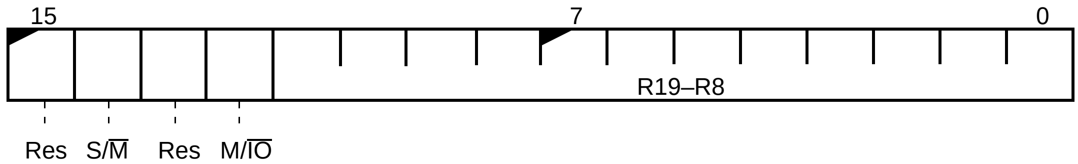
Table A-1 Internal Register Summary (continued)

Hex Offset	Mnemonic	Register Description	Comment
5C	T1CMPB	Timer 1 maxcount compare B register	
5A	T1CMPA	Timer 1 maxcount compare A register	
58	T1CNT	Timer 1 count register	
56	T0CON	Timer 0 mode/control register	
54	T0CMPB	Timer 0 maxcount compare B register	
52	T0CMPA	Timer 0 maxcount compare A register	
50	T0CNT	Timer 0 count register	
44	SPICON	Serial port interrupt control register	Master mode
42	WDCON	Watchdog timer interrupt control register	Master mode
40	I4CON	INT4 control register	Master mode
3E	I3CON	INT3 control register	Master mode
3C	I2CON	INT2 control register	Master mode
3A	I1CON	INT1 control register	Master mode
	T2INTCON	Timer 2 interrupt control register	Slave mode
38	I0CON	INT0 control register	Master mode
	T1INTCON	Timer 1 interrupt control register	Slave mode
36	DMA1CON	DMA 1 interrupt control register	Slave & master
34	DMA0CON	DMA 0 interrupt control register	Slave & master
32	TCUCON	Timer interrupt control register	Master mode
	T0INTCON	Timer 0 interrupt control register	Slave mode
30	INTSTS	Interrupt status register	Slave & master
2E	REQST	Interrupt request register	Slave & master
2C	INSERV	In-service register	Slave & master
2A	PRIMSK	Priority mask register	Slave & master
28	IMASK	Interrupt mask register	Slave & master
26	POLLST	Poll status register	Master mode
24	POLL	Poll register	Master mode
22	EOI	End-of-interrupt register	Master mode
	EOI	Specific end-of-interrupt register	Slave mode
20	INTVEC	Interrupt vector register	Slave mode
18	SSR	Synchronous serial receive register	
16	SSD0	Synchronous serial transmit 0 register	
14	SSD1	Synchronous serial transmit 1 register	
12	SSC	Synchronous serial control register	
10	SSS	Synchronous serial status register	

Figure A-1 Internal Register Summary

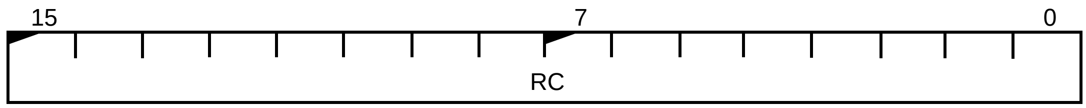
Offset (Hexadecimal)

FE



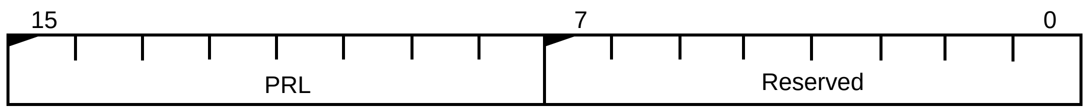
Peripheral Control Block Relocation Register (RELREG)
Page 4-4

F6



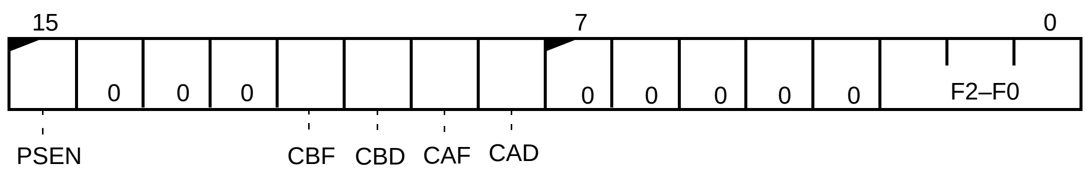
Reset Configuration Register (RESCON)
Page 4-5

F4



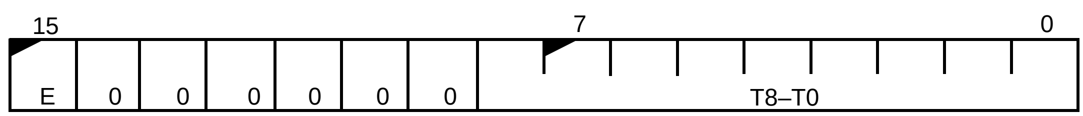
Processor Release Level Register (PRL)
Page 4-6

F0



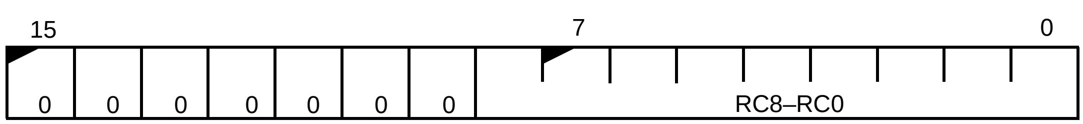
Power-Save Control Register (PDCON)
Page 4-7

E4



Enable RCU Register (EDRAM)
Page 6-2

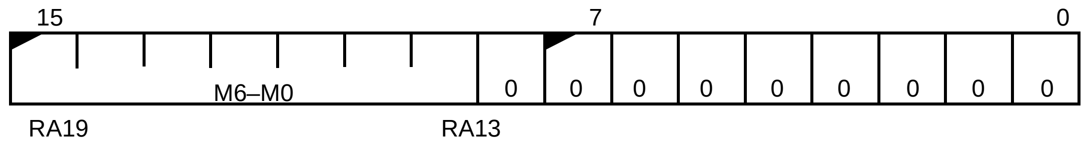
E2



Clock Prescaler Register (CDRAM)
Page 6-2

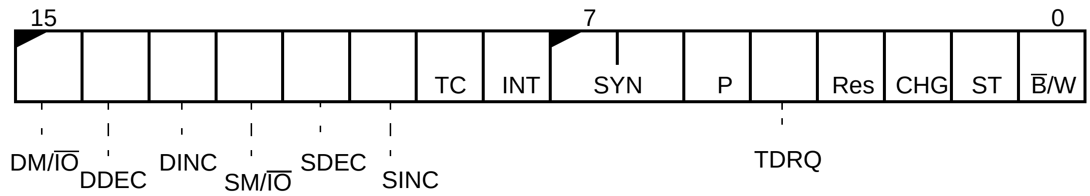
Figure A-1 Internal Register Summary (continued)

E0



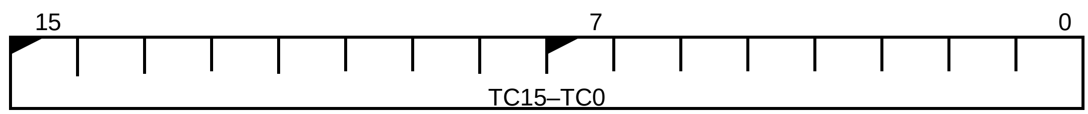
Memory Partition Register (MDRAM)
Page 6-1

CA



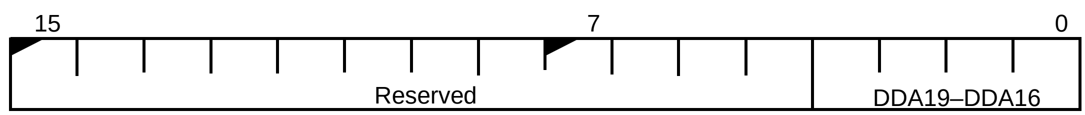
DMA 1 Control Register (D1CON)
Page 9-3

D8



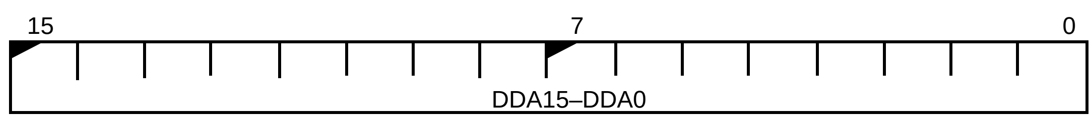
DMA 1 Transfer Count Register (D1TC)
Page 9-5

D6



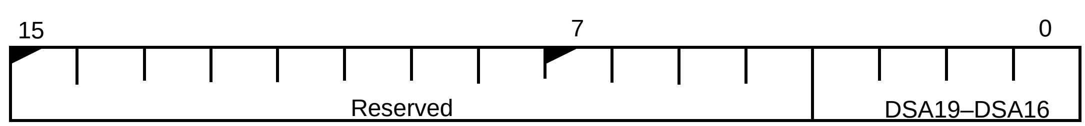
DMA 1 Destination Address High Register (D1DSTH)
Page 9-6

D4



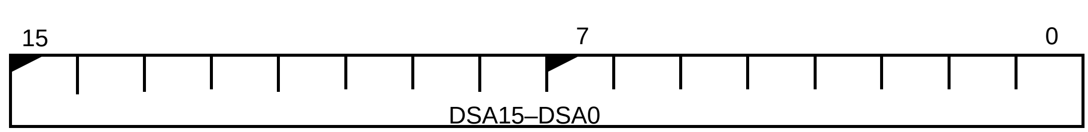
DMA 1 Destination Address Low Register (D1DSTL)
Page 9-7

D2



DMA 1 Source Address High Register (D1SRCH)
Page 9-8

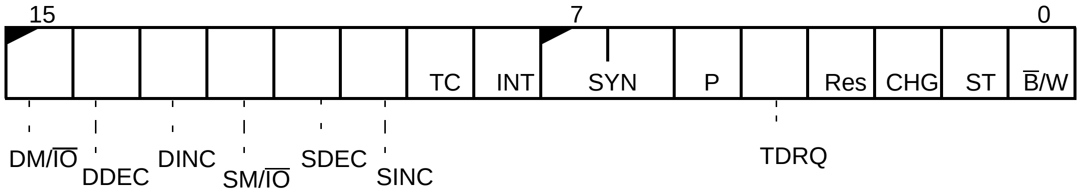
D0



DMA 1 Source Address Low Register (D1SRCL)
Page 9-9

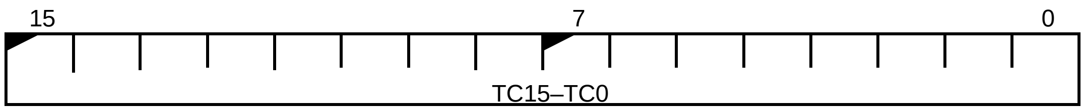
Figure A-1 Internal Register Summary (continued)

CA



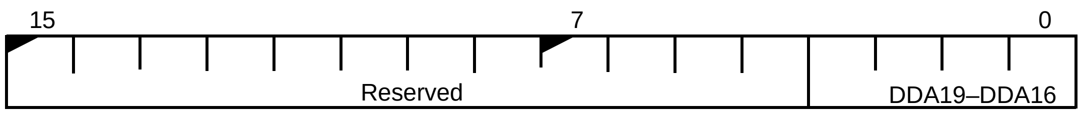
DMA 0 Control Register (D0CON)
Page 9-3

C8



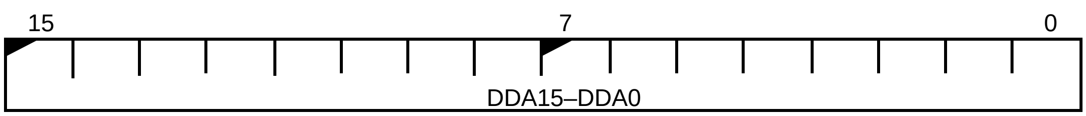
DMA 0 Transfer Count Register (D0TC)
Page 9-5

C6



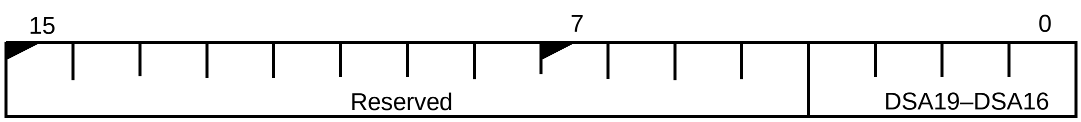
DMA 0 Destination Address High Register (D0DSTH)
Page 9-6

C4



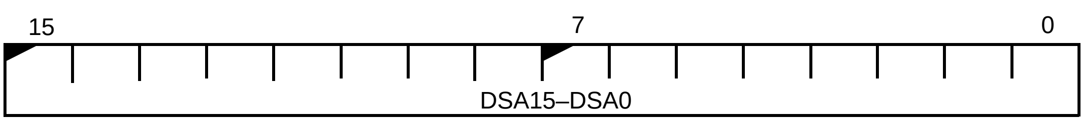
DMA 0 Destination Address Low Register (D0DSTL)
Page 9-7

C2



DMA 0 Source Address High Register (D0SRCH)
Page 9-8

C0



DMA 0 Source Address Low Register (D0SRCL)
Page 9-9

Figure A-1 Internal Register Summary (continued)

A8

15	14–8	7	6	5	4	3	2	1–0
1	M6–M0	EX	MS	1	1	1	R2	R1–R0

$\overline{PCS}$ and $\overline{MCS}$ Auxiliary Register (MPCS)
Page 5-10

A6

15–9	8	7	6	5	4	3	2	1–0
BA19–BA13	1	1	1	1	1	1	R2	R1–R0

Midrange Memory Chip Select Register (MMCS)
Page 5-8

A4

15–7	6	5	4	3	2	1–0
BA19–BA11	1	1	1	R3	R2	R1–R0

Peripheral Chip Select Register (PACS)
Page 5-12

A2

15	14–12	11	10	9	8	7	6	5	4	3	2	1–0
0 (A19)	UB2–UB0	1	1	1	1	R7	PSE	1	1	1	R2	R1–R0

Low Memory Chip Select Register (LMCS)
Page 5-6

A0

15	14–12	11	10	9	8	7	6	5	4	3	2	1–0
1 (A19)	LB2–LB0	0	0	0	0	R7	0	1	1	1	R2	R1–R0

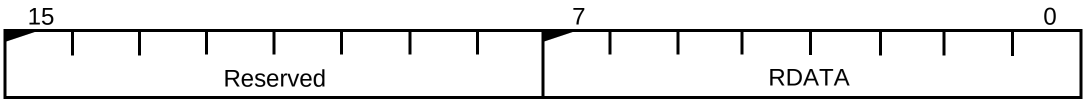
Upper Memory Chip Select Register (UMCS)
Page 5-4

88

15–0
BAUDDIV

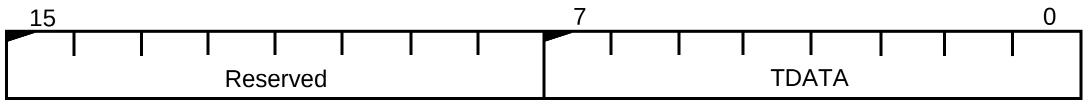
Serial Port Baud Rate Divisor Register (SPBAUD)
Page 10-7

Figure A-1 Internal Register Summary (continued)



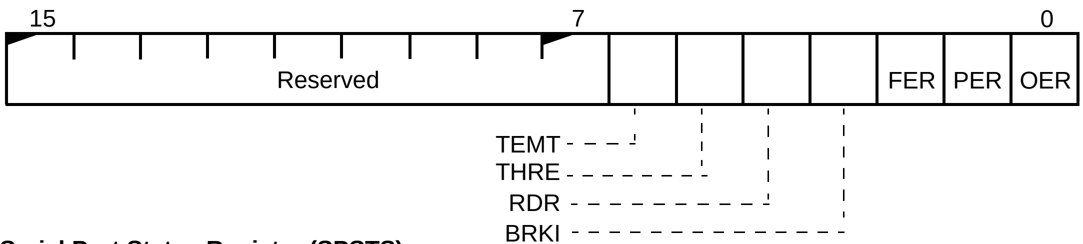
86

Serial Port Receive Data Register (SPRD)
Page 10-6



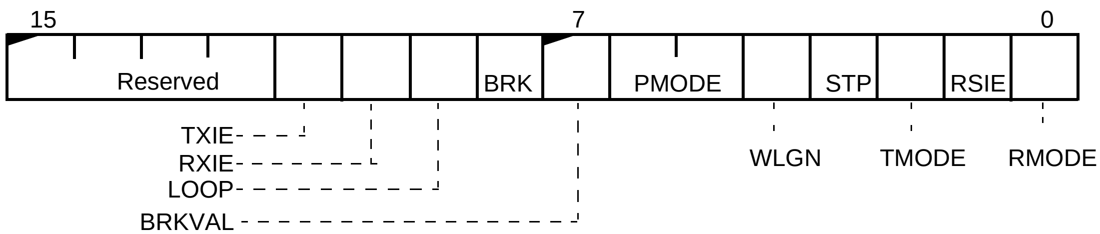
84

Serial Port Transmit Data Register (SPTD)
Page 10-5



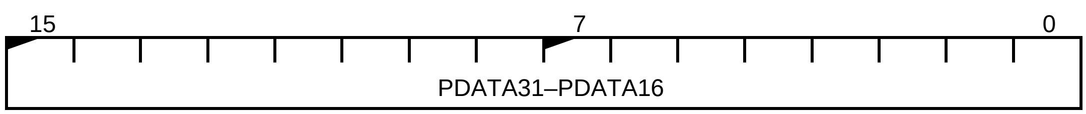
82

Serial Port Status Register (SPSTS)
Page 10-4



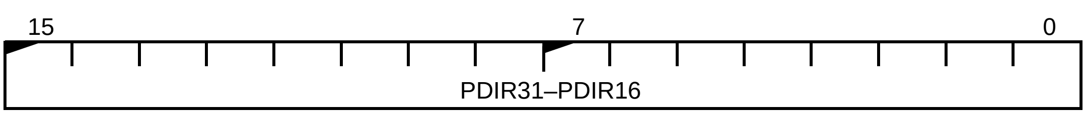
80

Serial Port Control Register (SPCT)
Page 10-2



7A

PIO Data 1 Register (PDATA1)
Page 12-5



78

PIO Direction 1 Register (PDIR1)
Page 12-4

Figure A-1 Internal Register Summary (continued)

76

15								7							0
PMODE31–PMODE16															

PIO Mode 1 Register (PIOMODE1)
Page 12-3

74

15								7							0
PDATA15–PDATA0															

PIO Data 0 Register (PDATA0)
Page 12-5

72

15								7							0
PDIR15–PDIR0															

PIO Direction 0 Register (PDIR0)
Page 12-4

70

15								7							0
PMODE15–PMODE0															

PIO Mode 0 Register (PIOMODE0)
Page 12-3

66

15								7							0
EN	$\overline{INH}$	INT	0	0	0	0	0	0	0	MC	0	0	0	0	CONT

Timer 2 Mode/Control Register (T2CON)
Page 8-5

62

15								7							0
TC15–TC0															

Timer 2 Maxcount Compare A Register (T2CMPA)
Page 8-7

Figure A-1 Internal Register Summary (continued)

60

15–0
TC15–TC0

Timer 2 Count Register (T2CNT)
Page 8-6

5E

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EN	$\overline{INH}$	INT	RIU	0	0	0	0	0	0	MC	RTG	P	EXT	ALT	CONT

Timer 1 Mode/Control Register (T1CON)
Page 8-3

5C

15–0
TC15–TC0

Timer 1 Maxcount Compare B Register (T1CMPB)
Page 8-7

5A

15–0
TC15–TC0

Timer 1 Maxcount Compare A Register (T1CMPA)
Page 8-7

58

15–0
TC15–TC0

Timer 1 Count Register (T1CNT)
Page 8-6

56

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EN	$\overline{INH}$	INT	RIU	0	0	0	0	0	0	MC	RTG	P	EXT	ALT	CONT

Timer 0 Mode/Control Register (T0CON)
Page 8-3

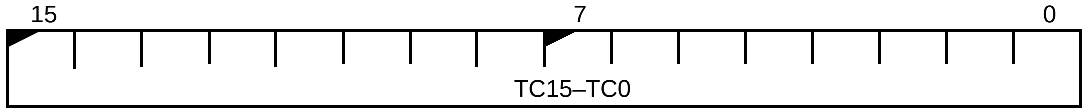
54

15–0
TC15–TC0

Timer 0 Maxcount Compare B Register (T0CMPB)
Page 8-7

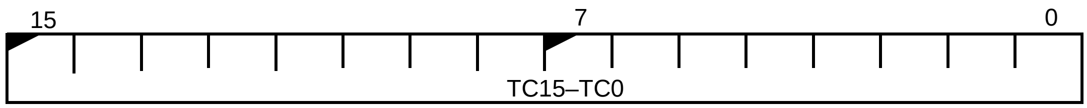
Figure A-1 Internal Register Summary (continued)

52



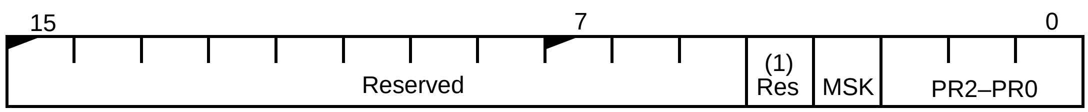
Timer 0 Maxcount Compare A Register (T0CMPA)
Page 8-7

50



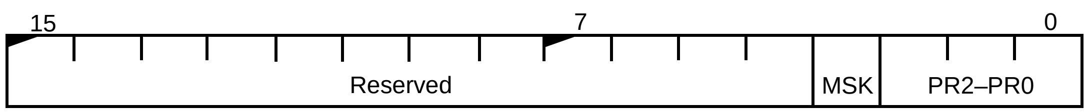
Timer 0 Count Register (T0CNT)
Page 8-6

44



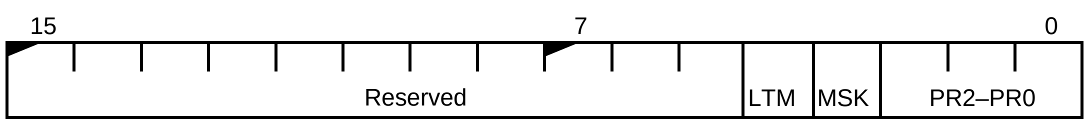
Serial Port Interrupt Control Register (SPICON)
Master Mode
Page 7-19

42



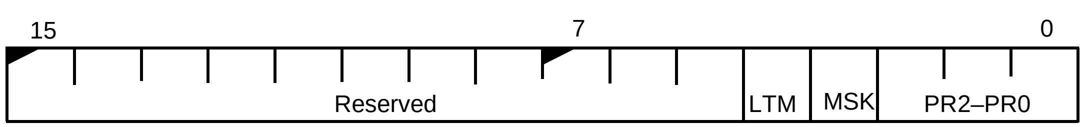
Watchdog Timer Interrupt Control Register (WDCON)
Master Mode
Page 7-18

40



INT4 Control Register (I4CON)
Master Mode
Page 7-15

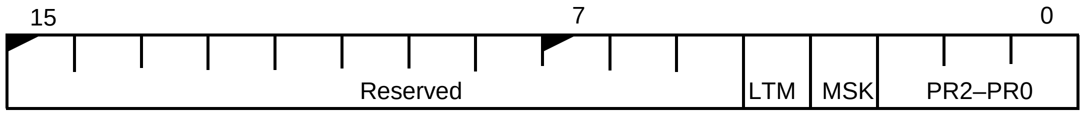
3E



INT3 Control Register (I3CON)
Master Mode
Page 7-15

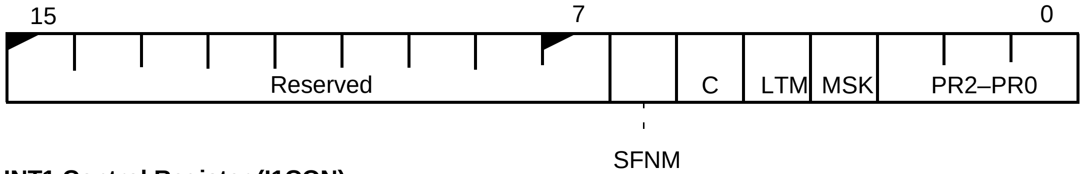
Figure A-1 Internal Register Summary (continued)

3C



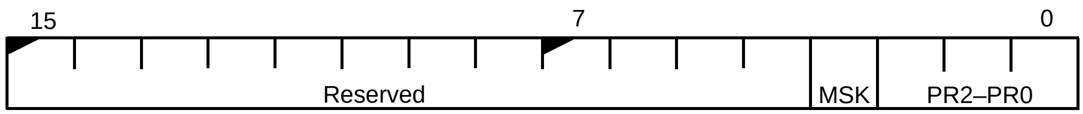
INT2 Control Register (I2CON)
Master Mode
Page 7-15

3A



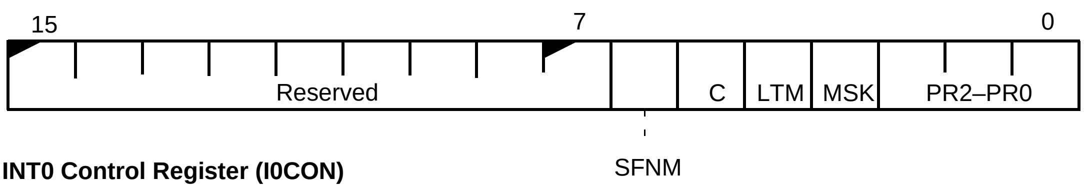
INT1 Control Register (I1CON)
Master Mode
Page 7-13

3A



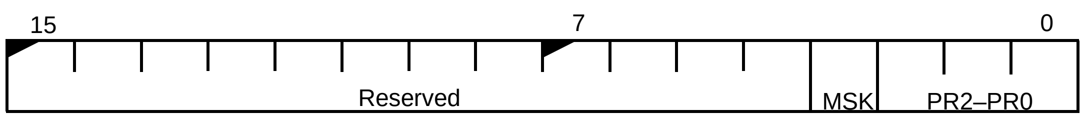
Timer 2 Interrupt Control Register (T2INTCON)
Slave Mode
Page 7-29

38



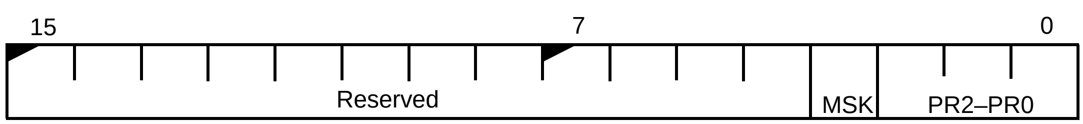
INT0 Control Register (I0CON)
Master Mode
Page 7-13

38



Timer 1 Interrupt Control Register (T1INTCON)
Slave Mode
Page 7-29

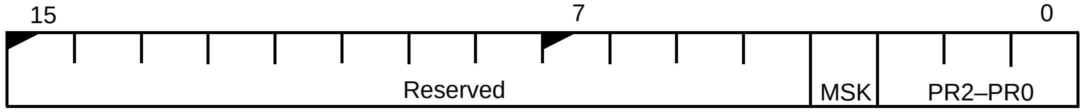
36



DMA 1 Interrupt Control Register (DMA1CON)
Master Mode—Page 7-17
Slave Mode—Page 7-29

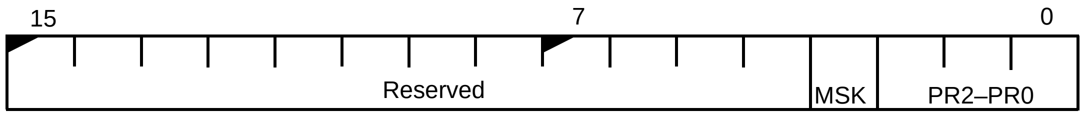
Figure A-1 Internal Register Summary (continued)

34



DMA 0 Interrupt Control Register (DMA0CON)
Master Mode—Page 7-17
Slave Mode—Page 7-29

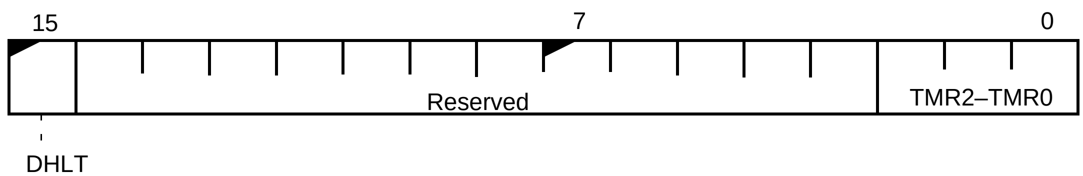
32



Timer Interrupt Control Register (TCUCON)
Master Mode—Page 7-17

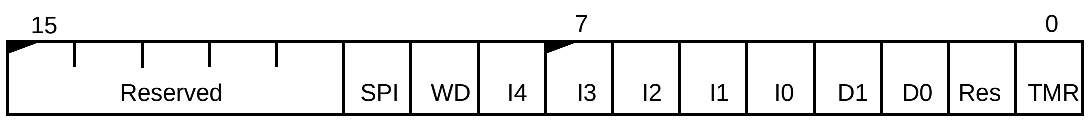
Timer 0 Interrupt Control Register (T0INTCON)
Slave Mode—Page 7-29

30



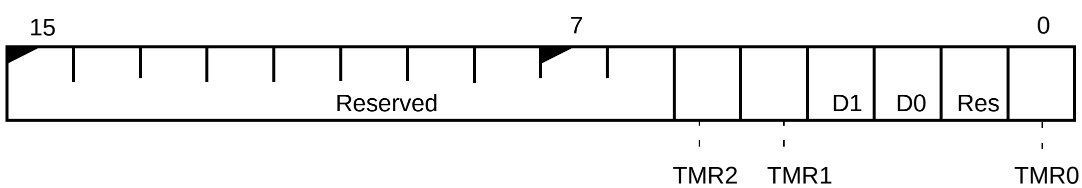
Interrupt Status Register (INTSTS)
Master Mode—Page 7-20
Slave Mode—Page 7-30

2E



Interrupt Request Register (REQST)
Master Mode
Page 7-21

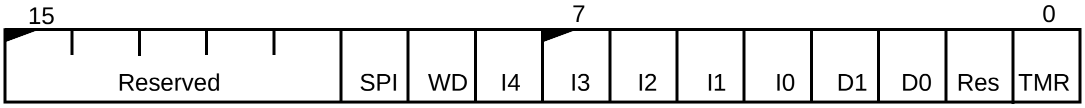
2E



Interrupt Request Register (REQST)
Slave Mode
Page 7-31

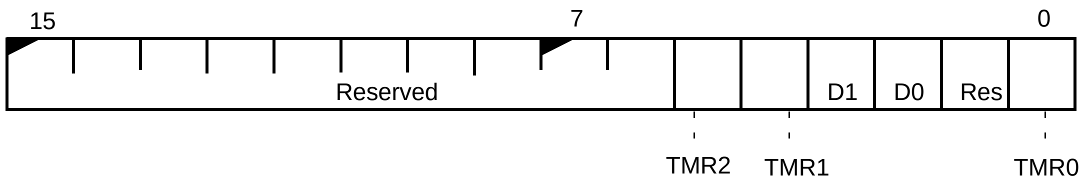
Figure A-1 Internal Register Summary (continued)

2C



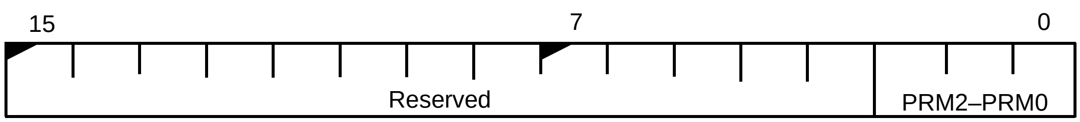
In-Service Register (INSERV)
Master Mode
Page 7-22

2C



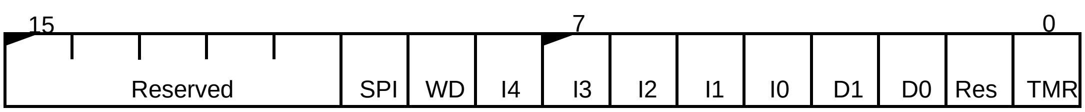
In-Service Register (INSERV)
Slave Mode
Page 7-32

2A



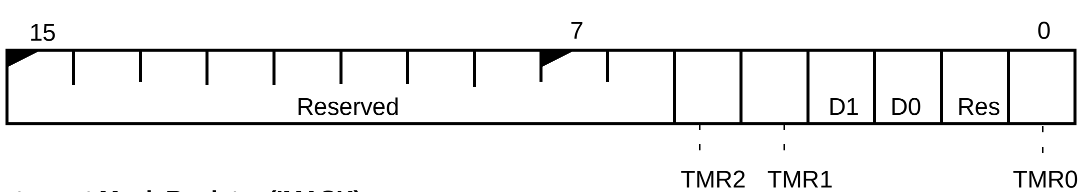
Priority Mask Register (PRIMSK)
Master Mode—Page 7-23
Slave Mode—Page 7-33

28



Interrupt Mask Register (IMASK)
Master Mode
Page 7-24

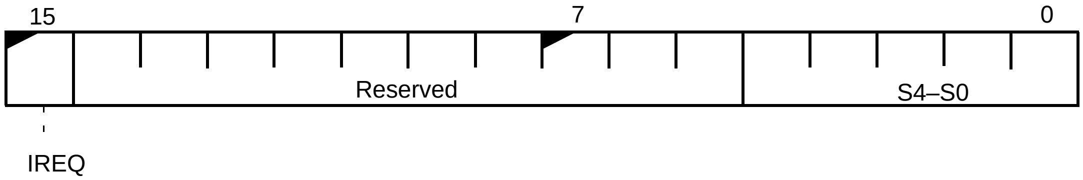
28



Interrupt Mask Register (IMASK)
Slave Mode
Page 7-34

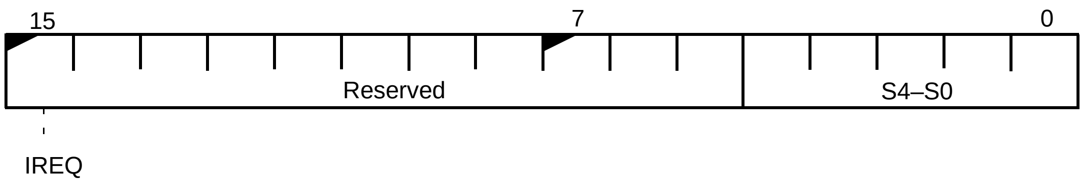
Figure A-1 Internal Register Summary (continued)

26



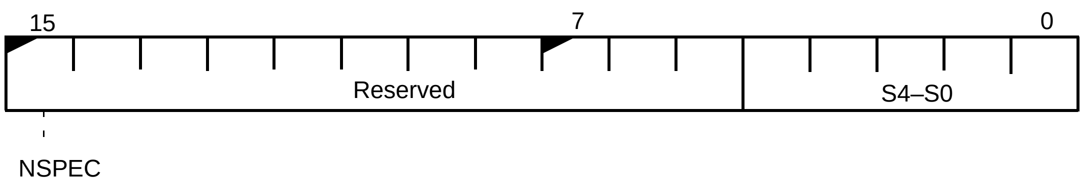
Poll Status Register (POLLST)
Master Mode
Page 7-25

24



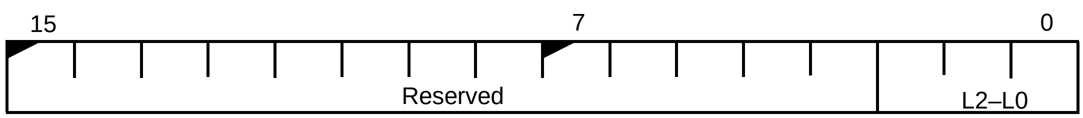
Poll Register (POLL)
Master Mode
Page 7-26

22



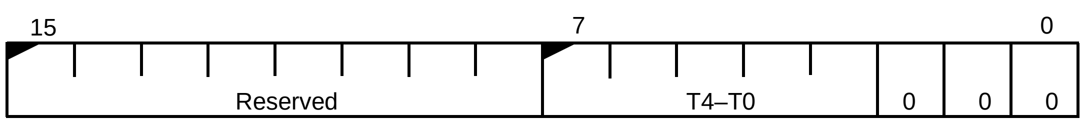
End-of-Interrupt Register (EOI)
Master Mode
Page 7-27

22



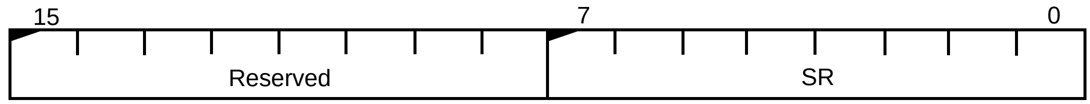
Specific End-of-Interrupt Register (EOI)
Slave Mode
Page 7-35

20

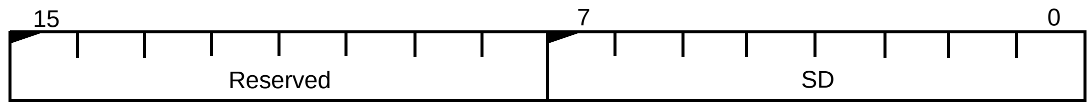


Interrupt Vector Register (INTVEC)
Slave Mode
Page 7-36

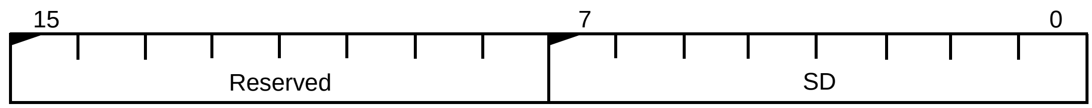
Figure A-1 Internal Register Summary (continued)

18


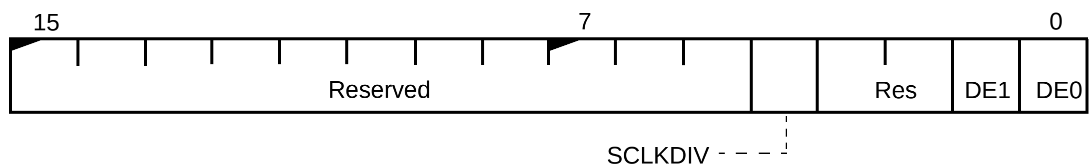
Synchronous Serial Receive Register (SSR)
Page 11-6

16


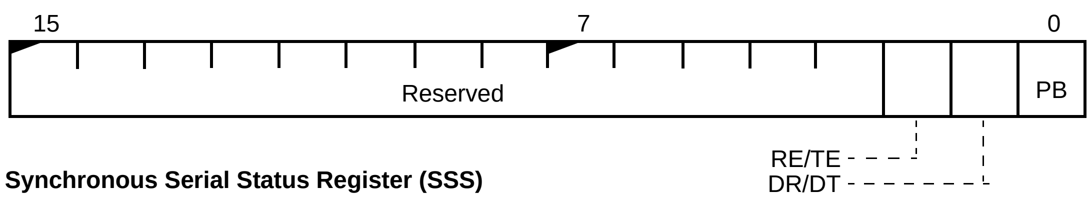
Synchronous Serial Transmit 0 Register (SSD0)
Page 11-5

14


Synchronous Serial Transmit 1 Register (SSD1)
Page 11-5

12


Synchronous Serial Control Register (SSC)
Page 11-4

10


Synchronous Serial Status Register (SSS)
Page 11-3

Symbols

A

B

C

D

E

F

H

I

L

M

N

O

P

R

S

T

U

W