Am186™ED/EDLV 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.

For technical support questions on all E86 products, send E-mail to lpd.support@amd.com.

Corporate Applications Hotline

(800) 222-9323	toll-free for U.S.
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 FusionE86℠ 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.
(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

The Am186™ED/EDLV microcontrollers provide a low-cost, high-performance solution for embedded system designers who wish to use the x86 architecture. By integrating multiple functional blocks with the CPU, the Am186ED/EDLV microcontrollers eliminate the need for off-chip system-interface logic. It is possible to implement a fully functional system with ROM and RAM including DRAM, serial interfaces, and custom I/O capability without additional system-interface logic.

The Am186ED/EDLV microcontrollers can operate at frequencies up to 40 MHz. And, the microcontroller includes an on-board PLL so that the input clock can be one-to-one with the internal processor clock. The Am186ED/EDLV microcontrollers are available in versions operating at 20, 25, 33, and 40 MHz.

The Am186ED/EDLV microcontrollers provide a natural migration path for 80C186/188 designs that need performance and cost enhancements. Because of the integrated DRAM controller, the Am186ED/EDLV microcontrollers provide significant memory cost savings for applications using more than 64K of RAM.

PURPOSE OF THIS MANUAL

This manual describes the technical features and programming interface of the Am186ED/EDLV microcontrollers.

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 Am186ED/EDLV microcontrollers.

USER'S MANUAL OVERVIEW

This manual contains information on the Am186ED/EDLV 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 Am186ED/EDLV microcontrollers.
- Chapter 2 describes the **programmer's model** of the Am186 and Am188 family microcontrollers, including the instruction set 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 **dram control unit**.
- Chapter 7 provides a description of the **interrupt control unit**.
- Chapter 8 provides a description of the **timer control unit**.
- Chapter 9 describes the **DMA controller**.
- Chapter 10 describes the **asynchronous serial ports**.
- Chapter 11 describes the **programmable I/O pins**.
- Appendix A includes a complete summary of **peripheral registers and fields**.

For complete information on the Am186ED/EDLV microcontrollers pin list, timing, thermal characteristics, and physical dimensions, please refer to the *Am186ED/EDLV Microcontrollers Data Sheet*, order # 21336.

AMD DOCUMENTATION

E86™ Microcontroller Family

ORDER NO.	DOCUMENT TITLE
21336A	**Am186ED/EDLV 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.
21076	**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℠ 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.) 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 microcontroller, the Am186ED/EDLV microcontrollers enable designers to reduce the cost, size, and power consumption of embedded systems, while increasing performance and functionality. The Am186ED/EDLV microcontrollers are a cost-effective, enhanced version of the AMD 80C186.

The Am186ED/EDLV microcontrollers are the ideal upgrade for 80C186 designs requiring 80C186-compatibility, increased performance, serial communications, and a glueless bus interface. Developed exclusively for the embedded marketplace, the Am186ED/EDLV microcontrollers increase the performance of existing 80C186 systems while decreasing their cost.

Because the Am186ED/EDLV microcontrollers integrate on-chip peripherals and offer up to twice the performance of an 80C186, 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 Am186ED/EDLV microcontrollers extend the AMD family of microcontrollers based on the industry-standard x86 architecture. The Am186ED/EDLV microcontrollers are a higher-performance, more integrated version of the 80C186 microprocessor, offering an attractive migration path. In addition, the Am186ED/EDLV microcontrollers offer application-specific features that can enhance the system functionality of the Am186ES and Am188ES microcontrollers. Upgrading to the Am186ED/EDLV microcontrollers is an attractive solution for several reasons:

- **Minimized total system cost**—New peripherals and on-chip system interface logic on the Am186ED/EDLV microcontrollers reduce the cost of existing 80C186 designs. Integrated DRAM controller reduces costs for systems using more than 64K of RAM.
- **x86 software compatibility**—80C186-compatible and upward-compatible with the other members of the AMD E86 family. The x86 architecture is the most widely used and supported computer architecture in the world.
- **Enhanced performance**—The Am186ED/EDLV microcontrollers increase the performance of 80C186 systems, and the nonmultiplexed address bus offers faster, unbuffered access to memory.
- **Enhanced functionality**—The new and enhanced on-chip peripherals of the Am186ED/EDLV microcontrollers include a DRAM controller, two asynchronous serial ports, 32 PIOs, a watchdog timer, additional interrupt pins, a pulse width demodulation option, DMA directly to and from the serial ports, 8-bit and 16-bit static bus sizing, a 16-bit reset configuration register, and enhanced chip-select functionality.

The Am186ED/EDLV microcontrollers are part of the AMD E86 family of embedded microcontrollers and microprocessors based on the x86 architecture. The E86 family includes the 80C186, 80C188, 80L186, 80L188, Am186EM, Am188EM, Am186EMLV, Am188™EMLV, Am186ES, Am188ES, Am186ESLV, Am188ESLV, Am186ED, Am186EDLV, Am186ER, Am188ER, Élan™SC300, ÉlanSC310, ÉlanSC400, ÉlanSC400, Am386®SX, Am386DX, and Am486®DX microprocessors and microcontrollers.

1.2 DISTINCTIVE CHARACTERISTICS

A block diagram of the microcontrollers is shown in Figure 1-1. The Am186ED/EDLV microcontrollers use a 16-bit external bus. The Am186ED and the Am186EDLV have the same feature set—the difference between them is operation voltages. For electrical characteristics, refer to *Am186ED/EDLV Microcontrollers Data Sheet*, order # 21336

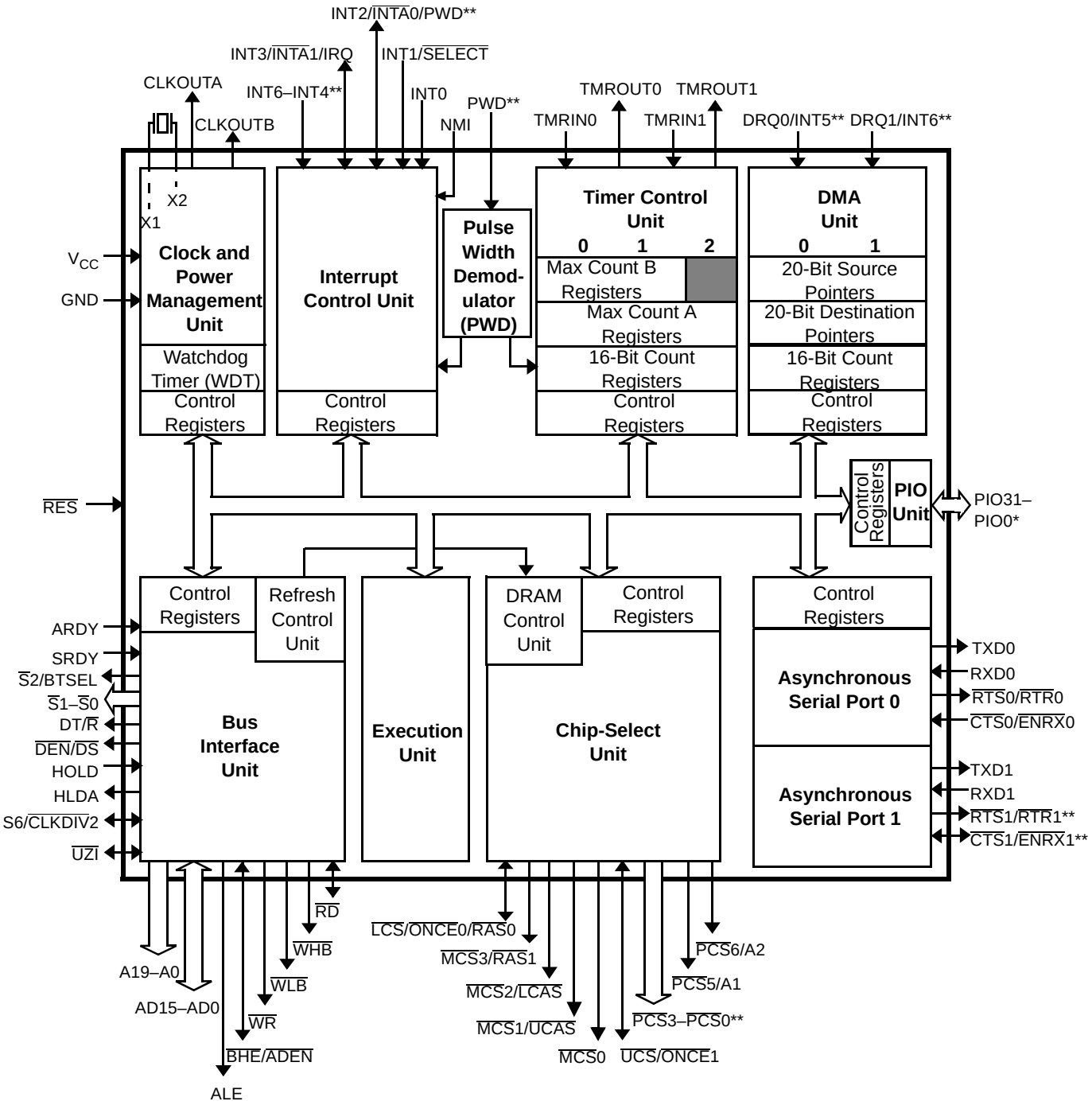
The Am186ED/EDLV microcontrollers include the following features:

- E86™ family 80C186- and 80C188-compatible microcontrollers with enhanced bus interface
  - — Lower system cost with higher performance
  - — 3.3-V ± 0.3-V operation (Am186EDLV microcontrollers)
- High performance
  - — 20-, 25-, 33-, and 40-MHz operating frequencies
  - — Supports zero-wait-state operation at 40 MHz with 70-ns static memory
  - — 1-Mbyte memory address space
  - — 64-Kbyte I/O space
- Enhanced features provide improved memory access and remove the requirement for a 2x clock input
  - — Nonmultiplexed address bus
  - — Processor operates at the clock input frequency
  - — 8-bit or 16-bit memory and I/O static bus option
  - — 8-bit and 16-bit boot mode for $\overline{\text{UCS}}$ accesses
  - — Improved alternate bus mastering
- Integrated DRAM controller
  - — Glueless interface to 50-ns DRAM at 40 MHz for no-wait state operation
  - — Supports 50-ns, 60-ns and 70-ns DRAM
  - — Multiplexed addresses for DRAM row and column accesses
  - — Two $\overline{\text{RAS}}$ signals that supports two banks of DRAM
  - — Two byte $\overline{\text{CAS}}$ signals
  - — Directly supports 4 Mbit (256 Kx16) fast page mode and extended data out mode DRAMs
  - — Prioritized $\overline{\text{PCS}}$ over DRAM space accesses
- Enhanced integrated peripherals provide increased functionality, while reducing system cost
  - — Thirty-two programmable I/O (PIO) pins
  - — Two full-featured asynchronous serial ports allow full-duplex, 7-bit, 8-bit, or 9-bit data transfers
  - — Serial port hardware handshaking with $\overline{\text{CTS}}$, $\overline{\text{RTS}}$, $\overline{\text{ENRX}}$, and $\overline{\text{RTR}}$ selectable for each port
  - — Multidrop 9-bit serial port protocol

- — Independent serial port baud rate generators
- — DMA to and from the serial ports
- — Watchdog timer can generate NMI or reset
- — A pulse width demodulation option
- — A data strobe, true asynchronous bus interface option
- — Reset configuration register
- — 9-bit protocol enhancements to the serial port with Mode 7

- Familiar 80C186 peripherals
  - — Two independent DMA channels
  - — Programmable interrupt controller with up to eight external and eight internal interrupts
  - — Three programmable 16-bit timers
  - — Programmable memory and peripheral chip-select logic
  - — Programmable wait state generator
  - — Power-save clock divider
- Software-compatible with the 80C186 and 80C188 microcontrollers with widely available native development tools, applications, and system software
- A compatible evolution of the Am186™ES and Am188™ES microcontrollers
- Available in the following packages:
  - — 100-pin, thin quad flat pack (TQFP)
  - — 100-pin, plastic quad flat pack (PQFP)

Figure 1-1 Am186ED/EDLV Microcontrollers Block Diagram



Notes:

** All PIO signals are shared with other physical pins. See the pin descriptions in Chapter 3 and Table 3-1 on page 3-15 for information on shared functions.*

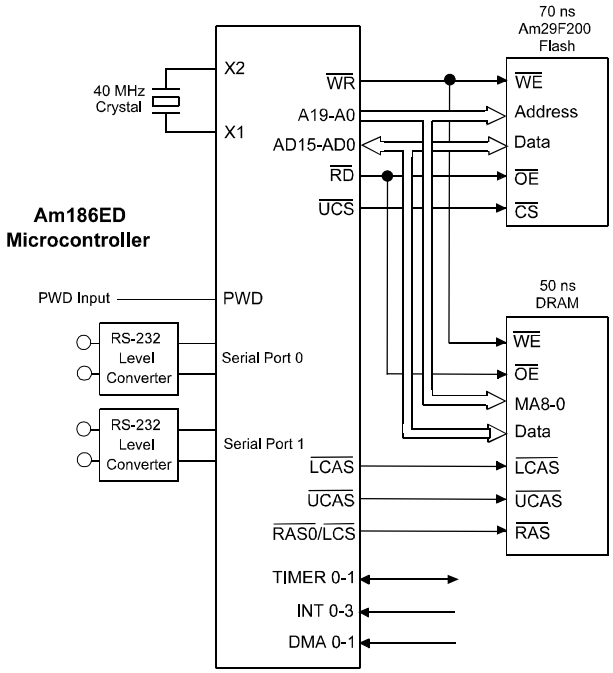
*** RTS1/RTR1 and CTS1/ENRX1 are multiplexed with PCS3 and PCS2, respectively. See the pin descriptions in Chapter 3.*

1.3 APPLICATION CONSIDERATIONS

The integration enhancements of the Am186ED/EDLV microcontrollers provide a high-performance, low-system-cost solution for 16-bit embedded microcontroller designs. The nonmultiplexed address bus eliminates the need for system-support logic to interface memory devices, while the multiplexed address/data bus maintains the value of previously engineered, customer-specific peripherals and circuits within the upgraded design.

Figure 1-2 illustrates an example system design that uses the integrated peripheral set to achieve high performance with reduced system cost.

Figure 1-2 Basic Functional System Design



1.3.1 Clock Generation

The integrated clock generation circuitry of the Am186ED/EDLV microcontrollers enables the use of a 1x crystal frequency. The design in Figure 1-2 achieves 40-MHz CPU operation, while using a 40-MHz crystal.

1.3.2 Memory Interface

The Am186ED/EDLV microcontrollers integrate a versatile memory controller which supports direct memory accesses to DRAM, SRAM, Flash, EPROM, and ROM. No external glue logic is required and all required control signals are provided. The peripheral chip selects have been enhanced to allow them to overlap the DRAM. This allows a small 1.75K portion of the DRAM memory space to be used for peripherals without bus contention.

The improved memory timing specifications of the Am186ED/EDLV microcontrollers allow for zero-wait-state operation using 50-ns DRAM or with 70-ns SRAM or Flash memory, both at a 40-MHz clock speed. 60-ns DRAM requires one wait state at 40 MHz and zero wait states at 33 MHz and below. 70-ns DRAM requires two wait states at 40 MHz, one wait state at 33 MHz, and zero wait states at 25 MHz and below. This reduces overall

system cost by enabling the use of commonly available memory speeds and taking advantage of DRAM's lower cost per bit over SRAM.

Figure 1-2 shows an implementation of an RS-232 console or modem communications port. The RS-232 to CMOS voltage-level converter is required for the electrical interface with the external device.

Figure 1-2 also illustrates the direct memory interface of the Am186ED/EDLV microcontrollers. The memory interface requires the following:

- The processor's A19–A0 bus connects to the memory address inputs.
- The AD bus connects to the data inputs and outputs.
- The chip selects connect to the memory chip-select inputs.
- The odd A1–A17 address pins connect to the DRAM multiplexed address bus.

Read operations require that the $\overline{RD}$ output connects to the DRAM Output Enable ($\overline{OE}$) pin. Write operations use the $\overline{WR}$ output connected to the DRAM Write Enable ($\overline{WE}$) pin. The $\overline{UCAS}$ and $\overline{LCAS}$ pins provide byte selection.

CHAPTER

2 PROGRAMMING

All members of the Am186 and Am188 family of microcontrollers, including the Am186ED/EDLV microcontrollers, contain the same basic set of registers, instructions, and addressing modes, and are compatible with the original industry-standard 186 microcontroller parts.

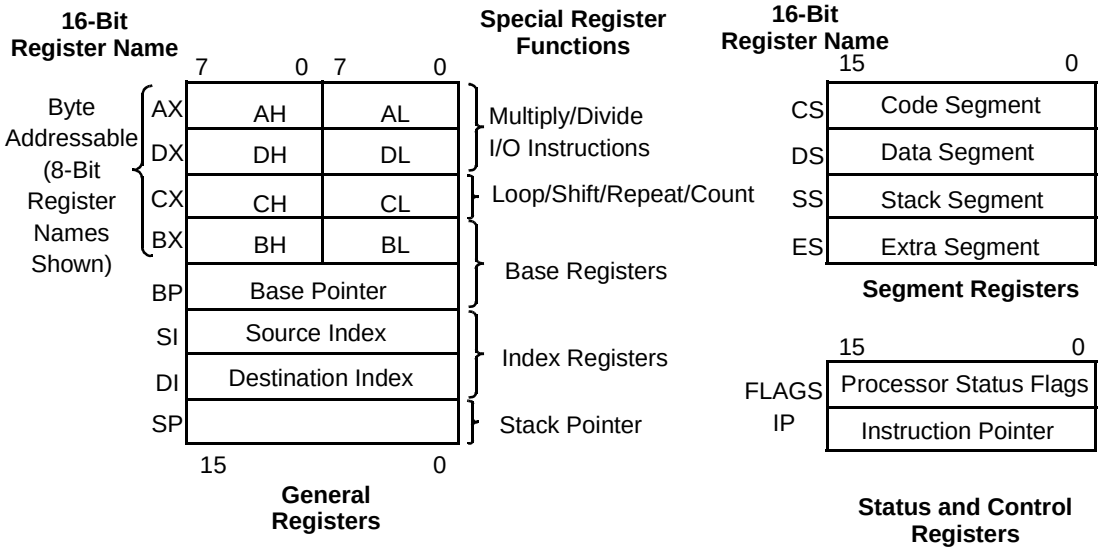
2.1 REGISTER SET

The base architecture of the Am186ED/EDLV 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 Am186ED/EDLV microcontrollers have additional 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 peripheral registers, are described in the chapters that follow.

Figure 2-1 Register Set

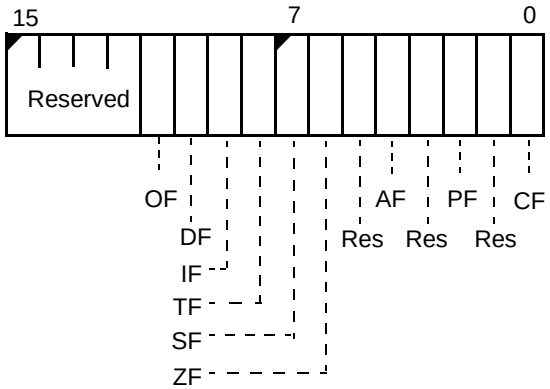


2.1.1 Processor Status Flags Register

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.

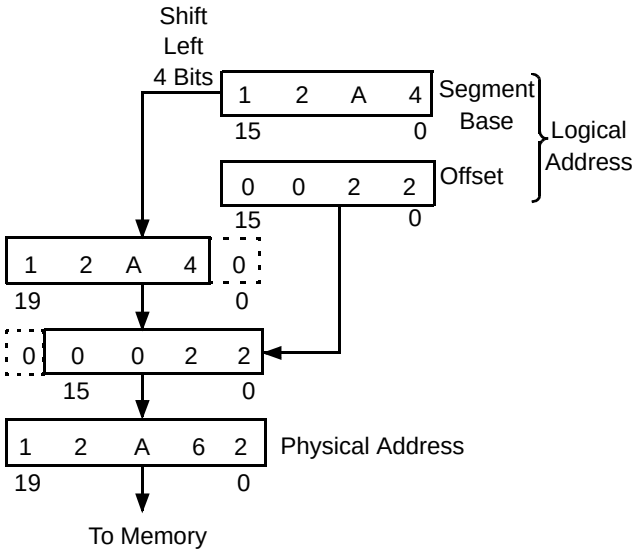
For example, if the segment register is loaded with 12A4h and the offset is 0022h, the resultant address is 12A62h (see Figure 2-3). To find the result:

1. The segment register contains 12A4h.
2. The segment register is shifted left 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 microcontrollers provide 64K of I/O space (see Figure 2-4).

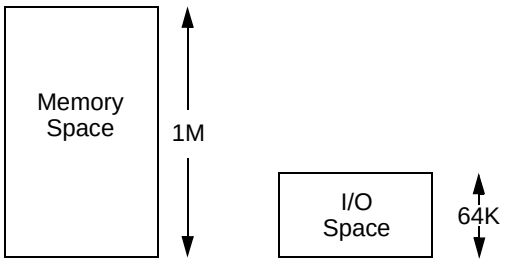
Figure 2-3 Physical Address Generation



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 Am186ED/EDLV 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 shares the standard 186 instruction set. (Refer to *Am186™ and Am188™ Family Instruction Set Manual*, order # 21267.) 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 later in this section.

Table 2-1 lists the instructions for the Am186ED/EDLV microcontrollers in alphabetical order.

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 Am186ED/EDLV microcontrollers 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 2-2).

Table 2-2 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

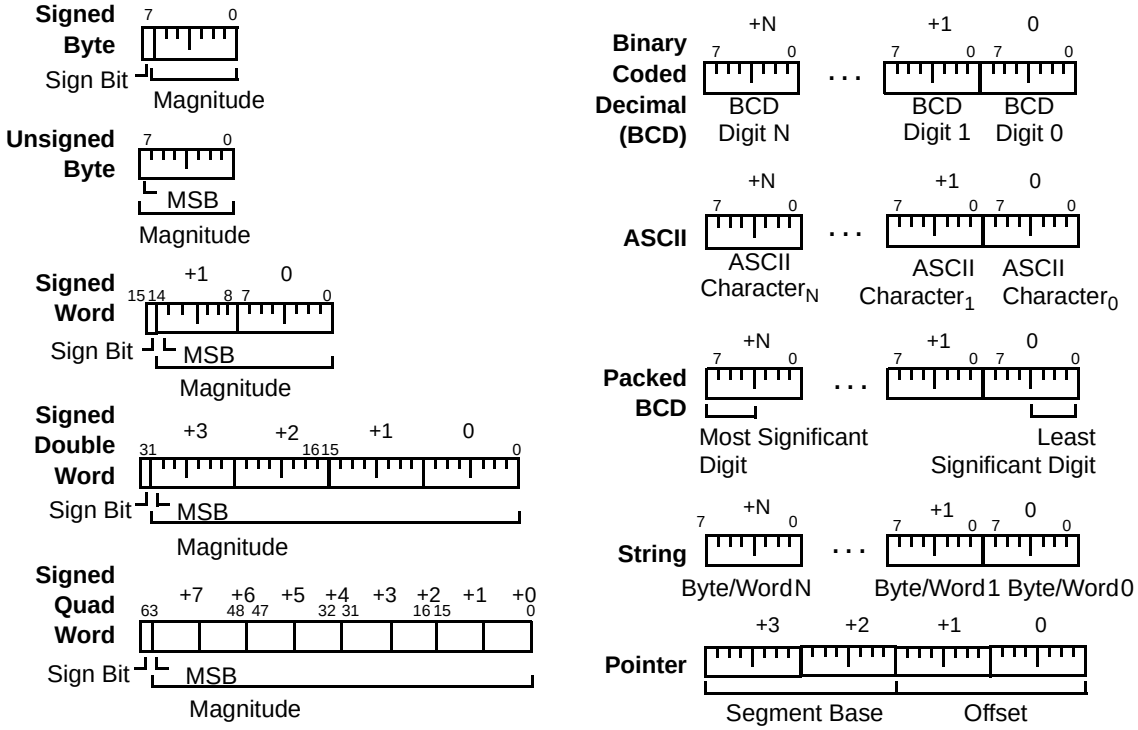
The Am186ED/EDLV microcontrollers directly support the following data types:

- **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.

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

In general, individual data elements must fit within defined segment limits. Figure 2-5 graphically represents the data types supported by the Am186ED/EDLV microcontrollers.

Figure 2-5 Supported Data Types



2.7 ADDRESSING MODES

The Am186ED/EDLV 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 2-3).

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 SI, DI, BX, or BP.
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 (SI or DI).
5. **Based Indexed Mode**—The operand offset is the sum of the contents of a base register and an index register.
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 2-3 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 Am186ED/EDLV 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) (A19/PIO9, A18/PIO8, A17/PIO7)**

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

While the Am186ED/EDLV microcontrollers directly interface DRAM, A19–A0 will serve as the nonmultiplexed address bus for SRAM, FLASH, PROM, EPROM, and peripherals. The odd address pins (A17, A15, A13, A11, A9, A7, A5, A3, and A1) will have both the row and column address during a DRAM space access. The odd address signals connect directly to the row and column multiplexed bus of the DRAM. The even address pins (A18, A16, A14, A12, A10, A8, A6, A4, A2, and A0) and A19 will have the initial address asserted during the full DRAM access. These signals will not transition during a DRAM access.

AD15–AD8 **Address and Data Bus (input/output, three-state, synchronous, level-sensitive)**

AD15–AD8—These time-multiplexed pins supply memory or I/O addresses and data to the system. This bus can supply an address to the system during the first period of a bus cycle (t_1). 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 deasserted, 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) can also be used to load system configuration information into the internal reset configuration register.

When accesses are made to 8-bit-wide memory regions, AD15–AD8 drive their corresponding address signals throughout the access. If the disable address phase and 8-bit mode are selected (see the $\overline{\text{ADEN}}$ description with the $\overline{\text{BHE}}$/$\overline{\text{ADEN}}$ pin), then AD15–AD8 are three-stated during t_1 and driven with their corresponding address signal from t_2 to t_4.

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). In 8-bit mode, AD7–AD0 supplies the data for both high and low bytes.

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 deasserted, 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) can also be used to load system configuration information into the internal reset configuration register.

ALE **Address Latch Enable (output, synchronous)**

This pin indicates to the system that an address appears on the address and data bus (AD15–AD0). The address is guaranteed to be valid on the trailing edge of ALE. This pin is three-stated during ONCE mode.

ALE is three-stated and held resistively Low during a bus hold condition. In addition, ALE has a weak internal pulldown resistor that is active during reset, so that an external device does not get a spurious ALE during reset.

ARDY **Asynchronous Ready (input, asynchronous, level-sensitive)**

This pin is a true asynchronous ready that indicates to the microcontroller that the addressed memory space or I/O device will complete a data transfer. The ARDY pin is asynchronous to CLKOUTA and is active High. To guarantee the number of wait states inserted, ARDY or SRDY must be synchronized to CLKOUTA. If the falling edge of ARDY is not synchronized to CLKOUTA as specified, an additional clock period can be added.

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.

$\overline{\text{BHE}}$/$\overline{\text{ADEN}}$ **Bus High Enable (three-state, output, synchronous) Address Enable (input, internal pullup)**

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

$\overline{\text{BHE}}$	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	Reserved

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

$\overline{\text{WLB}}$ and $\overline{\text{WHB}}$ implement the functionality of $\overline{\text{BHE}}$ and AD0 for High and Low byte-write enables. $\overline{\text{UCAS}}$ and $\overline{\text{LCAS}}$ implement High and Low-byte selection for DRAM devices.

$\overline{\text{BHE}}$/$\overline{\text{ADEN}}$ also signals DRAM refresh cycles when using the multiplexed address and data (AD) bus. A refresh cycle is indicated when both $\overline{\text{BHE}}$/$\overline{\text{ADEN}}$ and AD0 are High. During refresh cycles, the A bus is indeterminate and the AD bus is driven to FFFFh 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.

$\overline{\text{ADEN}}$—If $\overline{\text{BHE}}$/$\overline{\text{ADEN}}$ is held High or left floating during power-on reset, the address portion of the AD bus (AD15–AD0) is enabled or disabled during $\overline{\text{LCS}}$ and $\overline{\text{UCS}}$ bus cycles based on the DA bit in the LMCS and UMCS registers. If the DA bit is set, the AD bus will not drive the address during t_1. There is a weak internal pullup resistor on $\overline{\text{BHE}}$/$\overline{\text{ADEN}}$ so no external pullup is required. Disabling the address phase reduces power consumption.

If $\overline{\text{BHE}}$/$\overline{\text{ADEN}}$ is held Low on power-on reset, the AD bus drives both addresses and data, regardless of the DA bit setting. The pin is sampled on the rising edge of $\overline{\text{RES}}$. (S6 and $\overline{\text{UZI}}$ also assume their normal functionality in this instance. See Table 3-1 on page 15.) The internal pullup on $\overline{\text{ADEN}}$ is ~9 kohm.

Note: *For 8-bit accesses, AD15–AD8 are driven with addresses during the t_2–t_4 bus cycle, regardless of the setting of the DA bit in the UMCS and LMCS registers.*

CLKOUTA **Clock Output A (output, synchronous)**

This pin supplies the internal clock to the system. Depending on the value of the System Configuration register (SYSCON), CLKOUTA operates at either the PLL frequency (X1), the power-save frequency, or is held Low. CLKOUTA remains active during reset and bus hold conditions.

All AC timing specs that use a clock relate to CLKOUTA.

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

This pin supplies an additional clock with a delayed output compared to CLKOUTA. Depending upon the value of the System Configuration register (SYSCON), CLKOUTB operates at either the PLL frequency (X1), the power-save frequency, or is held Low. CLKOUTB remains active during reset and bus hold conditions.

CLKOUTB is not used for AC timing specs.

$\overline{\text{CTS}}$0/$\overline{\text{ENRX}}$0/PIO21

Clear-to-Send 0 (input, asynchronous)
Enable-Receiver-Request 0 (input, asynchronous)

$\overline{\text{CTS}}$0—This pin provides the Clear-to-Send signal for asynchronous serial port 0 when the $\overline{\text{ENRX}}$0 bit in the AUXCON register is 0 and hardware flow control is enabled for the port (FC bit in the serial port 0 control register is set). The $\overline{\text{CTS}}$0 signal gates the transmission of data from the associated serial port transmit register. When $\overline{\text{CTS}}$0 is asserted, the transmitter begins transmission of a frame of data, if any is available. If $\overline{\text{CTS}}$0 is deasserted, the transmitter holds the data in the serial port transmit register. The value of $\overline{\text{CTS}}$0 is checked only at the beginning of the transmission of the frame.

$\overline{\text{ENRX}}$0—This pin provides the Enable Receiver Request for asynchronous serial port 0 when the ENRX0 bit in the AUXCON register is 1 and hardware flow control is enabled for the port (FC bit in the serial port 0 control register is set). The $\overline{\text{ENRX}}$0 signal enables the receiver for the associated serial port.

$\overline{\text{DEN}}$/$\overline{\text{DS}}$/PIO5 **Data Enable (output, three-state, synchronous)**
Data Strobe (output, three-state, synchronous)

$\overline{\text{DEN}}$—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.

$\overline{\text{DS}}$—The data strobe provides a signal where the write cycle timing is identical to the read cycle timing. When used with other control signals, $\overline{\text{DS}}$ provides an interface for 68K-type peripherals without the need for additional system interface logic.

When $\overline{\text{DS}}$ is asserted, addresses are valid. When $\overline{\text{DS}}$ is asserted on writes, data is valid. When $\overline{\text{DS}}$ is asserted on reads, data can be asserted on the AD bus.

Note: *This pin resets to $\overline{\text{DEN}}$.*

DRQ0/INT5/PIO12

DMA Request 0 (input, synchronous, level-sensitive)
Maskable Interrupt Request 5 (input, asynchronous, edge-triggered)

DRQ0—This pin indicates to the microcontroller that an external device is ready for DMA channel 0 to perform a transfer. DRQ0 is level-triggered and internally synchronized. DRQ0 is not latched and must remain active until serviced.

INT5—If DMA 0 is not enabled or DMA 0 is not being used with external synchronization, INT5 can be used as an additional external interrupt

request. INT5 shares the DMA 0 interrupt type (0Ah) and register control bits.

INT5 is edge-triggered only and must be held until the interrupt is acknowledged.

DRQ1/INT6/PIO13

DMA Request 1 (input, synchronous, level-sensitive)
Maskable Interrupt Request 6 (input, asynchronous, edge-triggered)

DRQ1—This pin indicates to the microcontroller that an external device is ready for DMA channel 1 to perform a transfer. DRQ1 is level-triggered and internally synchronized. DRQ1 is not latched and must remain active until serviced.

INT6—If DMA 1 is not enabled or DMA 1 is not being used with external synchronization, INT6 can be used as an additional external interrupt request. INT6 shares the DMA 1 interrupt type (0Bh) and register control bits.

INT6 is edge-triggered only and must be held until the interrupt is acknowledged.

DT/$\overline{R}$/PIO4 **Data Transmit or Receive (output, three-state, synchronous)**

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

GND **Ground**

Ground pins connect the microcontroller to the system ground.

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

This pin is asserted High to indicate to an external bus master that the microcontroller has released 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. It then relinquishes control of the bus to the external bus master by asserting HLDA and floating $\overline{DEN}$, $\overline{RD}$, $\overline{WR}$, $\overline{S}2$–$\overline{S}0$, AD15–AD0, S6, A19–A0, $\overline{BHE}$, $\overline{WHB}$, $\overline{WLB}$, and DT/$\overline{R}$. The following chip selects are three-stated (then will be held High with an ~10-kohm resistor): $\overline{UCS}$, $\overline{LCS}$, $\overline{MCS}3$–$\overline{MCS}0$, $\overline{PCS}6$–$\overline{PCS}5$, $\overline{PCS}3$–$\overline{PCS}0$, $\overline{RAS}0$, $\overline{RAS}1$, $\overline{UCAS}$, and $\overline{LCAS}$. ALE is also three-stated (then will be held Low with an ~10-kohm resistor.

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 (for example, to 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.

The Am186ED/EDLV microcontroller's 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.

For more information, see the HLDA pin description.

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 interrupt recognition, the requesting device 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 INT1 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 interrupt recognition, the requesting device 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 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/PWD/PIO31
Maskable Interrupt Request 2 (input, asynchronous)
Interrupt Acknowledge 0 (output, synchronous)
Pulse Width Demodulator (input, Schmitt trigger)

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 interrupt recognition, the requesting device 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.

PWD—If pulse width demodulation is enabled, PWD processes a signal through the Schmitt trigger. PWD is used internally to drive TIMERIN0 and INT2, and PWD is inverted internally to drive TIMERIN1 and INT4. If INT2 and INT4 are enabled and timer 0 and timer 1 are properly configured, the pulse width of the alternating PWD signal can be calculated by comparing the values in timer 0 and timer 1.

In PWD mode, the signals TIMERIN0/PIO11, TIMERIN1/PIO0, and INT4/PIO30 can be used as PIOs. If they are not used as PIOs they are ignored internally. The level of INT2/$\overline{\text{INTA}}$0/PWD/PIO31 is reflected in the PIO data register for PIO31 as if it was a PIO.

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 can be edge-triggered or level-triggered. To guarantee interrupt recognition, the requesting device 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/PIO30 **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 can be edge-triggered or level-triggered. To guarantee interrupt recognition, the requesting device must continue asserting INT4 until the request is acknowledged.

When pulse width demodulation mode is enabled, the INT4 signal is used internally to indicate a High-to-Low transition on the PWD signal. When pulse width demodulation mode is enabled, INT4/PIO30 can be used as a PIO.

LCS/ONCE0/RAS0

Lower Memory Chip Select (output, synchronous, internal pullup)
ONCE Mode Request 0 (input)
Row Address Strobe 0

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. LCS is configured for 8-bit or 16-bit bus size by the auxiliary configuration register.

LCS is three-stated and held resistively High during a bus hold condition. In addition, LCS has an ~9 kohm internal pullup resistor that is active during reset.

ONCE0—During reset, this pin and ONCE1 indicate to the microcontroller the mode in which it should operate. ONCE0 and ONCE1 are sampled on the rising edge of 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, ONCE0 has a weak internal pullup resistor that is active only during reset.

RAS0—This pin is the row address strobe for the lower DRAM block. The selection of RAS0 or LCS functionality, along with their configurations, are set using the LCMS register.

RAS0 is three-stated and held resistively High during a bus hold condition. In addition, RAS0 has a weak internal pullup resistor that is active during reset.

MCS0/PIO14

Midrange Memory Chip Select 0
(output, synchronous, internal pullup)

This pin indicates 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. MCS0 can be programmed as the chip select for the entire middle chip select address range. This mode is recommended when using DRAM since the MCS1, MCS2, and MCS3 chip selects function as RAS and CAS signals for the DRAM interface and are not available as chip selects.

MCS0 is configured for 8-bit or 16-bit bus size by the auxiliary configuration register. MCS0 is three-stated and held resistively High during a bus hold condition. In addition, MCS0 has a weak internal pullup resistor that is active during reset.

$\overline{\text{MCS1}}$/$\overline{\text{UCAS}}$/PIO15

Midrange Memory Chip Select 1
(output, synchronous, internal pullup)
Upper Column Address Strobe

This pin indicates 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. $\overline{\text{MCS1}}$ is configured for 8-bit or 16-bit bus size via the auxiliary configuration register.

$\overline{\text{MCS1}}$ is three-stated and held resistively High during a bus hold condition. In addition, $\overline{\text{MCS1}}$ has a weak internal pullup resistor that is active during reset.

If $\overline{\text{MCS0}}$ is programmed to be active for the entire middle chip-select range, then this signal is available as a PIO or a DRAM control. If this signal is not programmed as a PIO or DRAM control and if $\overline{\text{MCS0}}$ is programmed for the entire middle chip-select range, this signal operates normally.

$\overline{\text{UCAS}}$—When either bank of DRAM is activated, the $\overline{\text{UCAS}}$ functionality is enabled. The $\overline{\text{UCAS}}$ activates when the DRAM access is for the AD15–AD8 byte. $\overline{\text{UCAS}}$ also activates at the start of a DRAM refresh access.

$\overline{\text{UCAS}}$ is three-stated and held resistively High during a bus hold condition. In addition, $\overline{\text{UCAS}}$ has a weak internal pullup resistor that is active during reset.

$\overline{\text{MCS2}}$/$\overline{\text{LCAS}}$/PIO24

Midrange Memory Chip Select 2
(output, synchronous, internal pullup)
Lower Column Address Strobe

This pin indicates 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. $\overline{\text{MCS2}}$ is configured for 8-bit or 16-bit bus size via the Auxiliary Configuration register.

$\overline{\text{MCS2}}$ is three-stated and held resistively High during a bus hold condition. In addition, it has a weak internal pullup resistor that is active during reset.

If $\overline{\text{MCS0}}$ is programmed to be active for the entire middle chip-select range, then this signal is available as a PIO or a DRAM control. If this pin is not programmed as a PIO or DRAM control and if $\overline{\text{MCS0}}$ is programmed for the entire middle chip-select range, this signal operates normally.

$\overline{\text{LCAS}}$—When either bank of DRAM is activated, the $\overline{\text{LCAS}}$ functionality is enabled. The $\overline{\text{LCAS}}$ activates when the DRAM access is for the AD7–AD0 byte. $\overline{\text{LCAS}}$ also activates at the start of a DRAM refresh access.

$\overline{\text{LCAS}}$ is three-stated and held resistively High during a bus hold condition. In addition, $\overline{\text{LCAS}}$ has a weak internal pullup resistor that is active during reset.

$\overline{MCS}3/\overline{RAS}1/PIO25$

Midrange Memory Chip Select 3 (output, synchronous, internal pullup)
Row Address Strobe 1 (output, synchronous)

$\overline{MCS}3$—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. $\overline{MCS}3$ is configured for 8-bit or 16-bit bus size by the Auxiliary Configuration register.

$\overline{MCS}3$ is three-stated and held resistively High during a bus hold condition. In addition, this pin has a weak internal pullup resistor that is active during reset.

If $\overline{MCS}0$ is programmed for the entire middle chip-select range, then this signal is available as a PIO or a DRAM control. If $\overline{MCS}3$ is not programmed as a PIO or DRAM control and if $\overline{MCS}0$ is programmed for the entire middle chip-select range, this signal operates normally.

$\overline{RAS}1$—This pin is the row address strobe for the upper DRAM block. The selection of $\overline{RAS}1$ or $\overline{UCS}$ functionality, along with their configurations, are set using the UMCS register. When $\overline{RAS}1$ is activated, the code activating $\overline{RAS}1$ must not reside in the $\overline{UCS}$ memory block. When $\overline{RAS}1$ is activated, $\overline{UCS}$ is automatically deactivated and remains negated.

$\overline{RAS}1$ is three-stated and held resistively High during a bus hold condition. In addition, $\overline{RAS}1$ has a weak internal pullup resistor that is active during 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 INT6–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 does 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.

$\overline{PCS}$1–$\overline{PCS}$0 **($\overline{PCS}$1/PIO17, $\overline{PCS}$0/PIO16)**
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.

The $\overline{PCS}$ chip selects can overlap either block of DRAM. The $\overline{PCS}$ chip selects must have the same or greater number of wait states as the bank of DRAM they overlap. The $\overline{PCS}$ signals take precedence over DRAM accesses when DRAM and memory-mapped peripherals overlap.

$\overline{PCS}$1–$\overline{PCS}$0 are three-stated and held resistively High during a bus hold condition. In addition, $\overline{PCS}$1–$\overline{PCS}$0 each have a weak internal pullup resistor that is active during reset.

Unlike the $\overline{UCS}$ and $\overline{LCS}$ chip selects, the $\overline{PCS}$ 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 earlier generations of the 80C186 and 80C188 microcontrollers. $\overline{PCS}$0–$\overline{PCS}$1 also have extended wait state options.

$\overline{PCS}$2/$\overline{CTS}$1/$\overline{ENRX}$1/PIO18
Peripheral Chip Select 2 (output, synchronous)
Clear-to-Send 1 (input, asynchronous)
Enable-Receiver-Request 1 (input, asynchronous)

$\overline{PCS}$2—This pin provides the Peripheral Chip Select 2 signal to the system when hardware flow control is not enabled for asynchronous serial port 1. The $\overline{PCS}$2 signal indicates 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.

The $\overline{PCS}$ chip selects can overlap either block of DRAM. The $\overline{PCS}$ chip selects must have the same or greater number of wait states as the bank of DRAM it overlaps. The $\overline{PCS}$ signals take precedence over DRAM accesses when DRAM and memory-mapped peripherals overlap.

$\overline{PCS}$2 is three-stated and held resistively High during a bus hold condition. In addition, $\overline{PCS}$2 has a weak internal pullup resistor that is active during reset.

Unlike the $\overline{UCS}$ and $\overline{LCS}$ chip selects, the $\overline{PCS}$ 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. $\overline{PCS}$2 also has extended wait state options.

$\overline{CTS}$1—This pin provides the Clear-to-Send signal for asynchronous serial port 1 when the ENRX1 bit in the AUXCON register is 0 and hardware flow control is enabled for the port (FC bit in the serial port 1 control register is set). The $\overline{CTS}$1 signal gates the transmission of data from the associated serial port transmit register. When $\overline{CTS}$1 is asserted, the transmitter will begin transmission of a frame of data, if

any is available. If $\overline{CTS}1$ is deasserted, the transmitter holds the data in the serial port transmit register. The value of $\overline{CTS}1$ is checked only at the beginning of the transmission of the frame.

$\overline{ENRX}1$—This pin provides the Enable Receiver Request for asynchronous serial port 1 when the ENRX1 bit in the AUXCON register is 1 and hardware flow control is enabled for the port (FC bit in the serial port 1 control register is set). The $\overline{ENRX}1$ signal enables the receiver for the associated serial port.

$\overline{PCS}3/\overline{RTS}1/\overline{RTR}1/PIO19$

Peripheral Chip Select 3 (output, synchronous)
Ready-to-Send 1 (output, asynchronous)
Ready-to-Receive 1 (output, asynchronous)

$\overline{PCS}3$—This pin provides the Peripheral Chip Select 3 signal to the system when hardware flow control is not enabled for asynchronous serial port 1. The $\overline{PCS}3$ signal indicates 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.

The $\overline{PCS}$ chip selects can overlap either block of DRAM. The $\overline{PCS}$ chip selects must have the same or greater number of wait states as the bank of DRAM they overlap. The $\overline{PCS}$ signals take precedence over DRAM accesses when DRAM and memory-mapped peripherals overlap.

$\overline{PCS}3$ is three-stated and held resistively High during a bus hold condition. In addition, $\overline{PCS}3$ has a weak internal pullup resistor that is active during reset.

Unlike the $\overline{UCS}$ and $\overline{LCS}$ chip selects, the $\overline{PCS}$ 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. $\overline{PCS}3$ also has extended wait state options.

$\overline{RTS}1$—This pin provides the Ready-to-Send signal for asynchronous serial port 1 when the $\overline{RTS}1$ bit in the AUXCON register is 1 and hardware flow control is enabled for the port (FC bit in the serial port 1 control register is set). The $\overline{RTS}1$ signal is asserted when the associated serial port transmit register contains data which has not been transmitted.

$\overline{RTR}1$—This pin provides the Ready-to-Receive signal for asynchronous serial port 1 when the $\overline{RTS}1$ bit in the AUXCON register is 0 and hardware flow control is enabled for the port (FC bit in the serial port 1 control register is set). The $\overline{RTR}1$ signal is asserted when the associated serial port receive register does not contain valid, unread data.

$\overline{\text{PCS}}$5/A1/PIO3 **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.

The $\overline{\text{PCS}}$ chip selects can overlap either block of DRAM. The $\overline{\text{PCS}}$ chip selects must have the same or greater number of wait states as the bank of DRAM they overlap. The $\overline{\text{PCS}}$ signals take precedence over DRAM accesses when DRAM and memory-mapped peripherals overlap.

$\overline{\text{PCS}}$5 is three-stated and held resistively High during a bus hold condition. In addition, $\overline{\text{PCS}}$5 has a weak internal pullup resistor that is active during reset.

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. $\overline{\text{PCS}}$5 also has extended wait state options.

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/PIO2 **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. The $\overline{\text{PCS}}$ signals take precedence over DRAM accesses when DRAM and memory-mapped peripherals overlap. During a bus hold condition, the signals will be put into a high-impedance mode and should have a pullup resistor. $\overline{\text{PCS}}$6 is held High during a bus hold condition or reset.

The $\overline{\text{PCS}}$ chip selects can overlap either block of DRAM. The $\overline{\text{PCS}}$ chip selects must have the same or greater number of wait states as the bank of DRAM they overlap. The $\overline{\text{PCS}}$ signals take precedence over DRAM accesses when DRAM and memory-mapped peripherals overlap.

$\overline{\text{PCS}}$6 is three-stated and held resistively High during a bus hold condition. In addition, $\overline{\text{PCS}}$6 has a weak internal pullup resistor that is active during reset.

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. $\overline{\text{PCS}}$6 also has extended wait state options.

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 Am186ED/EDLV microcontrollers provide 32 individually programmable I/O pins. Each PIO can be programmed with the following attributes: PIO function (enabled/disabled), direction (input/output), and weak pullup or pulldown. The pins that are multiplexed with PIO31–PIO0 are listed in Table 3-1 and Table 3-2.

After power-on reset, the PIO pins default to various configurations. The column titled *Power-On Reset Status* in Table 3-1 and Table 3-2 lists the defaults for the PIOs. Most of the PIO pins are configured as PIO inputs with pullup after power-on reset. The system initialization code must reconfigure any PIO pins 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/$\overline{R}$, $\overline{DEN}$, and SRDY pins also default to normal operation on power-on reset. PIO15 and PIO24 should be set to normal operation before enabling either bank of DRAM. PIO25 should be set to normal operation before enabling the upper bank of DRAM.

$\overline{RD}$ **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 to not 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 requires the microcontroller to perform a reset. When $\overline{RES}$ is asserted, the microcontroller immediately terminates its present activity, clears its internal logic, and transfers CPU control to the reset address, FFFF0h.

$\overline{RES}$ must be held Low for at least 1 ms.

$\overline{RES}$ can be asserted asynchronously 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.

Table 3-1 Numeric PIO Pin Designations

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	$\overline{\text{DEN}}$/$\overline{\text{DS}}$	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/INT5	Input with pullup
13	DRQ1/INT6	Input with pullup
14	$\overline{\text{MCS}}$0	Input with pullup
15	$\overline{\text{MCS}}$1/$\overline{\text{UCAS}}$	Input with pullup
16	$\overline{\text{PCS}}$0	Input with pullup
17	$\overline{\text{PCS}}$1	Input with pullup
18	$\overline{\text{PCS}}$2/$\overline{\text{CTS}}$1/$\overline{\text{ENRX}}$1	Input with pullup
19	$\overline{\text{PCS}}$3/$\overline{\text{RTS}}$1/$\overline{\text{RTR}}$1	Input with pullup
20	$\overline{\text{RTS}}$0/$\overline{\text{RTR}}$0	Input with pullup
21	$\overline{\text{CTS}}$0/$\overline{\text{ENRX}}$0	Input with pullup
22	TXD0	Input with pullup
23	RXD0	Input with pullup
24	$\overline{\text{MCS}}$2/$\overline{\text{LCAS}}$	Input with pullup
25	$\overline{\text{MCS}}$3/$\overline{\text{RAS}}$1	Input with pullup
26(1,2)	UZI	Input with pullup
27	TXD1	Input with pullup
28	RXD1	Input with pullup
29(1,2)	S6/$\overline{\text{CLKDIV}}$2	Input with pullup
30	INT4	Input with pullup
31	INT2/$\overline{\text{INTA}}$0/PWD	Input with pullup

Notes:

1. *These pins are used by many 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 $\overline{BHE}$/$\overline{ADEN}$ 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 Alphabetic PIO Pin Designations

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{CTS}$0/$\overline{ENRX}$0	21	Input with pullup
$\overline{DEN}$/$\overline{DS}$	5	Normal operation(3)
DRQ0/INT5	12	Input with pullup
DRQ1/INT6	13	Input with pullup
DT/$\overline{R}$	4	Normal operation(3)
INT2/$\overline{INTA}$0/PWD	31	Input with pullup
INT4	30	Input with pullup
$\overline{MCS}$0	14	Input with pullup
$\overline{MCS}$1/$\overline{UCAS}$	15	Input with pullup
$\overline{MCS}$2/$\overline{LCAS}$	24	Input with pullup
$\overline{MCS}$3/$\overline{RAS}$1	25	Input with pullup
$\overline{PCS}$0	16	Input with pullup
$\overline{PCS}$1	17	Input with pullup
$\overline{PCS}$2/$\overline{CTS}$1/$\overline{ENRX}$1	18	Input with pullup
$\overline{PCS}$3/$\overline{RTS}$1/$\overline{RTR}$1	19	Input with pullup
$\overline{PCS}$5/A1	3	Input with pullup
$\overline{PCS}$6/A2	2	Input with pullup
$\overline{RTS}$0/$\overline{RTR}$0	20	Input with pullup
RXD0	23	Input with pullup
RXD1	28	Input with pullup
S6/$\overline{CLKDIV}$2(1,2)	29	Input with pullup
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
TXD0	22	Input with pullup
TXD1	27	Input with pullup
$\overline{UZI}$(1,2)	26	Input with pullup

Notes:

1. *These pins are used by many 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 $\overline{BHE}$/$\overline{ADEN}$ 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.*

$\overline{RTS}0/\overline{RTR}0/PIO20$

Ready-to-Send 0 (output, asynchronous)
Ready-to-Receive 0 (output, asynchronous)

$\overline{RTS}0$—This pin provides the Ready-to-Send signal for asynchronous serial port 0 when the $\overline{RTS}0$ bit in the AUXCON register is 1 and hardware flow control is enabled for the port (FC bit in the serial port 0 control register is set). The $\overline{RTS}0$ signal is asserted when the associated serial port transmit register contains data which has not been transmitted.

$\overline{RTR}0$—This pin provides the Ready-to-Receive signal for asynchronous serial port 0 when the $\overline{RTS}0$ bit in the AUXCON register is 0 and hardware flow control is enabled for the port (FC bit in the serial port 0 control register is set). The $\overline{RTR}0$ signal is asserted when the associated serial port receive register does not contain valid, unread data.

RXD0/PIO23 **Receive Data 0 (input, asynchronous)**

This pin supplies asynchronous serial receive data from the system to asynchronous serial port 0.

RXD1/PIO28 **Receive Data 1 (input, asynchronous)**

This pin supplies asynchronous serial receive data from the system to asynchronous serial port 1.

$\overline{S}2$/BTSEL **Bus Cycle Status (output, three-state, synchronous)**
Boot Mode Select

$\overline{S}2$—This pin indicates to the system the type of bus cycle in progress. $\overline{S}2$ can be used as a logical memory or I/O indicator. $\overline{S}2$–$\overline{S}0$ float during bus hold and hold acknowledge conditions. The $\overline{S}2$–$\overline{S}0$ pins are encoded as shown in the table following.

BTSEL—The Am186ED/EDLV microcontrollers can boot from 8- or 16-bit wide nonvolatile memory, based on the state of the BTSEL pin. If BTSEL is pulled High or left floating, an internal pullup sets the boot mode option to 16-bit. If BTSEL is pulled resistively Low during reset, the 8-bit boot mode option is selected. The status of the BTSEL pin is latched on the rising edge of reset. If 8-bit mode is selected, the width of the memory region associated with $\overline{UCS}$ can be changed in the AUXCON register.

This signal should never be tied to V_{CC} or V_{SS} directly since this pin is driven during normal operation. This signal should be tied Low with an external resistor if the 8-bit boot mode is to be used. The internal pullup resistor on BTSEL is ~9 kohm.

$\overline{S}1$–$\overline{S}0$ **Bus Cycle Status (output, three-state, synchronous)**

These pins indicate to the system the type of bus cycle in progress. $\overline{S}1$ can be used as a data transmit or receive indicator. $\overline{S}1$–$\overline{S}0$ float during bus hold and hold acknowledge conditions. The $\overline{S}2$–$\overline{S}0$ pins are encoded as shown in the following table.

$\overline{S}2$	$\overline{S}1$	$\overline{S}0$	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/PIO29

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/PIO29 is held Low during power-on reset, the chip enters clock divided 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/$\overline{\text{CLKDIV}}$2/PIO29 defaults to a PIO input with pullup, so the pin does not need to be driven High externally.

SRDY/PIO6 **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/PIO11 **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. When PIO11 is enabled, TMRIN0 is pulled High internally.

TMRIN0 is driven internally by INT2/$\overline{\text{INTA}}$0/PWD when pulse width demodulation mode is enabled. The TMRIN0/PIO11 pin can be used as a PIO when pulse width demodulation mode is enabled.

TMRIN1/PIO0 **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. When PIO0 is enabled, TMRIN1 is pulled High internally.

TMRIN1 is driven internally by INT2/$\overline{\text{INTA}}$0/PWD when pulse width demodulation mode is enabled. The TMRIN1/PIO0 pin can be used as a PIO when pulse width demodulation mode is enabled.

TMROUT0/PIO10 **Timer Output 0 (output, synchronous)**

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

TMROUT1/PIO1 **Timer Output 1 (output, synchronous)**

This pin supplies the system with either a single pulse or a continuous waveform with a programmable duty cycle. TMROUT1 floats during a bus hold or reset.

TXD0/PIO22 **Transmit Data 0 (output, asynchronous)**

This pin supplies asynchronous serial transmit data to the system from serial port 0.

TXD1/PIO27 **Transmit Data 1 (output, asynchronous)**

This pin supplies asynchronous serial transmit data to the system from serial port 1.

$\overline{\text{UCS}}$/$\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 three-stated and held resistively High during a bus hold condition. In addition, $\overline{\text{UCS}}$ has an ~9 kohm internal pullup resistor that is active during reset.

After reset, $\overline{\text{UCS}}$ is active for the 64 Kbyte memory range from F0000h to FFFFFh, including the reset address of FFFF0h.

When $\overline{\text{RAS}}$1 is activated, the code activating $\overline{\text{RAS}}$1 must not reside in the $\overline{\text{UCS}}$ memory block. When $\overline{\text{RAS}}$1 is activated, $\overline{\text{UCS}}$ is automatically deactivated and remains negated. This allows code to boot from $\overline{\text{UCS}}$, copy its code to another memory device, then activate a DRAM bank in place of the $\overline{\text{UCS}}$ memory block.

$\overline{\text{ONCE}}$1—During reset this pin and $\overline{\text{LCS}}$/$\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.

$\overline{\text{UZI}}$/PIO26 **Upper Zero Indicate (output, synchronous)**

This pin lets the designer determine if 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). $\overline{\text{UZI}}$ is the logical AND of the inverted A19–A16 bits.

It asserts in the first period of a bus cycle and is held throughout the cycle.

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

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

$\overline{WHB}$ **Write High Byte (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 microcontroller designs, information is provided by $\overline{BHE}$, AD0, and $\overline{WR}$. 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.

$\overline{WLB}$ **Write Low Byte (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 microcontroller designs, this information is provided by $\overline{BHE}$, AD0, and $\overline{WR}$. 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.

$\overline{WR}$ **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. $\overline{WR}$ should be used for DRAM write enable.

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, AD7–AD0, ALE, $\overline{BHE}$/$\overline{ADEN}$, CLKOUTA, $\overline{RD}$, $\overline{S}$2–$\overline{S}$0, S6/$\overline{CLKDIV}$2, and $\overline{UZI}$.

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

If $\overline{BHE}$/$\overline{ADEN}$ is held Low during the rising edge of $\overline{RES}$, S6 and $\overline{UZI}$ 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 Am186ED/EDLV microcontrollers continue to provide the multiplexed AD bus and, in addition, provides 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 during the normal address portion of the bus cycle for accesses to $\overline{\text{RAS}}$0, $\overline{\text{RAS}}$1, $\overline{\text{UCS}}$, and/or $\overline{\text{LCS}}$ 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, decreasing power consumption and reducing processor switching noise. In 8-bit mode, the address is driven on AD15–AD8 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.

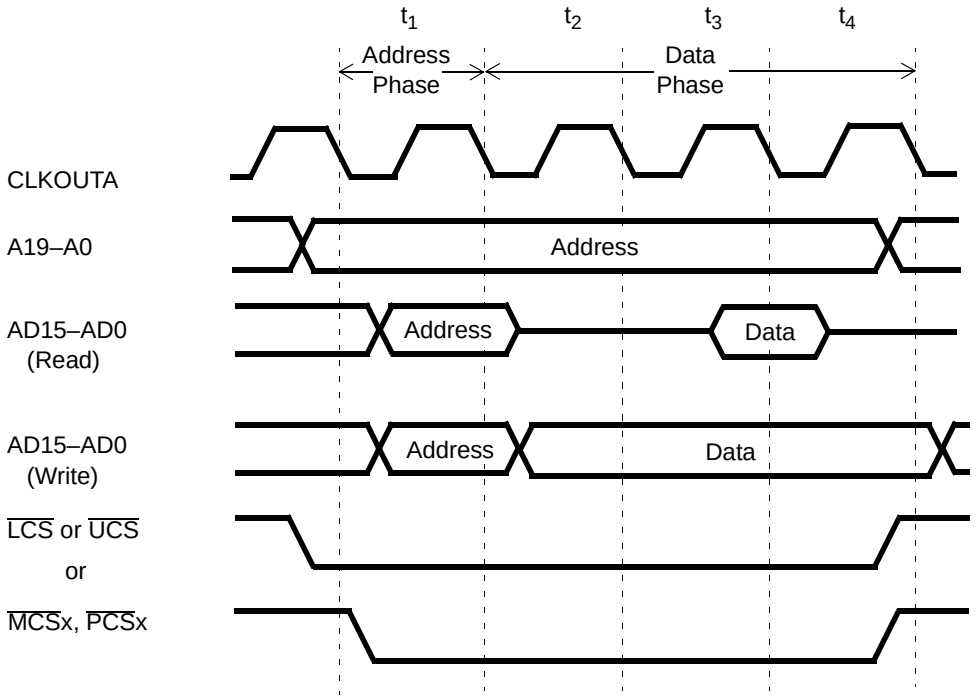
Figure 3-1 shows the affected signals during a normal read or write operation for 16-bit mode. The address and data are multiplexed onto the AD bus.

Figure 3-2 shows a 16-bit mode bus cycle when address bus disable is in effect. This results in the AD bus operating in a nonmultiplexed address/data mode. The A bus has the address during a read or write operation.

Figure 3-3 shows the affected signals during a normal read or write operation for 8-bit mode. The multiplexed address/data mode is compatible with the 80C186 and 80C188 microcontrollers and might be used to take advantage of existing logic or peripherals.

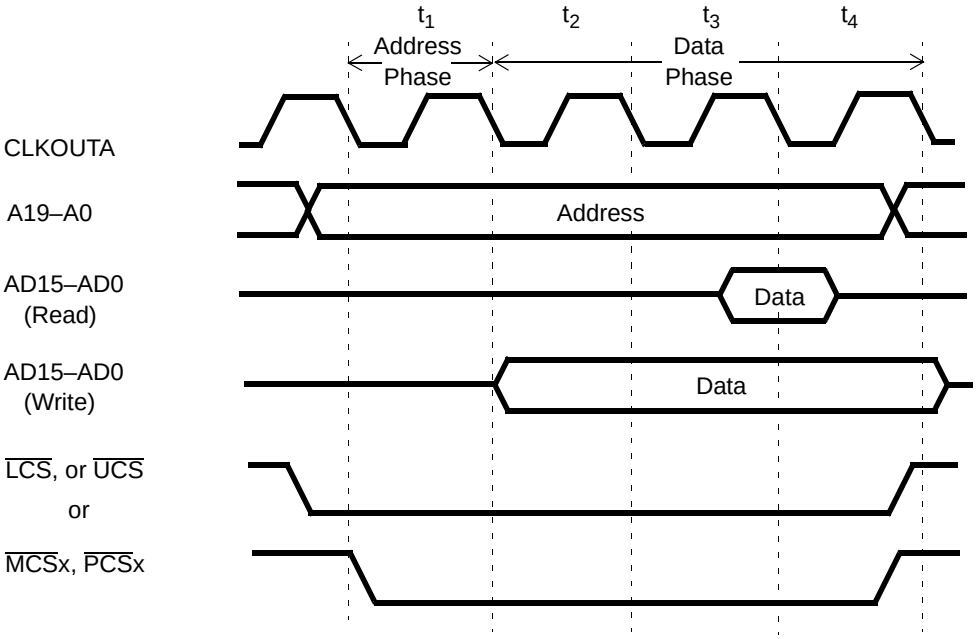
Figure 3-4 shows an 8-bit mode bus cycle when address bus disable is in effect. The address and data are not multiplexed. The AD7–AD0 signals have only data on the bus, while the AD bus has the address during a read or write operation.

Figure 3-1 16-Bit Mode—Normal Read and Write Operation



Note: *For a detailed description of DRAM control signals, see DRAM switching characteristics beginning on page 70.*

Figure 3-2 16-Bit Mode—Read and Write with Address Bus Disable in Effect



Note: *For a detailed description of DRAM control signals, see DRAM switching characteristics beginning on page 70.*

Figure 3-3 8-Bit Mode—Normal Read and Write Operation

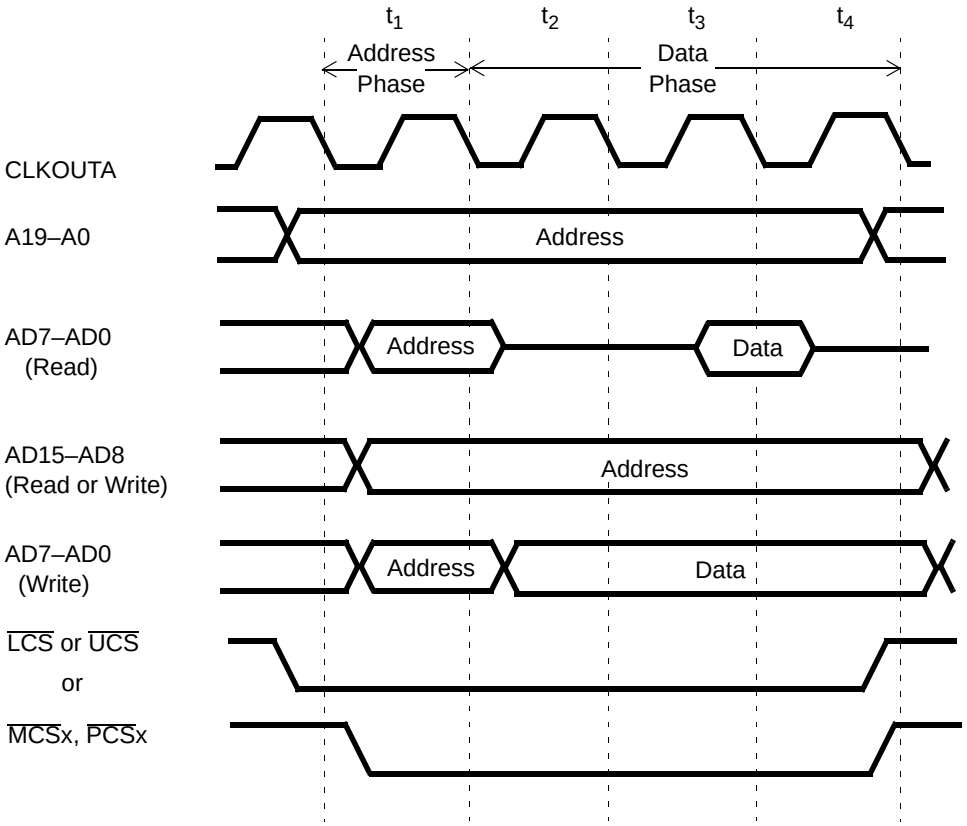
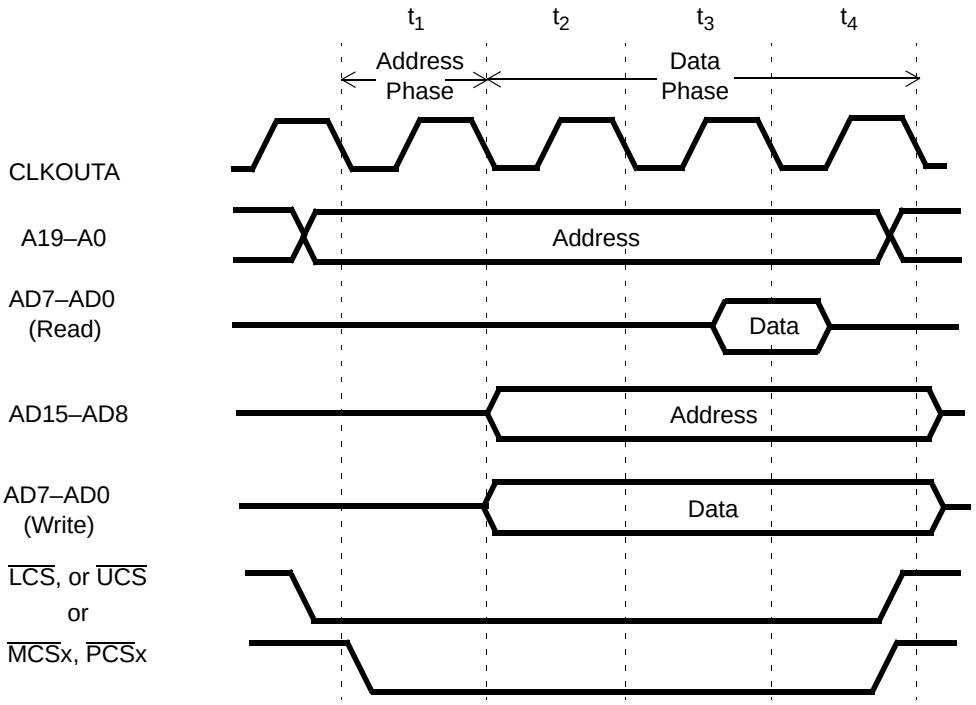


Figure 3-4 8-Bit Mode—Read and Write with Address Bus Disable in Effect



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 Am186ED/EDLV microcontrollers provide an enhanced bus interface unit with the following features:

- A nonmultiplexed address bus
- A static bus sizing option for 8-bit and 16-bit memory and I/O
- Separate byte write enables and $\overline{\text{CAS}}$ for high and low bytes
- A fully integrated DRAM controller
- Data strobe bus interface option

The standard 80C186/188 microcontroller multiplexed address and data bus requires system-interface logic and an external address latch. On the Am186ED/EDLV microcontrollers new byte write enables, DRAM 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 *Am186ED/EDLV Microcontrollers Data Sheet*, order# 21336.

3.3.1 Bus Mastering

When an external bus master requests control of the local bus (by asserting HOLD), the microcontroller completes the bus cycle in progress. It then relinquishes control of the bus to the external bus master by asserting HLDA and floating $\overline{\text{DEN}}$, $\overline{\text{RD}}$, $\overline{\text{WR}}$, $\overline{\text{S}}2$–$\overline{\text{S}}0$, AD15–AD0, S6, A19–A0, $\overline{\text{BHE}}$, $\overline{\text{WHB}}$, $\overline{\text{WLB}}$, and DT/$\overline{\text{R}}$. Internal pullups are active for all chip selects during HOLD. An internal pulldown in active for ALE during HOLD.

3.3.2 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, DRAM, and Flash/EPROM memory systems.

3.3.3 Static Bus Sizing

The 80C186 microcontroller provided a 16-bit wide data bus over its entire address range, memory, and I/O, but did not allow accesses to an 8-bit wide bus. However, the Am186ED/EDLV microcontrollers differ from the 80C186 microcontroller in allowing programmability for data bus widths through fields in the auxiliary configuration (AUXCON) register, as shown in Table 3-3.

The width of the data access should not be modified while the processor is fetching instructions from the associated address space or while the peripheral control block is overlaid on the affected address space.

Table 3-3 Programming Am186ED/EDLV Microcontrollers Bus Width

Space	AUXCON Field	Value	Bus Width	Comments
$\overline{UCS}$	USIZ	0	16 bits	Dependent on boot option[1]
		1	8 bits	
$\overline{LCS}$	LSIZ	0	16 bits	Default
		1	8 bits	
I/O	IOSIZ	0	16 bits	Default
		1	8 bits	
Other	MSIZ	0	16 bits	Default
		1	8 bits	

Note:

[1] $\overline{UCS}$ width on reset is determined by the $\overline{S2}$/BTSEL pin. If $\overline{UCS}$ boots as a 16-bit-space, it is not reconfigurable to 8-bit.

3.3.4 Byte Write Enables

The Am186ED/EDLV microcontrollers provide two signals that act as byte write enables—$\overline{WHB}$ (Write High Byte, AD15–AD8) and $\overline{WLB}$ (Write Low Byte, AD7–AD0). $\overline{WHB}$ is the logical OR of $\overline{BHE}$ and $\overline{WR}$ ($\overline{WHB}$ is Low when both $\overline{BHE}$ and $\overline{WR}$ are Low). $\overline{WLB}$ is the logical OR of A0 and $\overline{WR}$ ($\overline{WLB}$ is Low when both A0 and $\overline{WR}$ are both Low).

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

3.3.5 DRAM Controller

The key integration feature of the Am186ED/EDLV microcontrollers is a fully integrated DRAM controller. The Am186ED/EDLV microcontrollers are an extension of the features found in the Am186ES microcontroller. The DRAM controller has a glueless interface to 50-ns DRAM at 40 MHz for no-wait state operation and supports 50-ns, 60-ns and 70-ns DRAM.

- Multiplexed addresses for DRAM row and column accesses
- 8-bit and 16-bit boot mode for $\overline{UCS}$ accesses
- Two $\overline{RAS}$ signals that support two banks of DRAM
- Two byte $\overline{CAS}$ signals
- Directly supports 4 Mbit (256Kx16) fast page mode and extended data out mode DRAMs
- Prioritized $\overline{PCS}$ over DRAM space accesses

3.4 CLOCK AND POWER MANAGEMENT UNIT

The clock and power management unit of the Am186ED/EDLV 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 Am186ED/EDLV 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. 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.

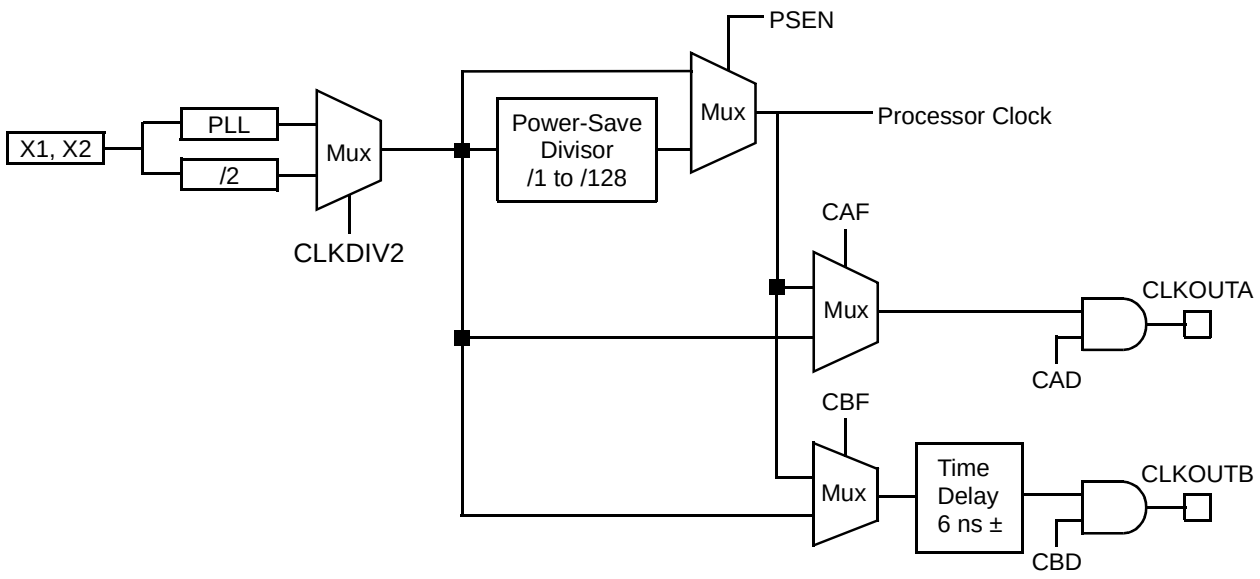
3.4.3 External Source Clock

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 3-5 shows the organization of the clocks. The original 80C186/188 microcontroller system clock has been renamed CLKOUTA. CLKOUTB is provided as an additional output.

Figure 3-5 Clock Organization



Note: *For frequencies under 16 MHz, use PLL bypass.*

CLKOUTA and CLKOUTB operate at either the fundamental 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) allows one clock to run at the PLL frequency and the other clock to 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. For an interrupt to be recognized, it must be valid before the internal clock's 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 integrated peripherals of the Am186ED/EDLV microcontrollers 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. Table 4-1 shows a map of the peripheral control block registers.

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 $\overline{RD}$, $\overline{WR}$, 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 Table 4-1. See Section 4.1.1 on page 4-3 for a complete description of the Peripheral Control Block Relocation register.

Table 4-1 Peripheral Control Block Register Map

Register Name	Offset	Page
Processor Control Registers: Chapters 4 and 6		
Peripheral control block relocation register	FEh	4-3
Reset configuration register	F6h	4-4
Processor release level register[1]	F4h	4-5
Auxiliary configuration register[1]	F2h	4-6
System configuration register	F0h	4-8
Watchdog timer control register	E6h	6-8
Enable RCU register[1]	E4h	6-7
Clock prescalar register[1]	E2h	6-6
See note 2 below.		
DMA Registers: Chapter 9		
DMA 1 control register	DAh	9-3
DMA 1 transfer count register	D8h	9-6
DMA 1 destination address high register	D6h	9-7
DMA 1 destination address low register	D4h	9-8
DMA 1 source address high register	D2h	9-9
DMA 1 source address low register	D0h	9-10
DMA 0 control register	CAh	9-3
DMA 0 transfer count register	C8h	9-6
DMA 0 destination address high register	C6h	9-9
DMA 0 destination address low register	C4h	9-8
DMA 0 source address high register	C2h	9-9
DMA 0 source address low register	C0h	9-10
Chip-Select Registers: Chapter 5		
$\overline{PCS}$ and $\overline{MCS}$ auxiliary register	A8h	5-10
Midrange memory chip-select register	A6h	5-8
Peripheral chip-select register	A4h	5-12
Lower memory chip-select register[1]	A2h	5-6
Upper memory chip-select register[1]	A0h	5-4
Serial Port 0 Registers: Chapter 10		
Serial port 0 baud rate divisor register	88h	10-14
Serial port 0 receive register	86h	10-13
Serial port 0 transmit register	84h	10-12
Serial port 0 status register	82h	10-10
Serial port 0 control register[1]	80h	10-5
PIO Registers: Chapter 11		
PIO data 1 register	7Ah	11-5
PIO direction 1 register	78h	11-4
PIO mode 1 register	76h	11-3
PIO data 0 register	74h	11-5
PIO direction 0 register	72h	11-4
PIO mode 0 register	70h	11-3

Register Name	Offset	Page
Timer Registers: Chapter 8		
Timer 2 mode/control register	66h	8-5
Timer 2 max count compare A register	62h	8-7
Timer 2 count register	60h	8-6
Timer 1 mode/control register	5Eh	8-3
Timer 1 max count compare B register	5Ch	8-7
Timer 1 max count compare A register	5Ah	8-7
Timer 1 count register	58h	8-6
Timer 0 mode/control register	56h	8-3
Timer 0 max count compare B register	54h	8-7
Timer 0 max count compare A register	52h	8-7
Timer 0 count register	50h	8-6
Interrupt Registers: Chapter 7		
Serial port 0 interrupt control register	44h	7-19
Serial port 1 interrupt control register	42h	7-19
INT4 interrupt control register	40h	7-17
INT3 control register	3Eh	7-16
INT2 control register	3Ch	7-16
INT1 control register	3Ah	7-14
INT0 control register	38h	7-14
DMA1/INT6 interrupt control register	36h	7-18
DMA0/INT5 interrupt control register	34h	7-18
Timer interrupt control register	32h	7-30
Interrupt status register	30h	7-20
Interrupt request register	2Eh	7-21
Interrupt in-service register	2Ch	7-23
Interrupt priority mask register	2Ah	7-24
Interrupt mask register	28h	7-25
Interrupt poll status register	26h	7-26
Interrupt poll register	24h	7-27
End-of-interrupt register	22h	7-28
Interrupt vector register	20h	7-37
Serial Port 1 Registers: Chapter 10		
Serial port 1 baud rate divisor register	18h	10-14
Serial port 1 receive register	16h	10-13
Serial port 1 transmit register	14h	10-12
Serial port 1 status register	12h	10-10
Serial port 1 control register[1]	10h	10-5

Notes:

[1] The register has been modified from the Am186ES/ Am188ES microcontrollers.

[2] The previous Memory Partition Register (MDRAM) has been removed and its functionality replaced with the $\overline{CAS}$-before-$\overline{RAS}$ refresh mode.

All unused addresses are reserved and should not be accessed.

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

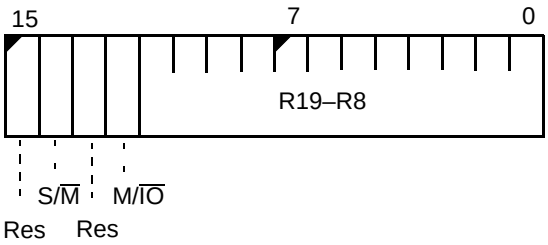
The peripheral control block is mapped into either memory or I/O space by programming the Peripheral Control Block Relocation (RELREG) register (see Figure 4-1). This register is a 16-bit register at offset FEh from the control block base address. The Peripheral Control Block Relocation register provides 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 Peripheral Control Block Relocation register contains a bit that places the interrupt controller into either slave mode or master mode.

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 Table 4-1.

Figure 4-1 Peripheral Control Block Relocation Register



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

Bit 15: Reserved—Set to 1.

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.

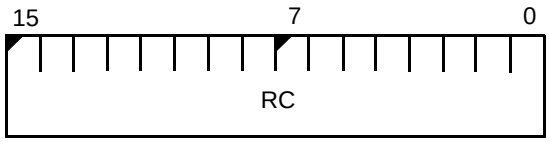
4.1.2 Reset Configuration Register (RESCON, Offset F6h)

The Reset Configuration (RESCON) register (see Figure 4-2) in the peripheral control block latches system-configuration information that is presented to the processor on the address/data bus (AD15–AD06) 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 Reset Configuration 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 Reset Configuration 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 4-2 Reset Configuration Register



The value of the RESCON register is system-dependent.

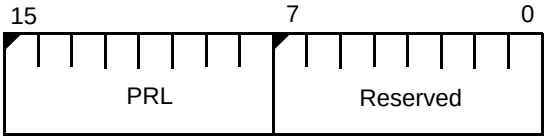
Bits 15–0: Reset Configuration (RC)—There is a one-to-one correspondence between address/data bus signals during the reset and the Reset Configuration register's bits. On the Am186ED/EDLV microcontrollers, AD15 corresponds to bit 15 of the Reset Configuration register, and so on. Once $\overline{\text{RES}}$ is deasserted, the Reset Configuration register holds its value. This value can be read by software to determine the configuration information.

The contents of the Reset Configuration 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 register (Figure 4-3) is a read-only register that specifies the processor version.

Figure 4-3 Processor Release Level Register



Bits 15–8: Processor Release Level (PRL)—This byte returns the current release level of the processor, as well as the identification of the family member.

Bits 7–0: Reserved

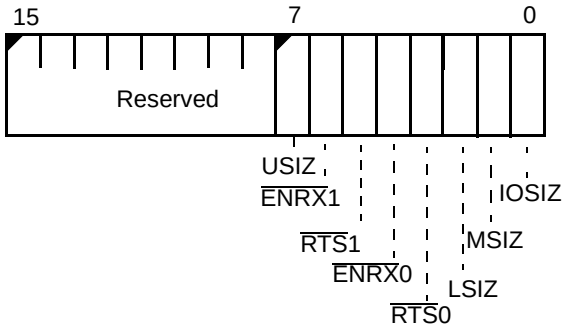
Table 4-2 Processor Release Level (PRL) Values

PRL Value	Processor Release Level
30h	A
31h	B

4.1.4 Auxiliary Configuration Register (AUXCON, Offset F2h)

The Auxiliary Configuration register is used to configure the asynchronous serial port flow-control signals and to configure the data bus width for memory and I/O accesses. The format of the Auxiliary Configuration register is shown in Figure 4-4.

Figure 4-4 Auxiliary Configuration Register



The reset value of this register is 0000h.

Bits 15–8: Reserved

Bit 7: $\overline{\text{UCS}}$ Data Bus Size (USIZ)—This bit determines the width of the bus for accesses to $\overline{\text{UCS}}$ space. If this bit is 1, 8-bit accesses are performed. If this bit is 0, 16-bit accesses are performed. This bit should not be modified while executing, reading or writing from $\overline{\text{UCS}}$ space. The value of this bit after processor reset is dependent on the $\overline{\text{S}}$2/BTSEL pin. During an external reset, the $\overline{\text{S}}$2/BTSEL pin is sampled. If $\overline{\text{S}}$2/BTSEL is high or floating, USIZ will be loaded with 0; if $\overline{\text{S}}$2/BTSEL is pulled low, USIZ will be loaded with 1. After a Watchdog Timer reset, USIZ will be loaded with the value that was sampled on $\overline{\text{S}}$2/BTSEL during the last external reset. $\overline{\text{S}}$2/BTSEL is normally an output used by emulators and should not be tied directly to ground. USIZ bit must be 0 (16-bit mode) when $\overline{\text{RAS}}$1 is enabled. Booting from 16-bit memory and switching to 8-bit memory is not supported. If USIZ is 0 after reset, then this bit is read-only.

Bit 6: Serial Port 1 Enable Receiver Request ($\overline{\text{ENRX}}$1)—When this bit is 1, the $\overline{\text{CTS}}$1/$\overline{\text{ENRX}}$1 pin is configured as $\overline{\text{ENRX}}$1. When this bit is 0, the $\overline{\text{CTS}}$1/$\overline{\text{ENRX}}$1 pin is configured as $\overline{\text{CTS}}$1. This bit is 0 after processor reset.

Bit 5: Serial Port 1 Request to Send ($\overline{\text{RTS}}$1)—When this bit is 1, the $\overline{\text{RTR}}$1/$\overline{\text{RTS}}$1 pin is configured as $\overline{\text{RTS}}$1. When this bit is 0, the $\overline{\text{RTR}}$1/$\overline{\text{RTS}}$1 pin is configured as $\overline{\text{RTR}}$1. This bit is 0 after processor reset.

Bit 4: Serial Port 0 Enable Receiver Request ($\overline{\text{ENRX}}$0)—When this bit is 1, the $\overline{\text{CTS}}$0/$\overline{\text{ENRX}}$0 pin is configured as $\overline{\text{ENRX}}$0. When this bit is 0, the $\overline{\text{CTS}}$0/$\overline{\text{ENRX}}$0 pin is configured as $\overline{\text{CTS}}$0. This bit is 0 after processor reset.

Bit 3: Serial Port 0 Request to Send ($\overline{\text{RTS}}$0)—When this bit is 1, the $\overline{\text{RTR}}$0/$\overline{\text{RTS}}$0 pin is configured as $\overline{\text{RTS}}$0. When this bit is 0, the $\overline{\text{RTR}}$0/$\overline{\text{RTS}}$0 pin is configured as $\overline{\text{RTR}}$0. This bit is 0 after processor reset.

Bit 2: $\overline{\text{LCS}}$ Data Bus Size (LSIZ)—This bit determines the width of the data bus for accesses to $\overline{\text{LCS}}$ space. If this bit is 1, 8-bit accesses are performed. If this bit is 0, 16-bit accesses are performed. This bit should not be modified while executing from $\overline{\text{LCS}}$ space or while the PCB is overlaid with $\overline{\text{LCS}}$ space. This bit is 0 after processor reset.

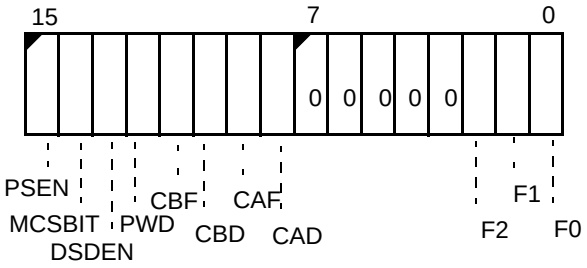
Bit 1: Midrange Data Bus Size (MSIZ)—This bit determines the width of the data bus for memory accesses which do not fall into the $\overline{\text{UCS}}$ or $\overline{\text{LCS}}$ address spaces, including $\overline{\text{MCS}}$ address space and $\overline{\text{PCS}}$ address space, if mapped to memory. If this bit is 1, 8-bit accesses are performed. If this bit is 0, 16-bit accesses are performed. This bit should not be modified while executing from the associated address space or while the PCB is overlaid on this address space. This bit is 0 after processor reset.

Bit 0: I/O Space Data Bus Size (IOSIZ)—This bit determines the width of the data bus for all I/O space accesses. If this bit is 1, 8-bit accesses are performed. If this bit is 0, 16-bit accesses are performed. This bit is 0 after processor reset. This bit should not be modified while the PCB is located in I/O space.

4.1.5 System Configuration Register (SYSCON, Offset F0h)

The format of the System Configuration register is shown in Figure 4-5.

Figure 4-5 System Configuration Register



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

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 occurs, including those interrupts generated by on-chip peripheral devices. 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.

Bit 14: MCS0 Only Mode Bit (MCSBIT)—This bit controls the $\overline{\text{MCS}}$0 only mode. When set to 0, the middle chip selects operate normally. When set to 1, $\overline{\text{MCS}}$0 is active over the entire $\overline{\text{MCS}}$ range. This bit is 0 after processor reset.

Bit 13: Data Strobe Mode of DEN Enable (DSDEN)—This bit enables the data strobe timings on the $\overline{\text{DEN}}$ pin. When this bit is set to 1, data strobe bus mode is enabled, and the $\overline{\text{DS}}$ timing for reads and writes is identical to the normal read cycle DEN timing. When this bit is set to 0, the $\overline{\text{DEN}}$ timing for both reads and writes is normal. The $\overline{\text{DEN}}$ pin is renamed $\overline{\text{DS}}$ in data strobe bus mode. This bit is 0 after processor reset.

During the bus cycle in which the DSDEN bit of the SYSCON register is written, the timing of the $\overline{\text{DEN}}$/$\overline{\text{DS}}$ pin is slightly different from normal. When a 1 is written to the DSDEN bit (which previously contained a 0), the falling edge of $\overline{\text{DEN}}$/$\overline{\text{DS}}$ occurs during PH2 of T_1 as it does during a normal write cycle, but the rising edge occurs during PH1 of T_4 in conformance with the data strobe timing. All writes after this have the normal data strobe timing until the DSDEN bit is reset.

When a 0 is written to the DSDEN bit (which previously contained a 1), the falling edge of $\overline{\text{DEN}}$/$\overline{\text{DS}}$ occurs during PH2 of T_2 as it does with the data strobe timing, but the rising edge occurs during PH2 of T_4 in conformance with normal write cycle timing. All writes after this have the normal write cycle timing until the DSDEN bit is set again.

Bit 12: Pulse Width Demodulation Mode Enable (PWD)—This bit enables pulse width demodulation mode. When this bit is set to 1, pulse width demodulation is enabled. When this bit is set to 0, pulse width demodulation is disabled. This bit is 0 after processor reset.

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). This bit is 0 after processor 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. This bit is 0 after processor 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). This bit is 0 after processor 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. This bit is 0 after processor 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. F2–-F0 is 000b after processor reset. 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 Am186ED/EDLV 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 4-3.

Table 4-3 Initial Register State After Reset

Register Name	Mnemonic	Value at Reset	Comments
Processor Status Flags	FLAGS	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)
Auxiliary Configuration	AUXCON	0000h	
System Configuration	SYSCON	0000h	
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 1 Control	SP1CT	0000h	Serial port interrupts disabled, no loopback, no break, BRKVAL low, no parity, word length = 7, 1 stop bit, transmitter and receiver disabled
Serial Port 0 Control	SP0CT	0000h	Serial port interrupts disabled, no loopback, no break, BRKVAL low, no parity, word length = 7, 1 stop bit, transmitter and receiver disabled
Serial Port 0 Baud Rate Divisor	SP0BAUD	0000h	
Serial Port 1 Baud Rate Divisor	SP1BAUD	0000h	
PIO Direction 1	PIODIR1	FFFFh	
PIO Mode 1	PIOMODE1	0000h	
PIO Direction 0	PIODIR0	FC0Fh	
PIO Mode 0	PIOMODE0	0000h	
Timer 2 Mode/Control	T2CON	0000h	
Timer 1 Mode/Control	T1CON	0000h	
Timer 0 Mode/Control	T0CON	0000h	
Serial Port 1 Interrupt Control	SP1CON	001Fh	Serial port interrupt masked, priority 7
Serial Port 0 Interrupt Control	SP0CON	001Fh	Serial port 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

Register Name	Mnemonic	Value at Reset	Comments
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/INT6	DMA1CON	000Fh	DMA1 interrupts masked, edge-triggered, priority 7
DMA0 Interrupt Control/INT5	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)
DMA 1 Control	D1CON	Undefined	Undefined at reset, except ST is set to 0
DMA 0 Control	D0CON	Undefined	Undefined at reset, except ST is set to 0
Watchdog Timer Control	WDTCON	C080h	

Note:

Registers not listed in this table are undefined at reset.

CHAPTER

5 CHIP SELECT UNIT

5.1 OVERVIEW

The Am186ED/EDLV 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 Am186ED/EDLV 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 5.5.5 on page 5-12).

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

Note: *The SYSCON register contains the MCSBIT field which determines behavior of the $\overline{MCS}0$ chip select. The $\overline{PCS}4$ chip select is not implemented on the Am186ED/EDLV microcontrollers.*

Table 5-1 Chip Select Register Summary

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

Except for the $\overline{UCS}$ 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 written. The $\overline{LCS}$ chip select is activated when the LMCS register is written, the $\overline{MCS}$ chip selects are activated after both the MMCS and MPCS registers have been written, and the $\overline{PCS}$ chip selects are activated after both the PACS and MPCS registers have been written.

5.2 CHIP SELECT TIMING

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

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

5.3 READY AND WAIT-STATE PROGRAMMING

The Am186ED/EDLV 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{PCS}$3–$\overline{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 zero 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{PCS}$3–$\overline{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 cycle, the access is extended until ready is asserted, which is followed by one more wait state followed by t_4.

The ARDY signal on the Am186ED/EDLV microcontrollers is a true asynchronous ready signal. The ARDY pin accepts a rising edge that is asynchronous to CLKOUTA and is active High. If the falling edge of ARDY is not synchronized to CLKOUTA as specified, an additional clock period may be added. For more information on the ARDY pin and SRDY pin, see the *Am186ED/EDLV Microcontrollers Data Sheet*, order# 21336A.

5.4 CHIP SELECT OVERLAP

Although programming the various chip selects on the Am186ED/EDLV microcontrollers 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.

Overlapping $\overline{\text{PCS}}$ with DRAM is fully supported as long as the $\overline{\text{PCS}}$ chip selects are programmed for a greater or equal number of wait states than that of the DRAM.

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 $\overline{\text{LCS}}$ or $\overline{\text{UCS}}$ 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 $\overline{\text{MCS}}$ and $\overline{\text{PCS}}$ 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 write to the MMCS and MPCS for the $\overline{\text{MCS}}$ chip selects, or by a write to the PACS and MPCS registers for the $\overline{\text{PCS}}$ 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 $\overline{\text{PCS}}$4 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 $\overline{\text{PCS}}$4 address space must follow the rules for overlapping chip selects. The ready and wait-state logic for $\overline{\text{PCS}}$6–$\overline{\text{PCS}}$5 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 $\overline{\text{PCS}}$ in I/O space with $\overline{\text{LCS}}$ 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.

The $\overline{\text{PCS}}$ chip selects can overlap DRAM blocks with different wait states and without external or internal bus contention. The $\overline{\text{RAS}}$ signal will assert along with the appropriate $\overline{\text{PCS}}$ signal. The $\overline{\text{UCAS}}$ and $\overline{\text{LCAS}}$ signals will not assert, preventing the DRAM from writing erroneously or driving the data bus during a read. The $\overline{\text{PCS}}$ signal must have the same or higher number of wait states than the DRAM. The $\overline{\text{PCS}}$ bus width will be determined by the LSIZ or USIZ bits which must be programmed for 16-bit bus widths associated with the DRAM. Overlapping the $\overline{\text{PCS}}$ chip selects with DRAM will make inaccessible a 1791-byte block of the DRAM. In its place, the peripherals associated with the $\overline{\text{PCS}}$ can be accessed. This is especially useful when the entire memory space is used with two banks of DRAM or a bank of DRAM and a 512K Flash.

5.5 CHIP SELECT REGISTERS

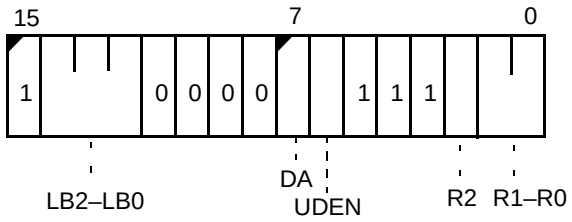
The following sections describe the chip select registers.

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

The Am186ED/EDLV 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, with external ready required and three wait states automatically inserted. The USIZ bit in the AUXCON register determines the bus width of the UCS address space. The USIZ bit holds the latched value of the inverse of the $\overline{S2}$/ BTSEL pin after external reset.

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



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 $\overline{UCS}$ or $\overline{RAS}$1 chip selects. The number of programmable bits has been reduced from the eight bits in the 80C186 microcontroller to three bits in the Am186ED/EDLV microcontrollers.

The Am186ED/EDLV microcontrollers provide an additional block size of 512K, which is not available on the 80C186 microcontroller. Table 5-2 outlines the possible configurations and differences with the 80C186 microcontroller.

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 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 $\overline{\text{UCS}}$ or $\overline{\text{RAS}}$1 is asserted. The AD bus is not affected by the state of the UDEN bit when the DA bit is set or cleared. If DA is set to 1, AD15–AD0 is not driven during the address phase of a bus cycle when $\overline{\text{UCS}}$ or $\overline{\text{RAS}}$1 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.

If $\overline{\text{BHE}}$/$\overline{\text{ADEN}}$ is held Low on the rising edge of $\overline{\text{RES}}$, 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 $\overline{\text{BHE}}$/$\overline{\text{ADEN}}$ is High on the rising edge of $\overline{\text{RES}}$, then the DA bit controls the AD15–AD0 disabling. $\overline{\text{BHE}}$/$\overline{\text{ADEN}}$ is internally pulled up.

See the description of the $\overline{\text{BHE}}$/$\overline{\text{ADEN}}$ pin in Chapter 3.

Bit 6: $\overline{\text{UCS}}$ DRAM Enable (UDEN)—The UDEN bit configures the $\overline{\text{UCS}}$ space as a DRAM bank. When UDEN is set to 1, the $\overline{\text{MCS}}$3 pin becomes $\overline{\text{RAS}}$1, and the $\overline{\text{MCS}}$1 and $\overline{\text{MCS}}$2 pins become $\overline{\text{UCAS}}$ and $\overline{\text{LCAS}}$ respectively.

The $\overline{\text{UCS}}$ pin is disabled when UDEN is set to 1. The system can then boot from a non-volatile memory using the $\overline{\text{UCS}}$ pin, and then switch $\overline{\text{UCS}}$ space to a DRAM bank after system initialization. The DA bit is still valid when UDEN is set. That is, even though the $\overline{\text{UCS}}$ pin is held high in DRAM mode, the address phase on AD15–AD0 will still be disabled during DRAM accesses to $\overline{\text{UCS}}$ space if DA is set to 1. If the block size programmed in LB2–LB0 does not match the size of the DRAM being used, then the full capacity of the DRAM will not be utilized.

Note: *The $\overline{\text{MCS}}$3–$\overline{\text{MCS}}$1 pins are multiplexed with programmable I/O pins. To enable their DRAM functionality, the PIO mode and PIO direction settings for the $\overline{\text{MCS}}$3–$\overline{\text{MCS}}$1 pins must be set to 0 for normal mode. For more information, see Chapter 11, Programmable I/O Pins.*

Bits 5–3: Reserved—Set to 1.

Bit 2: Ready Mode (R2)—The R2 bit is used to configure the ready mode for the $\overline{\text{UCS}}$ 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. DRAM is only supported in the ignore external ready condition (R2 = 1).

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 $\overline{\text{UCS}}$ memory area. From zero to three wait states can be inserted (R1–R0 = 00b to 11b). R1–R0 default to 11b at reset.

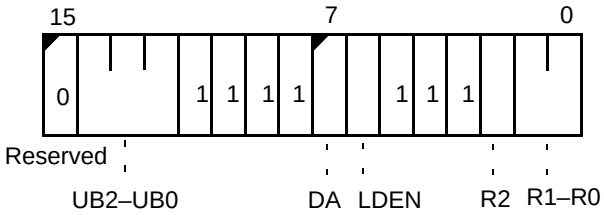
5.5.2 Lower Memory Chip Select Register (LMCS, Offset A2h)

The Am186ED/EDLV microcontrollers provide the $\overline{\text{LCS}}$ or $\overline{\text{RAS}}$0 chip select pin for the bottom of memory. Since the interrupt vector table is located at 00000h at the bottom of memory, the $\overline{\text{LCS}}$ pin has been provided to facilitate this usage. The $\overline{\text{LCS}}$ pin is not active on reset, but any write to the LMCS register activates this pin.

Before activating the $\overline{\text{LCS}}$ chip select, the width of the data bus for $\overline{\text{LCS}}$ space should be configured in the AUXCON register.

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

Figure 5-2 Lower Memory Chip Select Register



Bit 15: Reserved—Set to 0.

Bits 14–12: Upper Boundary (UB2–UB0)—The UB2–UB0 bits define the upper boundary of the memory accessed through the $\overline{\text{LCS}}$ or $\overline{\text{RAS}}$0 chip select. The number of programmable memory sizes for the LMCS register is reduced compared to the 80C186 microcontroller. Consequently, the number of programmable bits has been reduced from eight bits in the 80C186 microcontroller to three bits in the Am186ED/EDLV microcontrollers.

The Am186ED/EDLV microcontrollers have a block size of 512 Kbytes, which is not available on the 80C186 microcontroller. Table 5-3 outlines the possible configurations.

Table 5-3 LMCS Block Size Programming Values

Memory Block Size	Ending Address	UB2–UB0
64K	0FFFFh	000b
128K	1FFFFh	001b
256K	3FFFFh	011b
512K	7FFFFh	111b

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 $\overline{\text{LCS}}$ or $\overline{\text{RAS}}$0 is asserted. The AD bus is not affected by the state of the LDEN bit when the DA bit is set or cleared. If DA is set to 1, AD15–AD0 is not driven during the address phase of a bus cycle when $\overline{\text{LCS}}$ 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. This bit is 0 after processor reset.

If $\overline{\text{BHE}}$/$\overline{\text{ADEN}}$ is held Low on the rising edge of $\overline{\text{RES}}$, then AD15–AD0 is always driven regardless of the DA setting.

If $\overline{\text{BHE}}$/$\overline{\text{ADEN}}$ is High on the rising edge of $\overline{\text{RES}}$, then DA in the UMCS register and DA in the LMCS register control the AD15–AD0 disabling.

See the description of the $\overline{\text{BHE}}$/$\overline{\text{ADEN}}$ pin in Chapter 3.

Bit 6: $\overline{\text{LCS}}$ DRAM Enable (LDEN)—The LDEN bit selects the $\overline{\text{LCS}}$ space as a DRAM bank. When LDEN is set to 1, the $\overline{\text{LCS}}$ pin becomes $\overline{\text{RAS}}$0, and the $\overline{\text{MCS}}$1 and $\overline{\text{MCS}}$2 pins become $\overline{\text{UCAS}}$ and $\overline{\text{LCAS}}$ respectively.

The DA bit is still valid when the LDEN bit is set. That is, even though the $\overline{\text{LCS}}$ pin becomes $\overline{\text{RAS}}$0 in DRAM mode, the address phase on AD15–AD0 will still be disabled during DRAM accesses to $\overline{\text{LCS}}$ space if DA is set to 1. If the block size programmed in UB2–UB0 does not match the size of the DRAM being used, then the full capacity of the DRAM will not be utilized.

Note: *The $\overline{\text{MCS}}$3–$\overline{\text{MCS}}$1 pins are multiplexed with programmable I/O pins. To enable their DRAM functionality, the PIO mode and PIO direction settings for the $\overline{\text{MCS}}$3–$\overline{\text{MCS}}$1 pins must be set to 0 for normal mode. For more information, see Chapter 11, Programmable I/O Pins.*

Bits 5–3: Reserved—Set to 1.

Bit 2: Ready Mode (R2)—The R2 bit is used to configure the ready mode for the $\overline{\text{LCS}}$ 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. DRAM is only supported in the ignore external ready condition (R2 = 1).

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 $\overline{\text{LCS}}$ 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 Am186ED/EDLV 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.

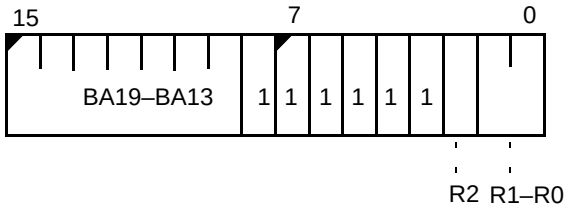
The Midrange Memory Chip Selects are programmed through two registers. The Midrange Memory Chip Select (MMCS) register (see Figure 5-3) determines the base address and the ready condition and wait states of the memory block accessed through the $\overline{\text{MCS}}$ pins. The $\overline{\text{PCS}}$ 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 write to activate these chip selects.

$\overline{\text{MCS}}$0 can be programmed to assert over the entire $\overline{\text{MCS}}$ address range through the MCSBIT in the System Configuration register. 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 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 $\overline{\text{LCS}}$.

Note: *The $\overline{\text{MCS}}$3–$\overline{\text{MCS}}$0 pins are multiplexed with programmable I/O pins. To enable the $\overline{\text{MCS}}$3–$\overline{\text{MCS}}$0 pins to function as chip selects, the PIO mode and PIO direction settings for the $\overline{\text{MCS}}$3–$\overline{\text{MCS}}$0 pins must be set to 0 for normal mode. For more information, see Chapter 11, Programmable I/O Pins.*

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

Figure 5-3 Midrange Memory Chip Select Register



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 $\overline{\text{LCS}}$ and $\overline{\text{RAS}}$0 chip select are not active. This is because 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 $\overline{\text{UCS}}$ or $\overline{\text{RAS}}$1 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 $\overline{\text{MCS}}$ 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 $\overline{\text{MCS}}$ memory area. From zero to three wait states can be inserted (R1–R0 = 00b to 11b).

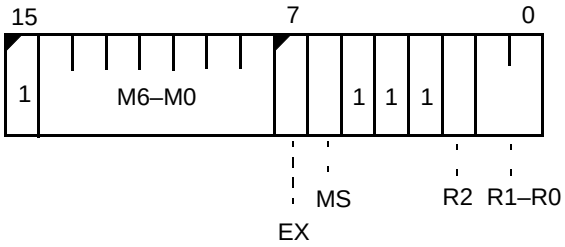
5.5.4 PCS and MCS Auxiliary Register (MPCS, Offset A8h)

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



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

If the MCSBIT in the SYSCON register is set, MCS0 asserts over the entire programmed block size. MCS3–MCS1 will continue to assert over their programmed range but are typically used as PIOs or as DRAM interface signals in this configuration.

Table 5-4 $\overline{\text{MCS}}$ 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{\text{PCS}}$6–$\overline{\text{PCS}}$5 pins are configured as chip selects or as alternate outputs for A2–A1. When this bit is set to 1, $\overline{\text{PCS}}$6–$\overline{\text{PCS}}$5 are configured as peripheral chip select pins. When EX is set to 0, $\overline{\text{PCS}}$5 becomes address bit A1 and $\overline{\text{PCS}}$6 becomes address bit A2.

Bit 6: Memory/ I/O Space Selector (MS)—This bit determines whether the $\overline{\text{PCS}}$ pins are active during memory bus cycles or I/O bus cycles. When MS is set to 1, the $\overline{\text{PCS}}$ outputs are active for memory bus cycles. When MS is set to 0, the $\overline{\text{PCS}}$ 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{\text{PCS}}$6–$\overline{\text{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{\text{PCS}}$6–$\overline{\text{PCS}}$5 chip selects. The value of R1–R0 determines the number of wait states inserted into an access to the $\overline{\text{PCS}}$ 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 microcontroller.

The Am186ED/EDLV 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 Am186ED/EDLV 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. $\overline{PCS}$ space may overlap either $\overline{RAS}$0 or $\overline{RAS}$1 DRAM 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.

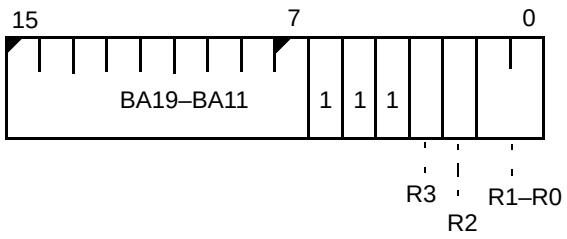
The $\overline{PCS}$ pins are not active on reset. The $\overline{PCS}$ pins are activated as chip selects by writing to the PACS and MPCS registers.

$\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.

Note: *The $\overline{MCS}$3–$\overline{MCS}$0 pins are multiplexed with programmable I/O pins. To enable the $\overline{MCS}$3–$\overline{MCS}$0 pins to function as chip selects, the PIO mode and PIO direction settings for the $\overline{MCS}$3–$\overline{MCS}$0 pins must be set to 0 for normal mode. For more information, see Chapter 11, Programmable I/O Pins.*

Figure 5-5 Peripheral Chip Select Register



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 original 80C186 microcontroller 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 Am186ED/EDLV microcontrollers.

When the $\overline{\text{PCS}}$ 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 DRAM CONTROL UNIT

6.1 OVERVIEW

The Am186ED/EDLV microcontrollers have a fully integrated DRAM controller. The Am186ED/EDLV microcontrollers provide glueless interface to 50-ns DRAM at 40 MHz for no-wait state operation. The Am186ED/EDLV microcontrollers support 50-ns, 60-ns, and 70-ns DRAM and multiplexed addresses for DRAM row and column accesses.

Note: *The PSRAM mode found on the Am186/188EM and Am186/188ES microcontrollers has been removed and replaced with a DRAM controller. This includes the variant PSRAM $\overline{LCS}$ timing and refresh strobe on $\overline{MCS}$3. The PSRAM mode bit in the LMCS register has been changed to the LDEN mode bit.*

All refresh cycles will contain three wait states to support 50-ns, 60-ns and 70-ns DRAMs at any frequency. A new bit is added to the UMCS register for the UDEN mode. When UDEN is enabled, the $\overline{UCS}$ pin will remain high and will not assert.

Two $\overline{RAS}$ signals support two banks of DRAM. Two $\overline{CAS}$ signals are also available for byte selection. The Am186ED/EDLV microcontrollers provide prioritized $\overline{PCS}$ over DRAM space accesses as well as 8-bit and 16-bit boot mode for $\overline{UCS}$ accesses. The Am186ED/EDLV microcontrollers directly support 4 Mbit (256K x 16) fast page mode and extended data out mode DRAMs.

The DRAM controller includes the $\overline{RAS}$0, $\overline{RAS}$1, $\overline{LCAS}$, $\overline{UCAS}$, $\overline{RD}$, $\overline{WR}$, and multiplexed address signals. The DRAM controller supports 50-ns, 60-ns, and 70-ns DRAMs with 0, 1, and 2 wait state operations respectively at 40 MHz. The DRAM controller will never perform a burst access. All accesses will be single accesses to DRAM. If the $\overline{PCS}$ chip selects are decoded to be in the DRAM address range, $\overline{PCS}$ accesses will take precedence over the DRAM.

6.2 DRAM OPERATION

The integrated DRAM controller directly interfaces DRAM to support a no-wait state DRAM interface up to 40 MHz. Internal wait states can be inserted to support slower DRAM; however, external ready detection is not supported. All signals required by the DRAM are generated on the Am186ED/EDLV microcontrollers and no external logic is required. The DRAM multiplexed address pins are connected to the odd address pins starting with A1 on the Am186ED/EDLV microcontrollers to MA0 on the DRAM. The correct row and column addresses are generated on these pins during a DRAM access. The $\overline{UCAS}$ and $\overline{LCAS}$ signals are used to select which byte of the DRAM is accessed during a read or write. The $\overline{RAS}$0 pin controls the lower bank of DRAM which starts at 00000h in the address map and is bounded by the lower memory size selected in the LMCS register. The $\overline{RAS}$1 pin controls the upper bank of DRAM which ends at FFFFFh and is bounded by the upper memory size in the UMCS register. When $\overline{RAS}$1 is enabled, $\overline{UCS}$ is automatically disabled. Neither, either, or both DRAM banks can be activated.

The user can re-enable $\overline{UCS}$ by clearing UDEN. Doing so will disable refreshing of the upper bank of DRAM. If the data in the upper bank of DRAM does not have to be retained, no special action is required. If the data in the upper bank of DRAM must be retained, two options are available. The refresh control unit counter can be monitored through the EDRAM

register. When the counter reaches all zeros, a refresh will occur. The user can then disable the upper bank of DRAM using the UDEN bit in the UCMS, access the $\overline{\text{UCS}}$-connected device, and then re-enable the upper bank of DRAM before the next refresh is scheduled to occur (usually 15.6 µs). This will retain the data in the upper bank of DRAM.

Alternatively, a software routine can conduct a read from all rows of the upper DRAM. Then the UDEN bit can be switched to enable $\overline{\text{UCS}}$ and disable $\overline{\text{RAS}}$1. The user then has the total refresh time (usually 16 ms) before the DRAM must be re-enabled so as to retain its data. After re-enabling the DRAM, the user should once again conduct reads on all the DRAM row addresses before letting the refresh controller resume refreshing the DRAM.

The $\overline{\text{PCS}}$0–$\overline{\text{PCS}}$6 signals can overlap DRAM blocks with different wait states and without external or internal bus contention. The $\overline{\text{RAS}}$0 or $\overline{\text{RAS}}$1 signals will assert along with the appropriate $\overline{\text{PCS}}$ signal. The $\overline{\text{UCAS}}$ and $\overline{\text{LCAS}}$ signals will not assert, preventing the DRAM from writing erroneously or driving the data bus during a read. The $\overline{\text{PCS}}$ signals must be configured to have the same or higher number of wait states than the DRAM. The bus width during $\overline{\text{PCS}}$ accesses is determined by the LSIZ or USIZ bus widths in the case of an overlap.

All DRAM space must be 16-bit sizes for upper and lower memory spaces.

The pin connections that should be made from the Am186ED/EDLV microcontrollers' address bus to the multiplexed DRAM address pins are in Table 6-1. The pin numbers for the PQFP and TQFP packages are provided below. The row address is the same as the non-muxed value that would be used in a non-muxed access (e.g., FLASH, SRAM).

The memory speed required to run with no-wait states depends on the operating frequency of the Am186ED/EDLV microcontrollers. Table 6-2 demonstrates the different access speed DRAMs and the impact of the wait state profile at different clock speeds.

Table 6-1 Am186ED/EDLV Microcontrollers DRAM Address Interface

Am186ED/ EDLV Pin	DRAM Pin	Row Address	Column Address	PQFP Pin Number	TQFP Pin Number
A1	MA0	A1	A2	39	62
A3	MA1	A3	A4	36	59
A5	MA2	A5	A6	34	57
A7	MA3	A7	A8	32	55
A9	MA4	A9	A10	30	53
A11	MA5	A11	A12	28	51
A13	MA6	A13	A14	26	49
A15	MA7	A15	A16	24	47
A17	MA8	A17	A18	22	45

Table 6-2 Microcontroller DRAM Access Speed

CPU Clock Speed	DRAM Speed	Wait States	Refresh Cycles
20MHz	70ns	0	7 clocks
25MHz	70ns	0	7 clocks
33MHz	60ns	0	7 clocks
	70ns	1	7 clocks
40MHz	50ns	0	7 clocks
	60ns	1	7 clocks
	70ns	2	7 clocks

6.2.1 Chip Select Modifications

The following chip selects in the Am186ED/EDLV microcontrollers have been modified: $\overline{\text{MCS}}$, $\overline{\text{PCS}}$, $\overline{\text{UCS}}$, and $\overline{\text{LCS}}$.

$\overline{\text{MCS}}$1, $\overline{\text{MCS}}$2 and $\overline{\text{MCS}}$3 have been modified as follows. $\overline{\text{MCS}}$1 becomes $\overline{\text{UCAS}}$ signal when the DRAM mode is enabled. The $\overline{\text{MCS}}$2 signal becomes $\overline{\text{LCAS}}$ when the DRAM mode is enabled. The $\overline{\text{MCS}}$3 signal can be enabled as the $\overline{\text{RAS}}$1 signal for an upper bank of DRAM. The $\overline{\text{RAS}}$1 signal will address DRAM from FFFFFh downward toward the 512 Kbyte range. When the $\overline{\text{MCS}}$0-only mode is enabled in the Am186ED/EDLV microcontrollers, the entire middle chip select range is selected through $\overline{\text{MCS}}$0. The remaining $\overline{\text{MCS}}$ pins are available for other functions.

Note: *The $\overline{\text{MCS}}$3–$\overline{\text{MCS}}$1 pins are multiplexed with programmable I/O pins. To enable their DRAM functionality, the PIO mode and PIO direction settings for the $\overline{\text{MCS}}$3–$\overline{\text{MCS}}$1 pins must be set to 0 for normal mode. For more information, see Chapter 11, Programmable I/O Pins.*

6.3 DRAM BANK CONFIGURATION

Figure 6-1 demonstrates a typical single bank DRAM system configuration with DRAM and other non-multiplexed devices such as FLASH, SRAM, EPROM and PROM. The upper bank of DRAM uses the $\overline{\text{RAS}}$1 signal. The $\overline{\text{UCS}}$ pin is held high when the upper DRAM bank is enabled.

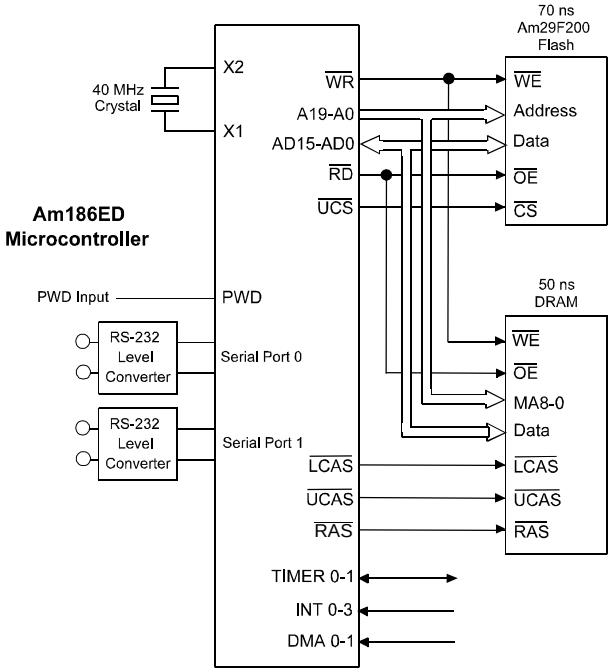
The $\overline{\text{RAS}}$0 and $\overline{\text{RAS}}$1 signals interface with two banks of DRAM. Depending on the $\overline{\text{UCS}}$ and $\overline{\text{UCS}}$ bank size, the $\overline{\text{RAS}}$0 and $\overline{\text{RAS}}$1 signals can support 64 Kbyte, 128 Kbyte, 256 Kbyte, and 512 Kbyte of memory decode.

The $\overline{\text{RAS}}$0 signal addresses the lower 512 Kbyte of memory space. The $\overline{\text{RAS}}$0 signal is multiplexed with the $\overline{\text{LCS}}$ signal. The $\overline{\text{RAS}}$0 signal will normally be used for the first or primary DRAM bank. The wait state configuration for $\overline{\text{RAS}}$0 is set in the LMCS register.

The $\overline{\text{RAS}}$1 signal interfaces with the upper 512 Kbyte of memory. The $\overline{\text{RAS}}$1 signal is multiplexed with the $\overline{\text{UCS}}$ signal. Do not switch from $\overline{\text{UCS}}$ to an upper DRAM space while executing from $\overline{\text{UCS}}$. The A17–A1 signals become multiplexed to directly interface with the DRAM. For $\overline{\text{RAS}}$1, the wait state configuration is set in the UMCS register.

$\overline{\text{CAS}}$0 asserts for even addresses. $\overline{\text{CAS}}$1 asserts for odd addresses.

Figure 6-1 Typical Am186ED Microcontroller System



6.4 DRAM CONTROLLER OPERATION

6.4.1 Option to Overlap DRAM with $\overline{\text{PCS}}$

The $\overline{\text{PCS}}$ signal has been modified to allow for overlap in memory space with the DRAM ($\overline{\text{RAS}}$0, $\overline{\text{RAS}}$1) space. Overlap of the $\overline{\text{PCS}}$ signal with $\overline{\text{UCS}}$, $\overline{\text{MCS}}$, or $\overline{\text{UCS}}$ in a non-DRAM mode is not recommended. If overlap of $\overline{\text{PCS}}$ with DRAM space occurs, the DRAM controller will assert $\overline{\text{RAS}}$ and stop the $\overline{\text{CAS}}$ signal from asserting. This will not modify the contents of the DRAM and the access will continue as a normal $\overline{\text{PCS}}$ access. When overlapping the $\overline{\text{PCS}}$ with DRAM, the number of wait states for $\overline{\text{PCS}}$ space must be equal to or greater than the number of wait states programmed for the DRAM. Table 6-3 demonstrates the possible valid and invalid configurations.

Table 6-3 Recommended Chip Select Overlap

	$\overline{\text{MCS}}$x	$\overline{\text{PCS}}$x	$\overline{\text{UCS}}$	$\overline{\text{UCS}}$
$\overline{\text{MCS}}$x	–	No	No	No
$\overline{\text{PCS}}$x	No	–	No	No
$\overline{\text{RAS}}$1	No	Yes	–	–
$\overline{\text{RAS}}$0	No	Yes	–	–

6.5 DRAM REFRESH CONTROL UNIT

The Refresh Control Unit (RCU) automatically generates refresh bus cycles. After a programmable period of time, the RCU generates a $\overline{\text{CAS}}$-before-$\overline{\text{RAS}}$ request to the bus interface unit. The RCU is fixed to three wait states for the DRAM 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 when the hold is negated. The circuit external bus master must negate the HOLD signal for at least one clock to allow the refresh cycle to execute.

6.5.1 DRAM Refresh

During a refresh cycle the AD bus will drive the address to FFFFh, which will prevent the $\overline{PCS}$ and $\overline{MCS}$ signals from asserting inadvertently. $\overline{PCS}$ and $\overline{MCS}$ decode should never contain the address FFFFFh. The $\overline{UCS}$ signal will not assert during a refresh cycle. If two banks of DRAM are being used in a system (i.e., $\overline{RAS}$0 and $\overline{RAS}$1) then both banks will be refreshed at the same time.

The interval counter (CDRAM register, offset E2h, and EDRAM register, offset E4h) is expanded by two bits. The refresh counter has a maximum timer count that will reach 51.2 microseconds at 40 MHz. See Table 6-4 and Equation 6-1.

The normal refresh rate on a DRAM is 15.6 µs. This refresh rate will allow for each of the 1024 row address to be refreshed in the required 16 ms. Some DRAMs might have different refresh rates for low power DRAMs and special considerations. Table 6-4 demonstrates the typical values that a programmer might want to use for refresh time intervals to be placed into the RC10–RC0 of the CDRAM register.

The Am186ED/EDLV microcontrollers will support DRAMs with a $\overline{CAS}$-before-$\overline{RAS}$ refreshing scheme. The Refresh Control Unit (RCU) has been modified to generate a refresh based on the system clock frequency. The maximum count value for the RCU is 51.2 microseconds at 40 MHz. The $\overline{CAS}$-before-$\overline{RAS}$ refresh cycle is seven clock cycles long. The RCU is an 11-bit counter that will insert a refresh bus cycle after the last bus cycle concludes to run the $\overline{CAS}$-before-$\overline{RAS}$ cycle.

Table 6-4 Refresh Interval Times

Frequency	CDRAM (hex value)	CDRAM (decimal value)	Refresh Interval Time
40 MHz	7FFh	2048	51.2 µs
	270h	624	15.6 µs
33 MHz	7FFh	2048	62.0 µs
	203h	515	15.6 µs
25 MHz	7FFh	2048	81.92 µs
	186h	390	15.6 µs
20 MHz	7FFh	2048	102.4 µs
	138h	312	15.6 µs

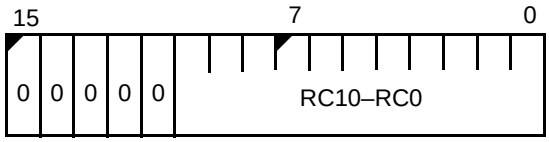
The equation to compute the refresh interval time is shown in Equation 6-1.

Equation 6-1 Refresh Interval Time Equation

Refresh Interval Time = Clock Period * CDRAM Counter Value

6.5.2 Clock Prescalar Register (CDRAM, Offset E2h)

Figure 6-2 Clock Prescalar Register



The CDRAM register is undefined on reset.

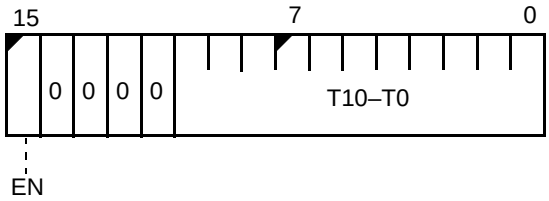
Bits 15–11: Reserved—Read back as 0.

Bits 10–0: Refresh Counter Reload Value (RC10–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.

6.5.3 Enable RCU Register (EDRAM, Offset E4h)

Figure 6-3 Enable RCU Register



The EDRAM register is set to 0000h on reset.

Bit 15: Enable RCU (EN)—Enables the refresh counter unit when set to 1. Clearing the EN bit at any time clears the refresh counter and stops refresh requests. This bit is 0 after processor reset.

Bits 14–11: Reserved—Read back as 0.

Bits 10–0: Refresh Count (T10–T0)—This read-only field contains the current value of the down counter that triggers refresh requests.

6.5.4 Watchdog Timer Control Register (WDTCON, Offset E6h)

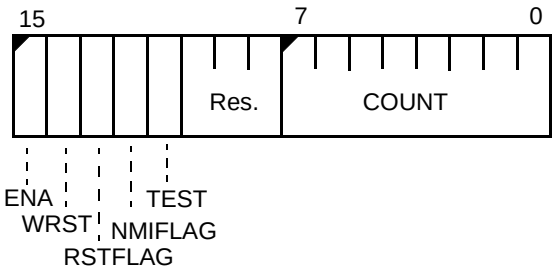
The Watchdog Timer Control register is a combined status and control register through which all watchdog timer functionality is implemented. The format of the watchdog timer control register is shown in Figure 6-4.

The watchdog timer (WDT) is enabled out of reset and configured to system reset mode with a maximum timeout count. The WDTCON register can be opened for a single write following reset. To open the WDTCON register for writing, the keyed sequence of 3333h followed by CCCCh must be written to the WDTCON register. The register can then be written with the new configuration. Any number of processor cycles, including memory and I/O reads and writes, can be inserted between the two halves of the key or between the key and the writing of the new configuration as long as they do not access the WDTCON register.

Note: *The Watchdog Timer (WDT) is active after reset.*

It is not possible to read the current count of the WDT; however, it can be reset by writing the keyed sequence of AAAAh followed by 5555h to the WDTCON register. Any number of processor cycles, including memory and I/O reads and writes, can be inserted between the two halves of the key as long as they do not access the WDTCON register. The current count should be reset before modifying the WDT timeout period to ensure that an immediate WDT timeout does not occur.

Figure 6-4 Watchdog Timer Control Register



The value of the WDTCON register at reset is C080h.

Bit 15: Watchdog Timer Enable (ENA)—When this bit is 1, the watchdog timer is enabled. When this bit is 0, the watchdog timer is disabled. This bit is 1 after processor reset.

Bit 14: Watchdog Reset (WRST)—When this bit is 1, the processor generates a WDT system reset when the WDT timeout count is reached. When this bit is 0, the processor generates an NMI interrupt when the WDT timeout count is reached if the NMIFLAG bit is 0. If the NMIFLAG bit is 1, a WDT system reset is generated upon WDT timeout. This bit is 1 after processor reset.

Bit 13: Reset Flag (RSTFLAG)—When this bit is 1, a watchdog timer reset event has occurred. This bit is cleared by any keyed read or write to this register or by an externally generated system reset. This bit is 0 after an external system reset or 1 after a WDT system reset.

Bit 12: NMI Flag (NMIFLAG)—When this bit is 1, a watchdog timer NMI event has occurred. This bit is cleared by any keyed write to this register. If this bit is set when a WDT timeout event occurs, a WDT system reset will be generated regardless of the setting of the WRST bit. This bit is 0 after processor reset.

Bit 11: Test Mode (TEST)—This bit is reserved for an internal test mode. Setting this bit activates a special test mode that generates early WDT timeouts. This bit is 0 after processor reset.

Bits 10–8: Reserved

Bits 7–0: WDT Timeout Count (COUNT)—This field determines the duration of the watchdog timer timeout interval. The duration is calculated using the following equation:

Duration = $2^{Exponent}$ / Frequency

Where Duration is the timeout period in seconds, Exponent is the value in the rightmost column of Table 6-6, determined by the programmed value of the COUNT field, and Frequency is the processor frequency in Hz. For example, the following calculation determines the WDT timeout period for a 40-MHz processor with the COUNT field set to 20h.

Duration = (2^{24} cycles) / (40,000,000 Hz)
= (16,777,216 cycles) / (40,000,000 cycles/second)
= .4194 seconds

Setting more than one bit in the COUNT field results in the shorter timeout value. This field is 80h after reset.

Table 6-5 Watchdog Timer COUNT Settings

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Exponent
0	0	0	0	0	0	0	0	N/A
X	X	X	X	X	X	X	1	10
X	X	X	X	X	X	1	0	20
X	X	X	X	X	1	0	0	21
X	X	X	X	1	0	0	0	22
X	X	X	1	0	0	0	0	23
X	X	1	0	0	0	0	0	24
X	1	0	0	0	0	0	0	25
1	0	0	0	0	0	0	0	26

Table 6-6 contains the watchdog timer durations for various watchdog timer COUNT settings and processor frequencies.

Table 6-6 Watchdog Timer Duration

Exponent	20 MHz	25 MHz	33 MHz	40 MHz
10	51 µs	40 µs	30 µs	25 µs
20	52 ms	41 ms	31 ms	26 ms
21	104 ms	83 ms	62 ms	52 ms
22	209 ms	167 ms	125 ms	104 ms
23	419 ms	335 ms	251 ms	209 ms
24	838 ms	671 ms	503 ms	419 ms
25	1.67 s	1.34 s	1.00 s	838 ms
26	3.35 s	2.68 s	2.01 s	1.67 s

CHAPTER 7

INTERRUPT CONTROL UNIT

7.1 OVERVIEW

The Am186ED/EDLV 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 up to eight external interrupt sources on the Am186ED/EDLV microcontrollers—seven maskable interrupt pins (INT6–INT0) and one nonmaskable interrupt (NMI) pin. There are eight internal interrupt sources that are not connected to external pins—three timers, two DMA channels, two asynchronous serial ports, and the watchdog timer NMI. INT5 and INT6 are multiplexed with DRQ0 and DRQ1 respectively. These two interrupts are available if the associated DMA is not enabled or is being used with internal synchronization.

The Am186ED/EDLV microcontrollers provide up to six interrupt sources that are not present on the 80C186 microcontroller:

- INT4, INT5, INT6; additional external interrupt pins that operate like the INT3–INT0 pins
- An internal, watchdog timer interrupt
- An internal interrupt from each of the two serial ports

The seven maskable interrupt request pins can be used as direct interrupt requests. INT4–INT0 can be either edge triggered or level triggered. INT6 and INT5 are edge triggered only. In addition, INT0 and INT1 can be configured in cascade mode for use with an external 82C59A-compatible interrupt controller. When INT0 is configured in cascade mode, the INT2 pin is automatically configured in its $\overline{\text{INTA}}$0 function. When INT1 is configured in cascade mode, the INT3 pin is automatically configured in its $\overline{\text{INTA}}$1 function. When programmed in cascade mode, INT0 must be programmed to a higher priority than INT1 whether INT1 is programmed in cascade mode or fully nested mode.

An external interrupt controller can be used as the system master by programming the internal interrupt controller to operate in slave mode using the PCB Relocation register. INT6–INT4 are not available in slave mode.

Maskable interrupts are automatically disabled when an interrupt is taken. Interrupt-service routines (ISRs) may re-enable interrupts by setting the IF flag. This allows interrupts of greater or equal priority to interrupt the currently executing ISR. Interrupts from the same source are disabled as long as the corresponding bit in the interrupt in-service register is set. INT1 and INT0 provide a special bit to enable special fully nested mode. When configured in special fully nested mode, the interrupt source may generate a new interrupt regardless of the setting of the in-service bit.

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 contains 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 two bits (multiplying by 4).

7.1.1.3 Maskable and Nonmaskable 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 nonmaskable. The nonmaskable interrupts are not affected by the setting of the IF flag.

The Am186ED/EDLV 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 nonmaskable 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 Am186ED/EDLV 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
Nonmaskable Interrupt (NMI)	02h	08h	N/A	1B		
Breakpoint Interrupt	03h	0Ch	N/A	1	INT3	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
Maskable Interrupts						
Timer 0 Interrupt	08h	20h	08h	2A		4, 5
Timer 1 Interrupt	12h	48h	08h	2B		4, 5
Timer 2 Interrupt	13h	4Ch	08h	2C		4, 5
Reserved for AMD Use	09h	24h				
DMA 0 Interrupt/INT5	0Ah	28h	0Ah	3		5
DMA 1 Interrupt/INT6	0Bh	2Ch	0Bh	4		5
INT0 Interrupt	0Ch	30h	0Ch	5		
INT1 Interrupt	0Dh	34h	0Dh	6		
INT2 Interrupt	0Eh	38h	0Eh	7		
INT3 Interrupt	0Fh	3Ch	0Fh	8		
INT4 Interrupt	10h	40h	10h	9		6
Asynchronous Serial Port 1 Interface	11h	42h	11h	9		6
Asynchronous Serial Port 0 Interrupt	14h	44h	14h	9		6
Reserved for AMD Use	15h–1Fh	54h–7Ch				

Notes:

Default priorities for the interrupt sources are used if the user does not reprogram priority levels.

1. *Interrupts generate as a result of an instruction execution.*
2. *Trace is performed in the same manner as 8086 and 8088.*
3. *An ESC opcode causes a trap.*
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 Nonmaskable Interrupts

Nonmaskable 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 Am186ED/EDLV microcontrollers. Nonmaskable 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 nonmaskable interrupts and software interrupts
- Interrupt priority for maskable hardware interrupts

7.1.3.1 Nonmaskable Interrupts and Software Interrupt Priority

The nonmaskable interrupts from 00h to 07h and software interrupts (INT instruction) always take priority over the maskable hardware interrupts. Within the nonmaskable and software interrupts, the trace interrupt has the highest priority, followed by the NMI interrupt, followed by the remaining nonmaskable 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 7-4 on page 7-19).

For example, if the INT6–INT0 interrupts are all changed to programmable priority six and no other programmable priorities are changed from the reset value of seven, then the INT6–INT0 interrupts take precedence over all other maskable interrupts. (Within INT6–INT0, the hierarchy is as follows: INT5>INT6>INT0>INT1>INT2>INT3>INT4.)

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, then setting the TF flag on the stack, and then popping the flags.

7.1.4.3 Nonmaskable Interrupt-NMI (Interrupt Type 02h)

This pin indicates to the microcontroller that an interrupt request has occurred. The NMI signal is the highest priority hardware interrupt and, unlike the INT6–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 does 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 not enable the maskable interrupts.

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 INT0 Detected Overflow Exception (Interrupt Type 04h)

Generated by an INT0 instruction if the OF bit is set in the Processor Status Flags (F) 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 Am186ED/EDLV 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 when INT0 or INT 1 is programmed in cascade mode, no interrupt acknowledge bus cycles are generated. The only external indication that an interrupt is being serviced is the processor reading the interrupt vector table.

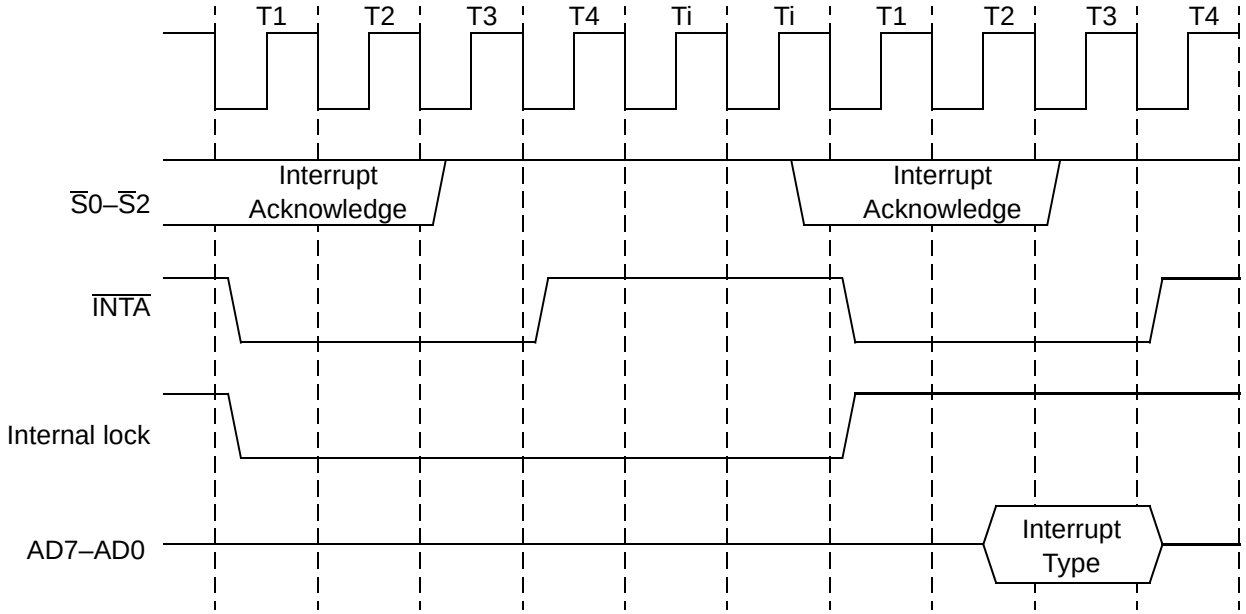
When an external interrupt controller is supplying the interrupt type, the processor generates two interrupt acknowledge bus cycles (see Figure 7-1). The interrupt type is written to the AD7–AD0 lines by the external interrupt controller during the second bus cycle.

When INTO0 is configured in cascade mode, it must be programmed to a higher priority than INT1 whether INT1 is programmed in cascade mode or fully nested mode.

Interrupt acknowledge bus cycles have the following characteristics:

- The two interrupt acknowledge cycles are locked.
- Two idle states are always inserted between the two interrupt acknowledge 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.$\overline{RD}$ 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 7-29 for a description of slave mode operation.

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

- Fully nested mode—seven interrupt lines with internally-generated interrupt types
- Cascade mode one—an interrupt line and interrupt acknowledge line pair with externally-generated interrupt types, plus five interrupt input lines with internally-generated types
- Cascade mode two—two pairs of interrupt and interrupt acknowledge lines with externally-generated interrupt types, and three interrupt input lines (INT6–INT4) with internally-generated type

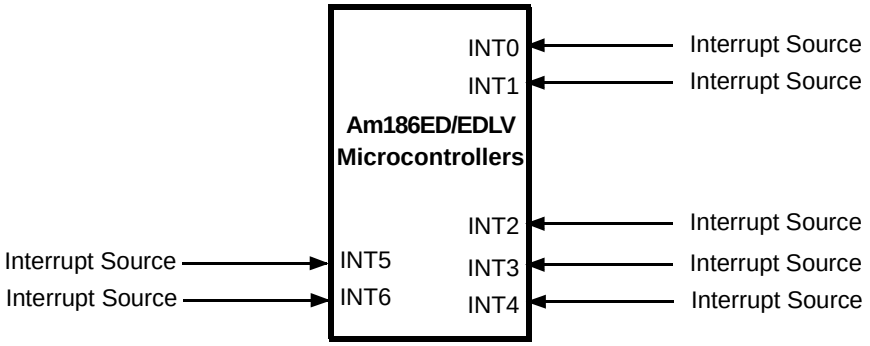
The basic modes of operation of the interrupt controller in master mode are similar to the 82C59A. The interrupt controller responds identically to internal interrupts in all three modes, the difference is only in the interpretation of function of the five external interrupt pins. The interrupt controller 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, seven pins are used as direct interrupt requests as in Figure 7-2. The interrupt types for these seven 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 high 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 EOI 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 Am186ED/EDLV microcontrollers have seven interrupt pins, two of which (INT2 and INT3) have dual functions. In fully nested mode, the seven pins are used as direct interrupt inputs and the corresponding interrupt types are generated internally. In cascade mode, four of the seven 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).

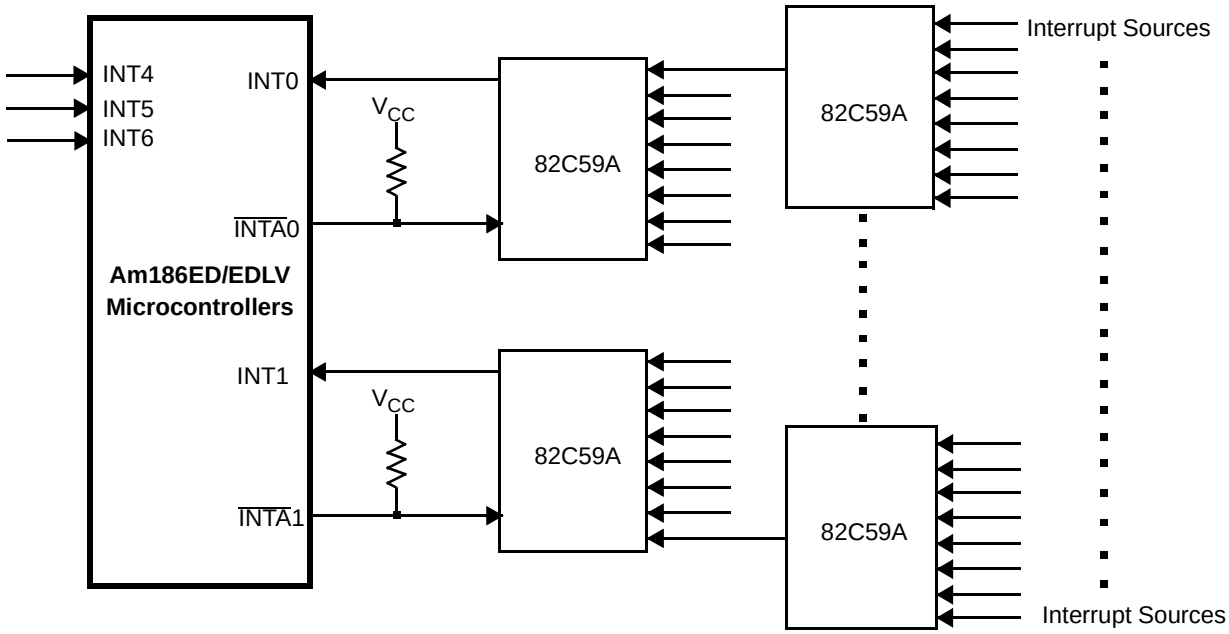
External sources in cascade mode use externally generated interrupt types. When an interrupt is acknowledged, two $\overline{\text{INTA}}$ cycles are initiated and the type is read into the microcontroller on the second cycle (see Section 7.1.5 on page 7-8). The capability to interface to one or two 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.

When INTO0 is configured in cascade mode, it must be programmed to a higher priority than INT1 whether INT1 is programmed in cascade mode or fully nested mode.

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

Special fully nested mode is entered by setting the SFNM bit in the INT0 or INT1 control registers. (See Section 7.3.1 on page 7-14.) 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 Am186ED/EDLV interrupt request pins. 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 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 Am186ED/EDLV 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

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 7-14). 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, which causes the in-service bit of the highest-priority source to be set.

To enable reading of the Poll register information without setting the indicated in-service bit, the Am186ED/EDLV microcontrollers provide a Poll Status register (Figure 7-14) 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.

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 7.3.13 on page 7-28).

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 7-29 for detailed information regarding slave mode register usage. On reset, the microcontroller is in master mode. Bit 14 of the Peripheral Control Block Relocation register (see Figure 4-1) 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
38h	I0CON	INT0 Control	INT0	
3Ah	I1CON	INT1 Control	INT1	
3Ch	I2CON	INT2 Control	INT2	
3Eh	I3CON	INT3 Control	INT3	
40h	I4CON	INT4 Control	INT4	
34h	DMA0CON	DMA0 Interrupt Control/INT5	INT5	
36h	DMA1CON	DMA1 Interrupt Control/INT6	INT6	
32h	TCUCON	Timer Interrupt Control	TMRIN1 TMRIN0 TMROUT1 TMROUT0	
44h	SP0CON	Serial Port 0 Interrupt Control		
42h	SP1CON	Serial Port 1 Interrupt Control		
30h	INTSTS	Interrupt Status		
2Eh	REQST	Interrupt Request	INT6–INT0	Read-only register
2Ch	INSERV	In-Service	INT6–INT0	
2Ah	PRIMSK	Priority Mask		
28h	IMASK	Interrupt Mask	INT6–INT0	
26h	POLLST	Poll Status		Read-only register
24h	POLL	Poll		Read-only register
22h	EOI	End of Interrupt		Write-only register

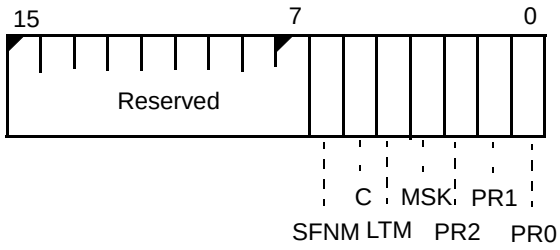
7.3.1 INT0 and INT1 Control Registers (I0CON, Offset 38h, I1CON, Offset 3Ah) (Master Mode)

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 INTA0, 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 INTA1, the interrupt acknowledge for INT1.

Figure 7-4 INT0 and INT1 Control Registers



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

Bits 15–7: Reserved—Set to 0.

Bit 6: Special Full y Nested Mode (SFNM)—When set to 1, enables special fully nested mode for INT0 or INT1.

Bit 5: Cascade Mode (C)—When set to 1, this bit enables cascade mode for INT0 or INT1.

When INTO0 is configured in cascade mode, it must be programmed to a higher priority than INT1 whether INT1 is programmed in cascade mode or fully nested 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 7.3.10 on page 7-25.

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 7-15.

Table 7-3 Priority Level

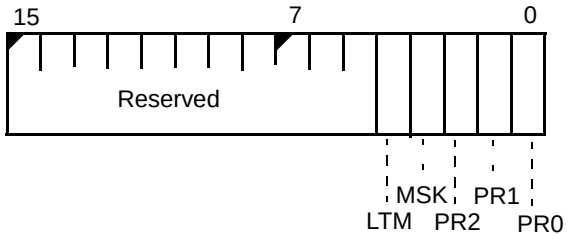
Priority	PR2–PR0
(High) 0	0 0 0b
1	0 0 1b
2	0 1 0b
3	0 1 1b
4	1 0 0b
5	1 0 1b
6	1 1 0b
(Low) 7	1 1 1b

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 INTA0 and INTA1 when cascade mode is implemented.

Figure 7-5 INT2 and INT3 Control Registers



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 7.3.10 on page 7-25.

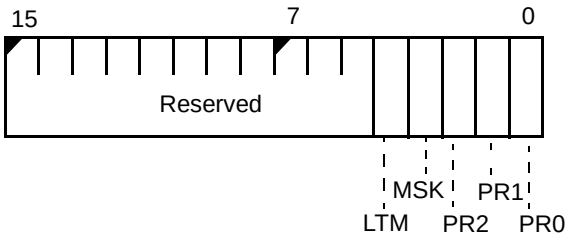
Bits 2–0: Priority Level (PR2–PR0)—This field determines the priority of INT2 or INT3 relative to the other interrupt signals, as shown in Table 7-4 on page 7-19.

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

The Am186ED/EDLV microcontrollers provide INT4, an additional external interrupt pin. This input behaves like INT3–INT0 on the 80C186 microcontroller with the exception that INT4 is only intended for use as a fully nested-mode interrupt source. INT4 is not available in cascade mode.

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



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.

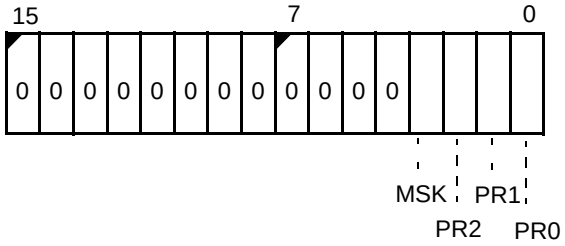
This bit is duplicated in the Interrupt Mask register. See the Interrupt Mask register in Section 7.3.10 on page 7-25.

Bits 2–0: Priority Level (PR)—This field determines the priority of INT4 relative to the other interrupt signals, as shown in Table 7-4 on page 7-19.

7.3.4 Timer and DMA Interrupt Control Registers (TCUCON, Offset 32h, DMA0CON/INT5CON, Offset 34h, DMA1CON/INT6CON, 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. See the DMA control registers for how to configure these pins as DMA requests or external interrupts.

Figure 7-7 Timer/DMA Interrupt Control Registers



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.

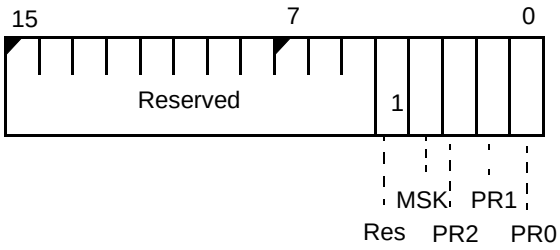
This bit is duplicated in the Interrupt Mask register. See the Interrupt Mask register in Section 7.3.10 on page 7-25.

Bits 2–0: Priority Level (PR2–PR0)—Sets the priority level for its corresponding source. See Table 7-4 on page 7-19.

7.3.5 Serial Port 0/1 Interrupt Control Registers (SP0CON/SP1CON, Offset 44h/42h) (Master Mode)

The serial port interrupt control registers control the operation of the serial ports' interrupt source (SP1 and SP0, bits 10–9 in the interrupt request register). Serial port 0 is assigned to interrupt type 14h and serial port 1 is assigned to interrupt type 11h. The control register format is shown in Figure 7-8.

Figure 7-8 Serial Port 0/1 Interrupt Control Register



The value of SP0CON and 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 7.3.10 on page 7-25.

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 7-4.

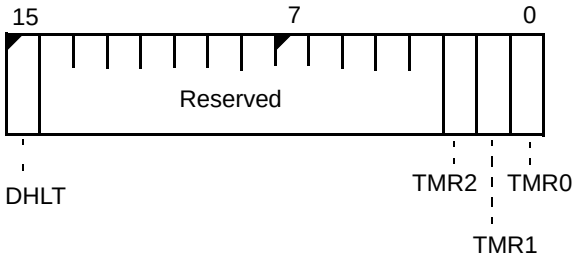
Table 7-4 Priority Level

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

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

The interrupt status register indicates the interrupt request status of the three timers.

Figure 7-9 Interrupt Status Register



Bit 15: DMA Halt (DHLT)—When set to 1, halts any DMA activity. This bit is automatically set to 1 when nonmaskable 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 logical OR of these timer interrupt requests.)

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

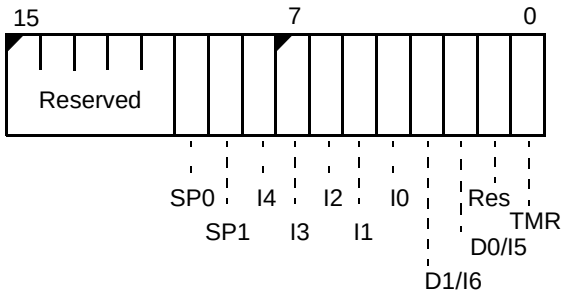
The hardware interrupt sources have interrupt request bits inside the interrupt controller. A read from this register yields the status of these bits. The Interrupt Request register is a read-only register. The format of the Interrupt Request register is shown in Figure 7-10.

For internal interrupts (SP0, SP1, D1/I6, D0/I5, 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 INT6–INT0 external interrupts, the corresponding bit (INT4–INT0) 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 7-10 Interrupt Request Register



The REQST register is undefined on reset.

Bits 15–11: Reserved

Bit 10: Serial Port 0 Interrupt Request (SP0)—This bit indicates the interrupt state of serial port 0. If enabled, the SP0 bit is the logical OR of all possible serial port interrupt sources (THRE, RDR, BRK1, BRK0, FER, PER, and OER status bits).

Bit 9: Serial Port 1 Interrupt Request (SP1)—This bit indicates the interrupt state of serial port 1. If enabled, the SP1 bit is the logical OR of all possible serial port interrupt sources (THRE, RDR, BRK1, BRK0, FER, PER, and OER status bits).

Bits 8–4: Interrupt Requests (INT4–INT0)—When set to 1, the corresponding INT pin has an interrupt pending (i.e., when INT0 is pending, INT0 is set).

Bit 3: DMA Channel 1/Interrupt 6 Request (D1/I6)—When set to 1, DMA channel 1 or INT6 has an interrupt pending.

Bit 2: DMA Channel 0/Interrupt 5 Request (D0/I5)—When set to 1, DMA channel 0 or INT5 has an interrupt pending.

Bit 1: Reserved

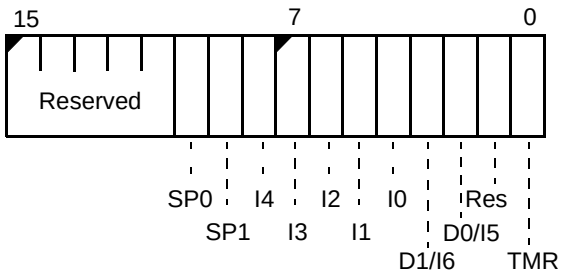
Bit 0: Timer Interrupt Request (TMR)—This bit indicates the state of the timer interrupts. This bit is the logical OR of the timer interrupt requests. When set to a 1, this bit indicates that the timer control unit has an interrupt pending.

The interrupt status register indicates the specific timer that is requesting an interrupt. See Section 7.3.6.

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

The bits in the In-Service 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.

Figure 7-11 Interrupt In-Service Register



The INSERV register is set to 0000h on reset.

Bits 15–11: Reserved

Bit 10: Serial Port 0 Interrupt In-Service (SP0)—This bit indicates the in-service state of serial port 0.

Bit 9: Serial Port 1 Interrupt In-Service (SP1)—This bit indicates the in-service state of the serial port 1.

Bits 8–4: Interrupt In-Service (INT4–INT0)—These bits indicate the in-service state of the corresponding INT pin.

Bit 3: DMA Channel 1/Interrupt 6 In-Service (D1/I6)—This bit indicates the in-service state of DMA channel 1 or INT6.

Bit 2: DMA Channel 0/Interrupt 5 In-Service (D0/I5)—This bit indicates the in-service state of DMA channel 0 or INT5.

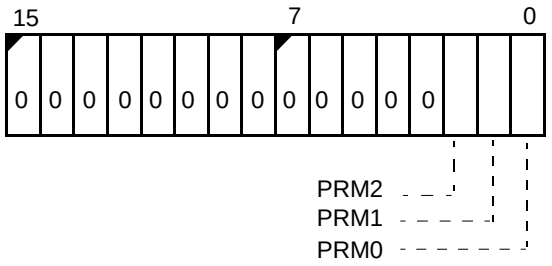
Bit 1: Reserved

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 requests. When set to a 1, this bit indicates that the corresponding timer interrupt request is in-service.

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

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

Figure 7-12 Priority Mask Register



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 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 7-5 Priority Level

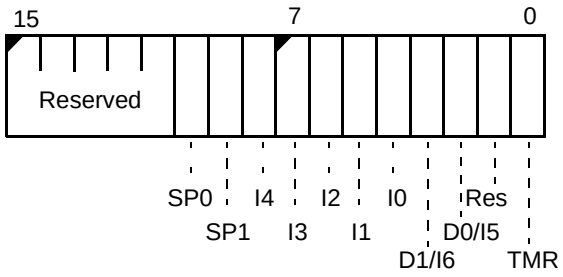
Priority	PR2–PR0
(High) 0	0 0 0b
1	0 0 1b
2	0 1 0b
3	0 1 1b
4	1 0 0b
5	1 0 1b
6	1 1 0b
(Low) 7	1 1 1b

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

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 interrupt control register. The format of the Interrupt Mask register is shown in Figure 7-13.

When a bit is set to 1 in this register, the corresponding interrupt source is masked off. When the bit is set to 0, the interrupt source is enabled to generate an interrupt request.

Figure 7-13 Interrupt Mask Register



The IMASK register is set to 07FDh on reset.

Bits 15–11: Reserved

Bit 10: Serial Port 0 Interrupt Mask (SP0)— When set to 1, this bit indicates that the serial port 0 interrupt is masked.

Bit 9: Serial Port 1 Interrupt Mask (SP1)—When set to 1, this bit indicates that the serial port 1 interrupt is masked.

Bits 8–4: Interrupt Mask (INT4–INT0)—When set to 1, an INT4–INT0 bit indicates that the corresponding interrupt is masked.

Bits 3–2: DMA Channel Interrupt Masks (D1/I6–D0/I5)—When set to 1, a D1/I6–D0/I5 bit indicates that the corresponding DMA or INT6/INT5 channel interrupt is masked.

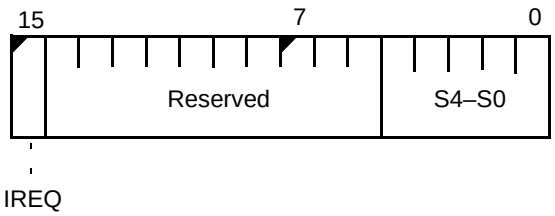
Bit 1: Reserved

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

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

The Poll Status register mirrors the current state of the Poll register. The Poll Status 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. This is a read-only register.

Figure 7-14 Poll Status Register



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.

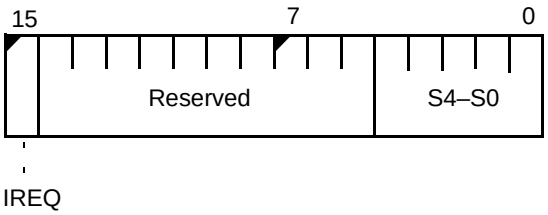
Bits 4–0: Poll Status (S4–S0)—Indicates the interrupt type of the highest priority pending interrupt (see Table 7-1 on page 7-4).

7.3.12 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. This is a read-only register.

Figure 7-15 Poll Register



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 (see Table 7-1). Reading the Poll register acknowledges the highest pending interrupt and allows 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.

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

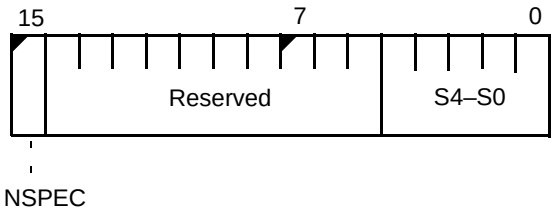
The End-of-Interrupt (EOI) register is a write-only register. The in-service flags in the In-Service register (see Section 7.3.8 on page 7-23) are reset by writing to the EOI register. Before executing the IRET instruction that ends an interrupt service routine (ISR), the ISR should write to the EOI 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 7-16 shows example code for a specific EOI reset. See Table 7-1 on page 7-4 for specific EOI values.

Figure 7-16 Example EOI Assembly Code

```
            ...
            ...                     ;ISR code
            ...
exit:       mov ax,int_type         ;load the interrupt type in ax
            mov dx, 0ff22h          ;write the interrupt type to EOI
            out dx,ax
            popa
            iret                    ;return from interrupt
```

Figure 7-17 End-of-Interrupt Register



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 is in S4–S0.

Bits 14–5: Reserved

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

7.4 SLAVE MODE OPERATION

When slave mode is used, the internal microcontroller 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 4-1 on page 4-3).

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 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 7-6. All registers can be read and written, unless specified otherwise.

Table 7-6 Interrupt Controller Registers in Slave Mode

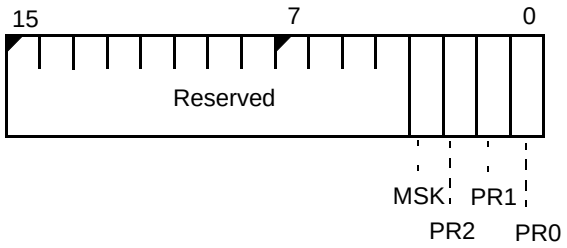
Offset	Register Mnemonic	Register Name	Affected Pins	Comments
3Ah	T2INTCON	Timer 2 Interrupt Control		Interrupt Type XXXXX101
38h	T1INTCON	Timer 1 Interrupt Control	TMRIN1 TMROUT1	Interrupt Type XXXXX100
36h	DMA1CON	DMA 1 Interrupt Control/INT6	DRQ1/INT6	Interrupt Type XXXXX011
34h	DMA0CON	DMA 0 Interrupt Control/INT5	DRQ0/INT5	Interrupt Type XXXXX010
32h	T0INTCON	Timer 0 Interrupt Control	TMRIN0 TMROUT0	Interrupt Type XXXXX000
30h	INTSTS	Interrupt Status		
2Eh	REQST	Interrupt Request		
2Ch	INSERV	In-Service		
2Ah	PRIMSK	Priority Mask		
28h	IMASK	Interrupt Mask		
22h	EOI	Specific EOI		
20h	INTVEC	Interrupt Vector		

7.4.3 Timer and DMA Interrupt Control Registers (T0INTCON, Offset 32h, T1INTCON, Offset 38h, T2INTCON, Offset 3Ah, DMA0CON/INT5, Offset 34h, DMA1CON/INT6, 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 7-18 Timer and DMA Interrupt Control Registers



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.

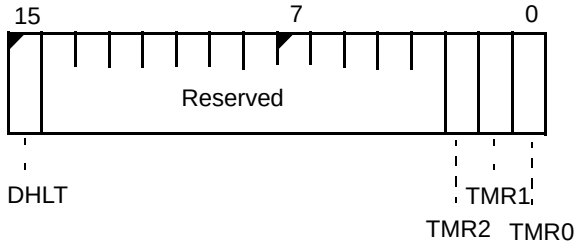
This bit is duplicated in the Interrupt Mask register. See the Interrupt Mask register in Section 7.4.8 on page 7-35.

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 7-4 on page 7-19.

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

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

Figure 7-19 Interrupt Status Register



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 nonmaskable interrupts occur and reset when an IRET instruction is executed.

Bits 14–3: Reserved

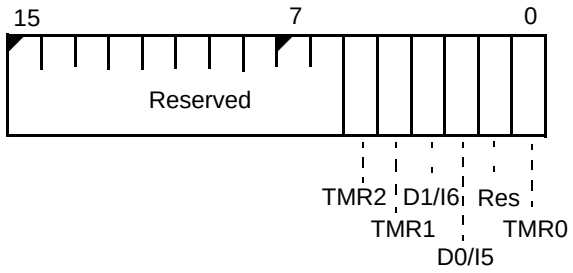
Bits 2–0: Timer Interrupt Request (TMR2–TMR0)—When set to 1, indicates the corresponding timer has an interrupt request pending.

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 7-20.

For internal interrupts (D1/I6, D0/I5, 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 7-20 Interrupt Request Register



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/I6–D0/I5)—When set to 1, D1/I6–D0/I5 indicate that the corresponding DMA channel or INT5/INT6 has an interrupt pending.

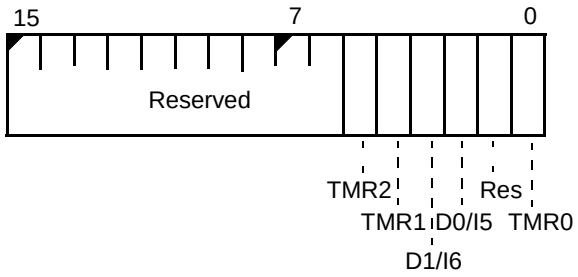
Bit 1: Reserved

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

The format of the In-Service register is shown in Figure 7-21. 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 7-21 Interrupt In-Service Register



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/I6–D0/I5)—When set to 1, the corresponding DMA channel INT5/INT6 is currently being serviced.

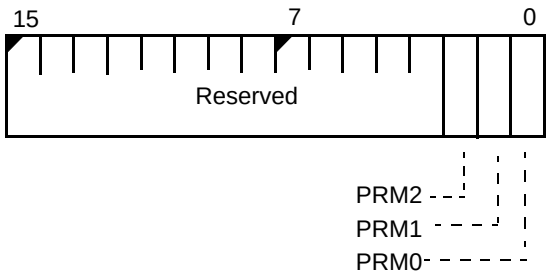
Bit 1: Reserved

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 7-22. The Priority Mask register provides the value that determines the minimum priority level at which maskable interrupts can generate an interrupt.

Figure 7-22 Priority Mask Register



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 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.

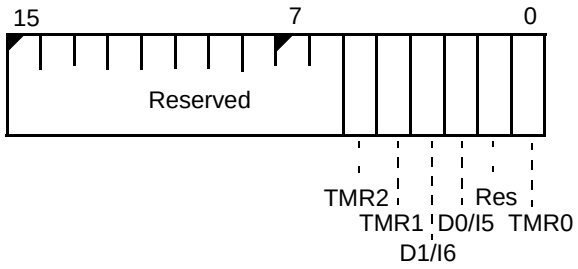
Table 7-7 Priority Level

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 7-23. 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 7-23 Interrupt Mask Register



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/I6–D0/I5)—These bits indicate the state of the mask bits of the corresponding DMA channel INT5/INT6 control register.

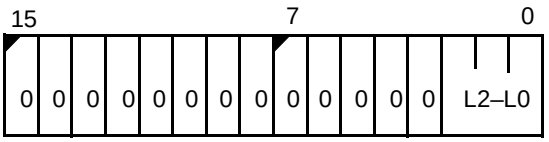
Bit 1: Reserved

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 7-24 Specific End-of-Interrupt Register



The EOI register is undefined on reset.

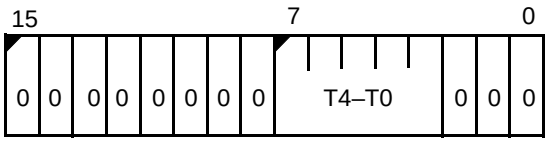
Bits 15–3: Reserved—Write as 0.

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. This is a 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 7-25 Interrupt Vector Register



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 7-6 on page 7-29.

Bits 2–0: Reserved—Read as 0.

CHAPTER

8 TIMER CONTROL UNIT

8.1 OVERVIEW

There are three 16-bit programmable timers in the Am186ED/EDLV 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 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.

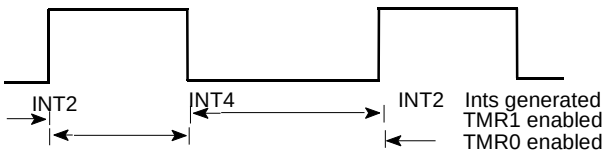
8.2 PULSE WIDTH DEMODULATION

For many applications, such as bar-code reading, it is necessary to measure the width of a signal in both its High and Low phases. The Am186ED/EDLV microcontrollers provide a pulse-width demodulation (PWD) option to fulfill this need. The PWD bit in the system configuration register (SYSCON) enables the PWD option. Please note that the Am186ED/EDLV microcontrollers do not support analog-to-digital conversion.

In PWD mode, TMRIN0, TMRIN1, INT2, and INT4 are configured internal to the microcontroller to support the detection of rising and falling edges on the PWD input pin (INT2/$\overline{\text{INTA}}$0/PWD) and to enable either timer 0 when the signal is High or timer 1 when the signal is Low. The INT4, TMRIN0, and TMRIN1 pins are not used in PWD mode and so are available for use as PIOs.

The following diagram shows the behavior of a system for a typical waveform.

Figure 8-1 Typical Waveform Behavior



The interrupt service routine (ISR) for the INT2 and INT4 interrupts should examine the current count of the associated timer, timer 1 for INT2 and timer 0 for INT4, in order to determine the pulse width. The ISR should then reset the timer count register in preparation for the next pulse.

Since the timers count at one quarter of the processor clock rate, this determines the maximum resolution that can be obtained. Further, in applications where the pulse width may be short, it may be necessary to poll the INT2 and INT4 request bits in the interrupt request register in order to avoid the overhead involved in taking and returning from an interrupt. Overflow conditions, where the pulse width is greater than the maximum count of the timer, can be detected by monitoring the Maximum Count (MC) bit in the associated timer or by setting the INT bit to enable timer interrupt requests.

8.3 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 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. For example, a 50% duty cycle waveform can be generated at 1/8 the frequency of the system clock using a 1h value for maxcount A and maxcount B.

8.3.1 Timer Operating Frequency

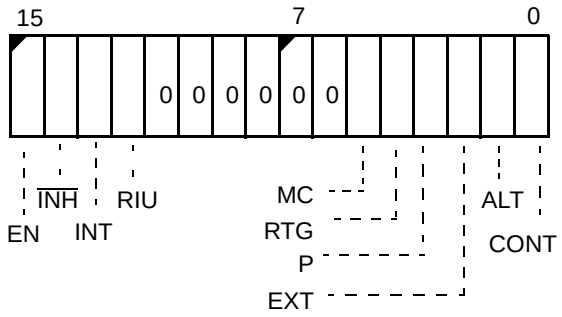
Each timer is serviced on every fourth clock cycle. Therefore, a timer can operate at a maximum speed of one-quarter of the internal clock frequency. A timer can be clocked externally at the same maximum frequency of one-fourth of the internal clock frequency. However, because of internal synchronization 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 internal clock. If power-save mode is in effect, the timers operate at the reduced power-save clock rate.

8.3.2 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 8-2.

Figure 8-2 Timer 0 and Timer 1 Mode and Control Registers



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: Prescalar 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. When the internal clock is used, the timer input pin is available for use as a programmable I/O pin.

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.

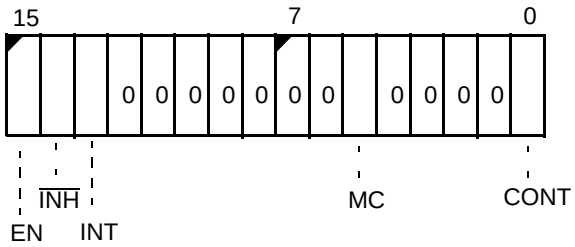
When CONT is set to 0, EN is cleared after each timer count sequence and the timer clears and then halts on reaching the maximum count. If CONT=0 and ALT=1, the timer counts to the maxcount compare A register value and resets, then it counts to the B register value and resets and halts.

Note: *The TMRIN0, TMRIN1, TMROUT0, and TMROUT1 pins are multiplexed with programmable I/O pins. To enable the timer pin functionality, the PIO mode and PIO direction settings for these pins need to be set to 0 for normal operation. For more information, see Chapter 11, Programmable I/O Pins.*

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

This register controls the functionality of timer 2. See Figure 8-3.

Figure 8-3 Timer 2 Mode and Control Register



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. This bit cannot be written to unless the $\overline{\text{INH}}$ bit is set to 1 during the same write.

Bit 14: Inhibit Bit ($\overline{\text{INH}}$)—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.

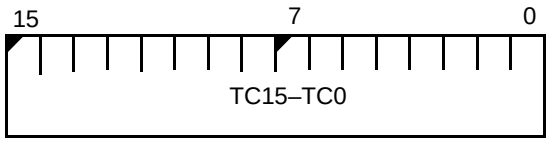
Bit 0: Continuous Mode Bit (CONT)—When CONT is set to 1, it causes the associated timer to run continuously. When set to 0, EN is cleared after each timer count sequence and the timer halts on reaching the maximum count.

8.3.4 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 8-4.

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

Figure 8-4 Timer Count Registers



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.

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

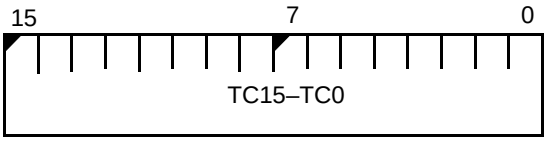
These registers serve as comparators for their associated count registers. Timer 0 and timer 1 each have two maximum count compare registers. See Figure 8-5.

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 wave forms 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 8-5 Timer Maxcount Compare Registers



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 Am186ED/EDLV microcontrollers provide 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 Am186ED/EDLV microcontrollers. Word transfers are only supported when both the source and destination are configured at 16-bit address spaces. Two bus cycles (a minimum of eight clocks) are necessary for each data transfer.

Each channel accepts a DMA request from one of three sources: a channel request pin (DRQ1–DRQ0), timer 2, or an asynchronous serial port. 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.

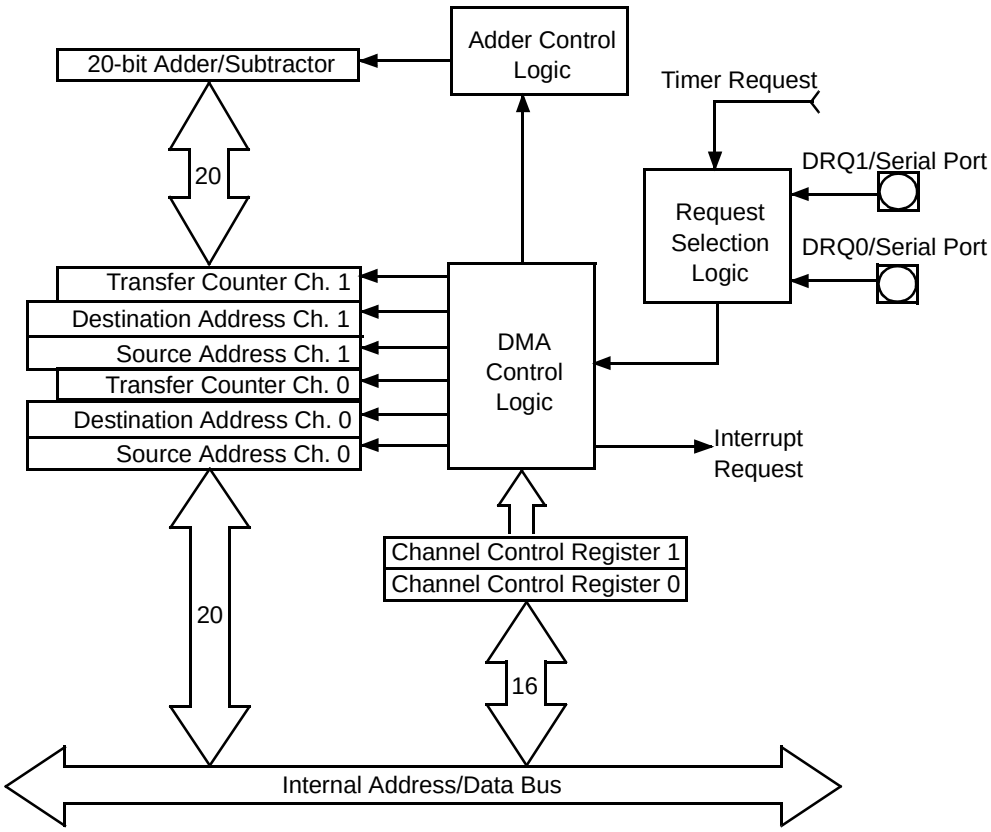
Table 9-1 DMA Controller Register Summary

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. On the Am186ED/EDLV microcontrollers, 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 following sections describe the control registers that are used to configure and operate the two DMA channels.

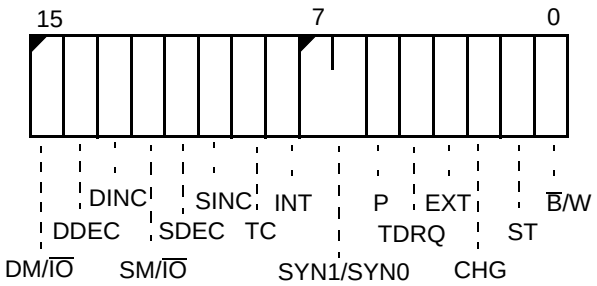
9.3.1 DMA Control Registers (D0CON, Offset CAh, D1CON, Offset DAh)

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 with the last transfer
- The mode of synchronization
- The relative priority of the DMA channel with respect to the other DMA channel
- Whether timer 2 DMA requests are enabled or disabled
- Whether bytes or words are transferred when both source and destination are configured as 16-bit address spaces. Only byte transfers are supported to 8-bit bus width address spaces.
- Whether a DRQ pin is used for external interrupts

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



The value of D0CON and D1CON at reset is undefined except ST is set to 0.

Bit 15: Destination Address Space Select (DM/$\overline{IO}$)—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 ($\overline{B}$/W, bit 0). The address remains constant if the increment and decrement bits are set to the same value (00b or 11b).

Bit 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 ($\overline{B}$/W, bit 0). The address remains constant if the increment and decrement bits are set to the same value (00b or 11b).

Bit 12: Source Address Space Select (SM/$\overline{IO}$)—When SM/$\overline{IO}$ 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. The value of this field is ignored if TDRQ (bit 4) is set to 1. For more information on DMA synchronization, see Section 9.4 on page 9-11. This field is 11b after processor reset.

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 2 Synchronization (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: External Interrupt Enable Bit (EXT)—This bit enables the external interrupt functionality of the corresponding DRQ pin. If this bit is set to 1, the external pin is an INT pin and requests on the pin are processed by the interrupt controller; the associated DMA channel does not respond to changes on the DRQ pin. When this bit is set to 0, the pin functions as a DRQ pin.

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. This bit always reads as 0.

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 1 during the same register write. This bit is 0 after processor reset.

Bit 0: Byte/Word Select ($\overline{B}$/W)—On the Am186ED/EDLV microcontrollers, when $\overline{B}$/W is set to 1, word transfers are selected. When $\overline{B}$/W is set to 0, byte transfers are selected. Only byte transfers are supported when either the source or the destination bus width is 8 bits.

9.3.2 Serial Port/DMA Transfers

The Am186ED/EDLV microcontrollers have the added feature of being able to DMA to and from the serial ports. This is accomplished by programming the DMA controller to perform transfers between a data buffer (located either in memory or I/O space) and a serial port data register (SP0TD, SP1TD, SP0RD, or SP1RD). It is important to note that when a DMA channel is in use by a serial port, the corresponding external DMA request signal is deactivated.

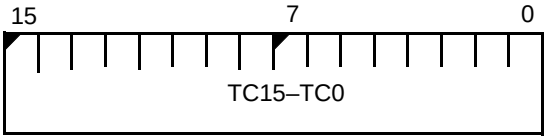
For DMA to the serial port, the transmit data register address, either I/O mapped or memory mapped, should be specified as a byte destination for the DMA by writing the address of the register into the DMA destination low and DMA destination high registers. The destination address (the address of the transmit data register) should be configured as a constant throughout the DMA operation. The serial port transmitter acts as the synchronizing device so the DMA channel should be configured as destination synchronized.

For DMA from the serial port, the receive data register address, either I/O mapped or memory mapped, should be specified as a byte source for the DMA by writing the address of the register into the DMA Source and DMA Source High registers. The source address (the address of the receive data register) should be configured as a constant throughout the DMA. The serial port receiver acts as the synchronizing device so the DMA channel should be configured as source synchronized.

9.3.3 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 each 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



The value of D0TC and D1TC at reset is undefined.

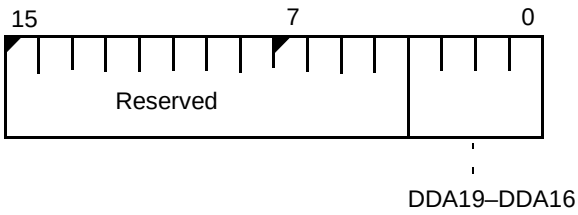
Bits 15–0: DMA Transfer Count (TC15–TC0)—Contains the transfer count for a DMA channel. Value is decremented by 1 after each transfer.

9.3.4 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 20-bit address 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 address registers for that channel must be initialized. These addresses 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 Am186ED/EDLV microcontrollers 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



The value of D0DSTH and D1DSTH at reset is undefined.

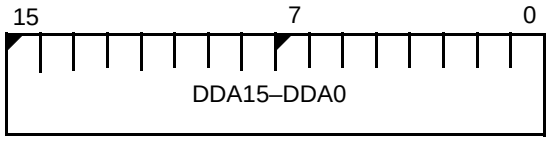
Bits 15–4: Reserved

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

9.3.5 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



The value of D0DSTL and D1DSTL at reset is undefined.

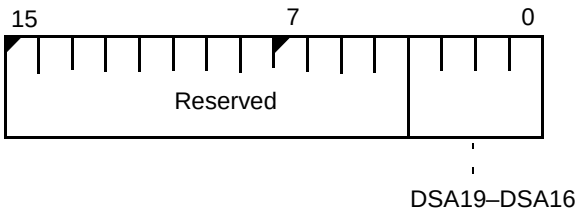
Bits 15–0: DMA Destination Address Low (DDA15–DDA0)—These bits are driven onto A15–A0 during the write phase of a DMA transfer.

9.3.6 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 20-bit address 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 address registers for that channel must be initialized. These addresses 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 Am186ED/EDLV microcontrollers 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



The value of D0SRCH and D1SRCH at reset is undefined.

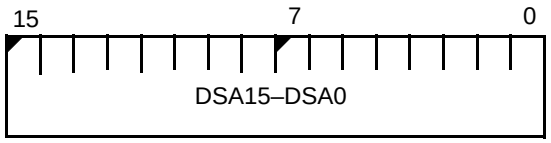
Bits 15–4: Reserved

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

9.3.7 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



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 gives 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

Synchronization Type	Maximum DMA Transfer Rate (Mbytes/sec)			
	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

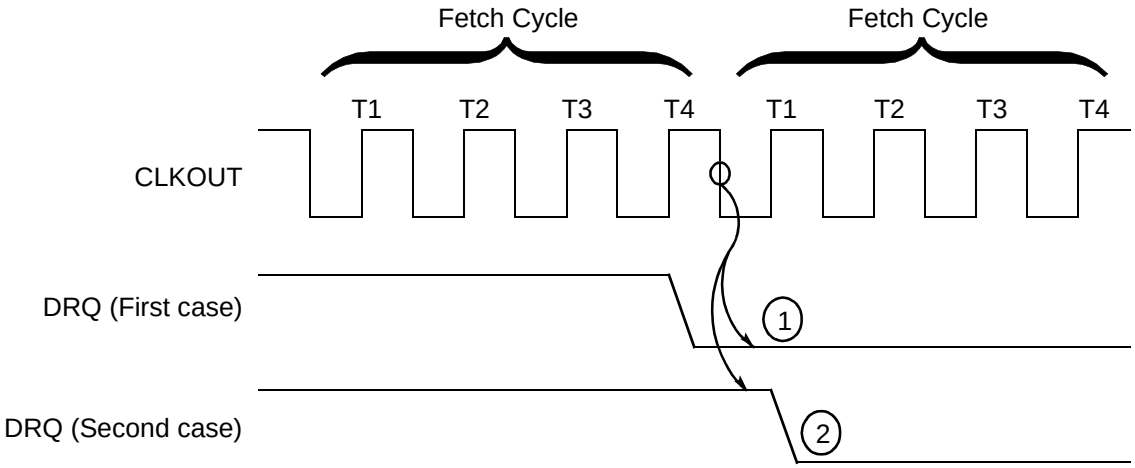
9.4.1 Synchronization Timing

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:

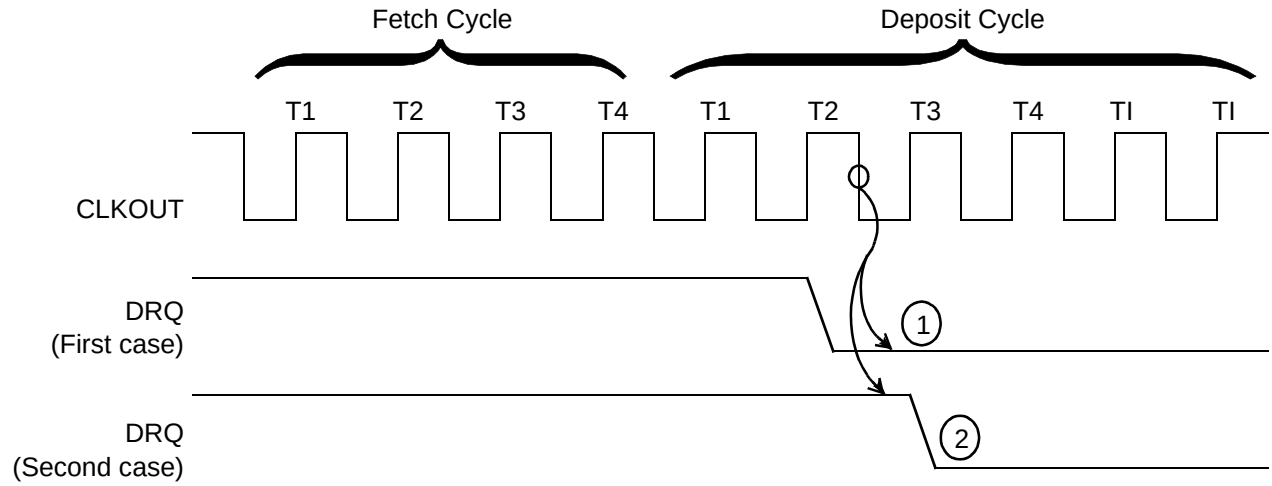
1. *This source-synchronized transfer is not followed immediately by another DMA transfer.*
2. *This source-synchronized transfer is immediately followed by another DMA transfer because DRQ is not deasserted soon enough.*

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 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.

DMA is also suspended during LOCKED bus cycles or during bus holds.

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, a 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 PORTS

10.1 OVERVIEW

The Am186ED/EDLV microcontrollers provide two independent asynchronous serial ports. These ports provide full-duplex, bidirectional data transfer using several industry-standard communications protocols. The serial ports may be used as sources or destinations of DMA transfers.

The asynchronous serial ports support the following features:

- Full-duplex operation
- 7-bit, 8-bit, or 9-bit data transfers
- Odd, even, or no parity
- One stop bit
- Two lengths of break characters
- Error detection
  - — Parity errors
  - — Framing errors
  - — Overrun errors
- Hardware handshaking with the following selectable control signals:
  - — Clear-to-send ($\overline{CTS}$)
  - — Enable-receiver-request ($\overline{ENRX}$)
  - — Ready-to-send ($\overline{RTS}$)
  - — Ready-to-receive ($\overline{RTR}$)
- DMA to and from the serial ports
- Separate maskable interrupts for each port
- Multidrop protocol (9-bit) support
- Independent baud rate generators
- Maximum baud rate of 1/16th of the CPU clock
- Double-buffered transmit and receive

10.1.1 Serial Port Flow Control

The Am186ED/EDLV microcontrollers provide two identical asynchronous serial ports. Four external pins are available for each port, as shown in Table 10-1.

Table 10-1 Serial Port External Pins

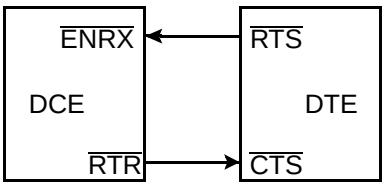
Pin Designation	Pin Function
RXD0, RXD1	Receives serial port data
TXD0, TXD1	Transmits serial port data
$\overline{CTS}$0, $\overline{CTS}$1/ $\overline{ENRX}$0, $\overline{ENRX}$1	Clear to send or enable receiver request
$\overline{RTS}$0, $\overline{RTS}$1/ $\overline{RTR}$0, RTR1	Ready to send or ready to receive

Each port is provided with two data pins (RXD0/RXD1 and TXD0/TXD1) and two flow control signals ($\overline{RTS}$0/$\overline{RTS}$1, $\overline{RTR}$0/$\overline{RTR}$1). The flow control signals are configurable by software to support several different protocols. These protocols are discussed below. Hardware flow control is enabled for the serial port when the FC bit in the serial port control register is set.

10.1.1.1 Hardware Flow Control

The Am186ED/EDLV microcontrollers implement only a subset of the data communication equipment/data terminal equipment (DCE/DTE) protocol. The DCE/DTE protocol provides flow control where one serial port is receiving data and the other serial port is sending data, as shown in Figure 10-1.

Figure 10-1 DCE/DTE Protocol



$\overline{RTS}$ = Request to send output from transmitter
$\overline{CTS}$ = Clear to send input to transmitter
$\overline{RTR}$ = Ready to receive output from receiver
$\overline{ENRX}$ = Enable receiver request input to receiver

10.1.1.2 DCE/DTE Protocol

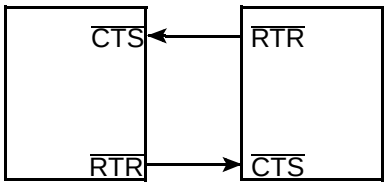
The serial ports of the Am186ED/EDLV microcontrollers can function as either DTE or DCE devices. To implement the DCE device, the ENRX bit should be set and the RTS bit should be cleared for the associated serial port. To implement the DTE device, the ENRX bit should be cleared and the RTS bit should be set for the associated serial port. These bits are located in the AUXCON register.

In the DCE/DTE protocol, the DTE device sends data. When data is available to send, the DTE device asserts the $\overline{RTS}$ signal. The $\overline{RTS}$ signal is interpreted by the DCE device as a request to enable its receiver. The DCE device signals the DTE device that it is ready to receive data by asserting the $\overline{RTR}$ signal. The interface is asymmetrical since the DTE device cannot signal the DCE device that it is ready to receive data; neither can the DCE device signal that it is ready to send data.

10.1.1.3 CTS/RTR Protocol

Note: *The clear-to-send/ready-to-receive (CTS/RTR) protocol provides flow control when both ports are sending and receiving data, as shown in Figure 10-2.*

Figure 10-2 CTS/RTR Protocol



$\overline{CTS}$ = Clear to send input to transmitter
$\overline{RTR}$ = Ready to receive output from receiver

The serial ports of the Am186ED/EDLV microcontrollers can be configured for the CTS/RTR protocol by clearing both the ENRX and the RTS bits for the associated serial port. This is the default configuration.

The CTS/RTR protocol provides a symmetrical interface. When a device is ready to receive data, i.e., there is no unread data in the serial port receive register, the device asserts the $\overline{RTR}$ signal. A device does not begin transmitting data until the $\overline{CTS}$ signal is asserted.

10.1.1.4 Flow Control and Transmission Rates

Although the use of flow control can eliminate the possibility of overrun errors—data loss due to reception of new data before the last received data has been read—it can have an effect on serial port transmission rates.

On the Am186ED/EDLV microcontrollers, the $\overline{RTR}$ signal is not asserted until valid data from a previous transmission has been read out of the receive register, i.e., the RDR bit in the serial port status register is cleared. This means that the transmitting serial port will not begin transmission of the next data item until software running on the Am186ED/EDLV microcontrollers has had a chance to read the last data item. This is not the case when flow control is not being used and data continues to be sent, allowing software a minimum of one frame transmission time to read the data in order to prevent overrun errors. To minimize this delay, the receive register should be read as soon as possible after the completion of the reception of data. Using the serial port receiver as a DMA source provides the shortest possible delay.

10.2 PROGRAMMABLE REGISTERS

The asynchronous serial ports are programmed through the use of 10, 16-bit peripheral registers. See Table 10-2.

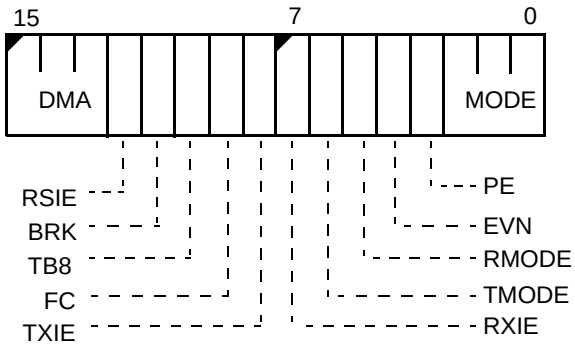
Table 10-2 Asynchronous Serial Ports Register Summary

Offset from PCB	Register Mnemonic	Register Name
80h	SP0CT	Serial Port 0 Control
82h	SP0STS	Serial Port 0 Status
88h	SP0BAUD	Serial Port 0 Baud Rate Divisor
86h	SP0RD	Serial Port 0 Receive
84h	SP0TD	Serial Port 0 Transmit
10h	SP1CT	Serial Port 1 Control
12h	SP1STS	Serial Port 1 Status
18h	SP1BAUD	Serial Port 1 Baud Rate Divisor
16h	SP1RD	Serial Port 1 Receive
14h	SP1TD	Serial Port 1Transmit

10.2.1 Serial Port 0/1 Control Registers (SP0CT/SP1CT, Offset 80h/10h)

The serial port control registers control both the transmit and receive sections of the serial port. The format of the serial port control registers is shown in Figure 10-3.

Figure 10-3 Serial Port Control Register



The value of SP0CT/SP1CT at reset is 0000h.

Bits 15–13: DMA Control Field (DMA)—This field configures the serial port for use with DMA transfers according to the following table.

Table 10-3 DMA Control Bits

DMA Bits	Receive	Transmit
000b	No DMA	No DMA
001b	DMA0	DMA1
010b	DMA1	DMA0
011b	Reserved	Reserved
100b	DMA0	No DMA
101b	DMA1	No DMA
110b	No DMA	DMA0
111b	No DMA	DMA1

DMA transfers to a serial port function as destination-synchronized DMA transfers. A new transfer is requested when the transmit holding register is empty. This corresponds with the assertion of the THRE bit in the serial port status register in non-DMA mode. When the port is configured for DMA transmits, the corresponding transmit interrupt is disabled regardless of the setting of the TXIE bit.

DMA transfers from the serial port function as source-synchronized DMA transfers. A new transfer is requested when the serial port receive register contains valid data. This corresponds with the assertion of the RDR bit in the serial port status register in non-DMA mode. When the port is configured for DMA receives, the corresponding receive interrupt is disabled regardless of the setting of the RXIE bit. Receive status interrupts may still be taken, as configured by the RSIE bit.

Hardware handshaking may be used in conjunction with serial port DMA transfers.

When a DMA channel is being used for serial port transmits or receives, the DMA request is generated internally. The corresponding external DMA request signals, DRQ0 or DRQ1, are not active for serial port DMA transfers.

Bit 12: Receive Status Interrupt Enable (RSIE)—This bit enables the serial port to generate an interrupt request when an exception occurs during data reception. When this bit is set, interrupt requests are generated for the error conditions reported in the serial port status register (BRK0, BRK1, OER, PER, FER).

Bit 11: Send Break (BRK)—When this bit is set, the TXD pin is driven Low regardless of the data being shifted out of the transmit register.

A short break, as reported by the BRK0 bit in the status register, is a continuous Low on the TXD output for a duration of more than one frame transmission time M, where M = start bit + data bits (+ parity bit)+ stop bit. The transmitter can be used to time the break by setting the BRK bit when the transmitter is empty (indicated by the TEMT bit of the serial port status register), writing the serial port transmit register with data, then waiting until the TEMT bit is again set before resetting the BRK bit.

A long break, as reported by the BRK1 bit in the status register, is a continuous Low on the TXD output for a duration of more than two frame transmission times plus the transmission time for three additional bits (2M+3). The transmitter can be used to time the break as follows:

1. Wait for the TEMT bit in the status register to be set.
2. Set the BRK bit.
3. Perform two sequential writes to the transmit register.
4. Wait for the TEMT bit in the status register to be set again.
5. Write a character with the low nibble zeroed and the high nibble High (for example, F0h).
6. Clear the BRK bit. The character being transmitted continues to hold the TXD pin Low for the required additional 3-bit transmission time.

Note: *The transmitter can only be used to time the break if hardware flow control is disabled. If flow control is enabled, setting the BRK bit will still force the TXD line Low, but the receiving device may deassert the CTS input, inhibiting the clocking out of the character in the transmit data register.*

In order for the serial port to recognize either a short or long break with the timing discussed above, the break must begin outside of a data frame. If the serial port is currently receiving a data frame (i.e., a frame with at least one high bit), that frame will be completely received before timing for break detection begins.

Bit 10: Transmit Bit 8 (TB8)—This bit is transmitted as the ninth data bit in modes 2, 3, and 7 (see the mode field description). This bit is not buffered and is cleared after every transmission. In order to transmit a character with the 8th data bit High, the following protocol should be followed:

1. Wait for the TEMT bit in the status register to become set.
2. Write the control register with this bit set.
3. Write the character to be transmitted.

Bit 9: Flow Control Enable (FC)—When this bit is 1, hardware flow control is enabled for the associated serial port. When this bit is 0, hardware flow control is disabled for the associated serial port. The nature of the flow control signals is determined by the setting of the ENRX0/ENRX1 and RTS0/RTS1 bits in the AUXCON register. See the discussion of the AUXCON register and Section 10.1.1 on page 10-1 for more information. If this bit is 1 for serial port 0, the associated pins are used as flow control signals, overriding their function as Peripheral Chip Select signals. This bit is 0 after processor reset.

Bit 8: Transmitter Ready Interrupt Enable (TXIE)—When this bit is set, the serial port generates an interrupt request whenever the transmit holding register is empty (THRE bit in the status register is set), indicating that the transmitter is available to accept a new character for transmission. When this bit is reset, the serial port does not generate transmit interrupt requests. Interrupt requests continue to be generated as long as the TXIE bit is set and the transmitter does not contain valid data to transmit, i.e., the THRE bit in the status register remains set.

Bit 7: Receive Data Ready Interrupt Enable (RXIE) —When this bit is set, the serial port generates an interrupt request whenever the receive register contains valid data (RDR bit in the status register is set). When this bit is reset, the serial port does not generate receive interrupt requests. Interrupt requests continue to be generated as long the RXIE bit is set and the receiver contains unread data (the RDR bit in the status register is set).

Bit 6: Transmit Mode (TMODE)—When this bit is set, the transmit section of the serial port is enabled. When this bit is reset, the transmitter and transmit interrupt requests are disabled.

Bit 5: Receive Mode (RMODE)—When this bit is set, the receive section of the serial port is enabled. When this bit is reset, the receiver is disabled.

Bit 4: Even Parity (EVN)—This bit determines the parity sense. When EVN is set, even parity checking is enforced (even number of 1s in frame). When EVN is reset, odd parity checking is enforced (odd number of 1s in frame).

Note: *This bit is valid only when the PE bit is set (parity enabled).*

Bit 3: Parity Enable (PE)—When this bit is set, parity generation and checking is enabled. When this bit is reset, parity generation and checking is disabled. Parity should not be enabled when the serial port is configured in mode 7 or an interrupt request will be generated based on the parity bit.

Bits 2–0: Mode of Operation (MODE)—This field determines the operating mode for the serial port. The valid modes and their descriptions are shown in Table 10-4.

Mode 1 supports 7 data bits when parity is enabled or 8 data bits with parity disabled. When using parity, the eighth bit becomes the parity bit and is generated for transmits, or checked for receives automatically by the processor.

Table 10-4 Serial Port MODE Settings

MODE	Description	Data Bits	Parity Bits	Stop Bits
0	Reserved			
1	Data Mode 1	7 or 8	1 or 0	1
2	Data Mode 2	9	N/A	1
3	Data Mode 3	8 or 9	1 or 0	1
4	Data Mode 4	7	N/A	1
5	Reserved			
6	Reserved			
7	Data Mode 7	9	N/A	1

Mode 2—When configured in this mode, the serial port receiver will not complete a data reception unless the ninth data bit is set (High). Any character received with the ninth data bit reset (Low) is ignored. The transmit portion of the port behaves identically with mode 3 operation.

In a serial multidrop configuration, multiple serial ports are attached to the same serial line. The master serial port is configured in mode 3 or mode 7 while the slave serial ports are configured in mode 2. The master polls the other devices by sending out status request packets. Each of these status request packets begins with an address byte (i.e., ninth data bit is set). The slave ports report a receive character for the address byte since the ninth bit is set. Each port then attempts to match the address against its own address. If the addresses do not match, the port remains in mode 2 and ignores the remainder of the message. If the addresses match, software reconfigures the port into either mode 3 or mode 7. The two mode 3 or 7 ports are able to exchange data freely.

It should be noted that only ports which are actively exchanging data (i.e., ports in modes 3 or 7) should have hardware handshaking enabled. If this is not the case, multiple devices may be driving the hardware handshaking lines. For this reason, hardware handshaking is not supported for the mode 2 configuration and should not be enabled. In addition, if it is possible for more than two devices to be configured as mode 3 or mode 7 at any one time, hardware handshaking should not be enabled.

Mode 3—Supports 8 data bits when parity is enabled or 9 data bits with parity disabled. When not using parity, the ninth bit (bit 8) for transmission is set by writing a 1 to the TB8 field in the Serial Port Control register. The ninth data bit for a receive can be read in the RB8 field of the serial port status register. See the discussion of the TB8 and RB8 fields for more information.

This mode can be used in conjunction with mode 2 (see above) to allow for multidrop communications over a common serial link. In this case, parity must be disabled. In this configuration, software interprets receive characters as data as long as the ninth data bit is reset (Low). When a character is received with the ninth bit set, software should compare the lower eight bits against the port ID. If the port ID matches the receive data, the port should remain in mode 3. If the port ID does not match the receive data, the port should be reconfigured to mode 2.

Mode 4—In this mode, each frame consists of 7 data bits, a start bit, and a stop bit. Parity is not available in this mode.

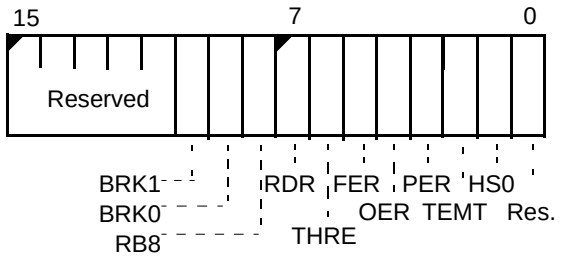
Mode 7—Mode 7 supports 9 data bit frames with no parity. The ninth bit (bit 8) for transmission is set by writing a 1 to the TB8 field in the Serial Port Control register. The ninth data bit for a receive can be read in the RB8 field of the Serial Port Status register. See the discussion of the TB8 and RB8 fields for more information.

This mode can be used in conjunction with mode 2 (see above) to allow for a multidrop communications over a common serial link. In this configuration, software interprets received characters as data as long as the 9th bit is reset (Low). When a character is received with the ninth bit set, the serial port generates an interrupt request. This interrupt request is not maskable via the RXIE or RSIE bits in the Serial Port Control (SP0CT/SP1CT) register but can be masked via the MSK bit in the Serial Port Interrupt Control (SP0CON/SP1CON) registers. The interrupt on the ninth bit set allows software to be informed of the special frames (such as addresses or flags) when using the DMA controller to receive data from the serial port. The RB8 bit must be cleared by software to remove the interrupting condition.

10.2.2 Serial Port 0/1 Status Registers (SP0STS/SP1STS, Offset 82h/12h)

The Serial Port Status Registers provide information about the current status of the associated serial port. The THRE and TEMT fields provide the software with information about the state of the transmitter. The BRK1, BRK0, RB8, RDR, FER, OER, and PER bits provide information about the receiver. The HS0 bit reflects the value of the serial port's associated CTS/ENRX signal. The THRE, TEMT, and HS0 bits are updated during each processor cycle. The format of the Serial Port Status register is shown in Figure 10-4.

Figure 10-4 Serial Port 0/1 Status Register



Bits 15–11: Reserved

Bit 10: Long Break Detected (BRK1)—This bit is set when a long break is detected on the asynchronous serial interface. A long break is defined as a Low signal on the RXD pin for greater than 2M+3 bit times, where M = (start bit + # data bits + # parity bits + stop bit).

If the serial port is receiving a character when the break begins, the reception of the character will be completed (generating a framing error) before timing for the break begins. To guarantee detection with the specified 2M+3 bit times, the break must begin outside of a frame.

Note: *This bit should be reset by software.*

Bit 9: Short Break Detected (BRK0)—This bit is set when a short break is detected on the asynchronous serial interface. A short break is defined as a Low signal on the RXD pin for greater than M bit times, where M = (start bit + # data bits + # parity bits + stop bit).

If the serial port is receiving a character when the break begins, the reception of the character will be completed (generating a framing error) before timing for the break begins. To guarantee detection with the specified M bit times, the break must begin outside of a frame.

Note: *This bit should be reset by software.*

Bit 8: Received Bit 8 (RB8)—This bit contains the ninth data bit received in modes 2, 3, and 7. (See Serial Port Control register definition.)

Note: *This bit should be reset by software.*

Bit 7: Receive Data Ready (RDR)—When this bit is set, the corresponding Receive Data register contains valid data. This field is read-only. The RDR bit can only be reset by reading the associated SP0RD/SP1RD register.

Bit 6: Transmit Holding Register Empty (THRE)—When this bit is set, the transmit holding register is ready to accept data for transmission. This field is read-only.

Bit 5: Framing Error Detected (FER)—When this bit is set, the serial port has detected a framing error. Framing errors are generated when the receiver samples the RXD line as Low when it expected the stop bit.

Note: *This bit should be reset by software.*

Bit 4: Overrun Error Detected (OER)—This bit is set when the processor detects an overrun error. An overrun error occurs when the serial port overwrites valid, unread data in the receive register, resulting in loss of data.

Note: *This bit should be reset by software.*

Bit 3: Parity Error Detected (PER)—This bit is set when the processor detects a parity error (modes 1 and 3).

Note: *This bit should be reset by software.*

Bit 2: Transmitter Empty (TEMT)—When this bit is set, 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 1: Handshake Signal 0 (HS0)—This bit reflects the inverted value of the external $\overline{\text{CTS}}$ pin. If $\overline{\text{CTS}}$ is asserted, HS0 is set to 1. This bit is read-only.

Bit 0: Reserved

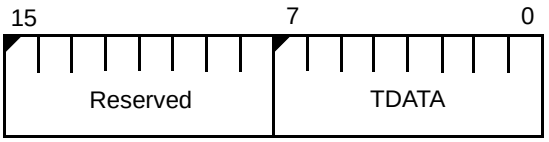
10.2.3 Serial Port 0/1 Transmit Registers (SP0TD/SP1TD, Offset 84h/14h)

The transmit registers (Figure 10-5) are written by software with the value to be transmitted over the serial interface. The transmitter is double-buffered; data to be transmitted is copied from the transmit register to the transmit shift register (which is not accessible to software) before transmitting. The value of the ninth data bit for transmission in modes 2, 3, and 7 is set in the TB8 field in the Serial Port Control registers. The state of the transmit and transmit shift registers is reflected in the TEMT and THRE bits in the associated Serial Port Status register.

When hardware handshaking is enabled, the transmitter will not transmit data while $\overline{\text{RTS}}$/$\overline{\text{RTR}}$ input is deasserted. Data is held in the transmit and transmit shift registers without affecting the transmit pin.

The serial port transmit register in the Am186EM microcontroller is renamed in the Am186ED/EDLV microcontrollers as the Serial Port 0 Transmit register.

Figure 10-5 Serial Port 0/1 Transmit Registers



The value of SPTD at reset is undefined.

Bits 15–8: Reserved

Bit 7–0: Transmit Data (TDATA)—This field contains data to be transmitted through the asynchronous serial port.

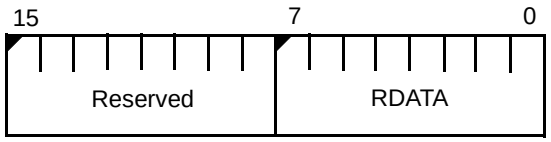
10.2.4 Serial Port 0/1 Receive Registers (SP0RD/SP1RD, Offset 86h/16h)

These registers (Figure 10-6) contain data received over the serial port. The receiver is double-buffered; 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.

The Receive-Data-Ready (RDR) bit in the serial port status register reports the current state of this register. When the RDR bit is set, the receive register contains valid unread data. The RDR bit is automatically cleared when the receive register is read.

When hardware handshaking is enabled, the $\overline{\text{CTS}}$/$\overline{\text{ENRX}}$ signals are deasserted while the receive register contains valid unread data. Reading the receive register causes the $\overline{\text{CTS}}$/$\overline{\text{ENRX}}$ signals to be asserted. This behavior prevents overrun errors, but may result in delays between character transmissions.

Figure 10-6 Serial Port Receive 0/1 Registers



The value of SPRD at reset is undefined.

Bits 15–8: Reserved

Bits 7–0: Receive Data (RDATA)—This field holds valid data received over the serial line only when the RDR bit in the associated serial port control register is set.

10.2.5 Serial Port 0/1 Baud Rate Divisor Registers (SP0BAUD/SP1BAUD, Offset 88h/18h)

Each of the asynchronous serial ports has a baud rate divisor register, so the two ports can operate at different rates.

These registers (Figure 10-7) specify a clock divisor for the generation of the serial clock that controls the associated serial port. The baud rate divisor 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. Since power-save mode is automatically exited when an interrupt is taken, serial port transmits and receives may be corrupted if the serial port is in use and interrupts are enabled during power-save mode.

A general formula for the baud rate divisor is:

BAUDDIV = (Processor Frequency ÷ (16 × Baud Rate))

The maximum baud rate is 1/16 of the internal processor clock and is achieved by setting BAUDDIV=0001h. This results in a baud rate of 2500 Kbit at 20 MHz, 1562.5 Kbit at 25MHz, 2062.5 Kbit at 33 MHz, and 1250 Kbit at 40 MHz. A BAUDDIV setting of zero results in no transmission or reception of data.

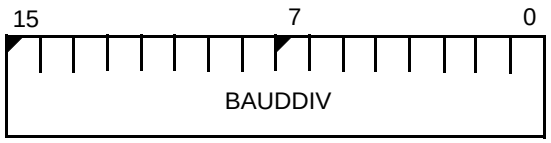
The serial port receiver can tolerate a 3.0% overspeed and 2.5% underspeed baud rate deviance.

Table 10-5 Common Baud Rates

Baud Rate	Divisor Based on CPU Clock Rate			
	20 MHz	25 MHz	33 MHz	40 MHz
300	4166	5208	6875	8333
600	2083	2604	3437	4166
1050	1190	1488	1964	2380
1200	1041	1302	1718	2083
1800	694	868	1145	1388
2400	520	651	859	1041
4800	260	325	429	520
7200	173	217	286	347
9600	130	162	214	260
19200	65	81	107	130
28800	43	54	71	86
38400	33	40	53	65
56000	22	28	36	45
57600	22	27	35	43
76800	16	20	26	32
115200	10	13	18	22
128000	9	12	16	19
153600	8	10	13	16
187500	5	7	8	10

Note: *A 1% error applies to all values in the above table.*

Figure 10-7 Serial Port 0/1 Baud Rate Divisor Registers



The value of SPBAUD at reset is 0000h.

Bits 15–0: Baud Rate Divisor (BAUDDIV)—This field specifies the divisor for the internal processor clock.

CHAPTER

11 PROGRAMMABLE I/O PINS

11.1 OVERVIEW

Thirty-two pins on the Am186ED/EDLV 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 11-1 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/$\overline{R}$, $\overline{DEN}$, and SRDY pins also default to normal operation on power-on reset. PIO15 and PIO24 should be set to normal operation before enabling either bank of DRAM. PIO25 should be set to normal operation before enabling the upper bank of DRAM.

When the $\overline{PCS}$ or $\overline{MCS}$ pins are used as PIO inputs and the bus is arbitrated, an internal pullup of ~9 kohms is activated, even if the pullup option for the PIO is not selected.

Figure 11-1 Programmable I/O Pin Operation

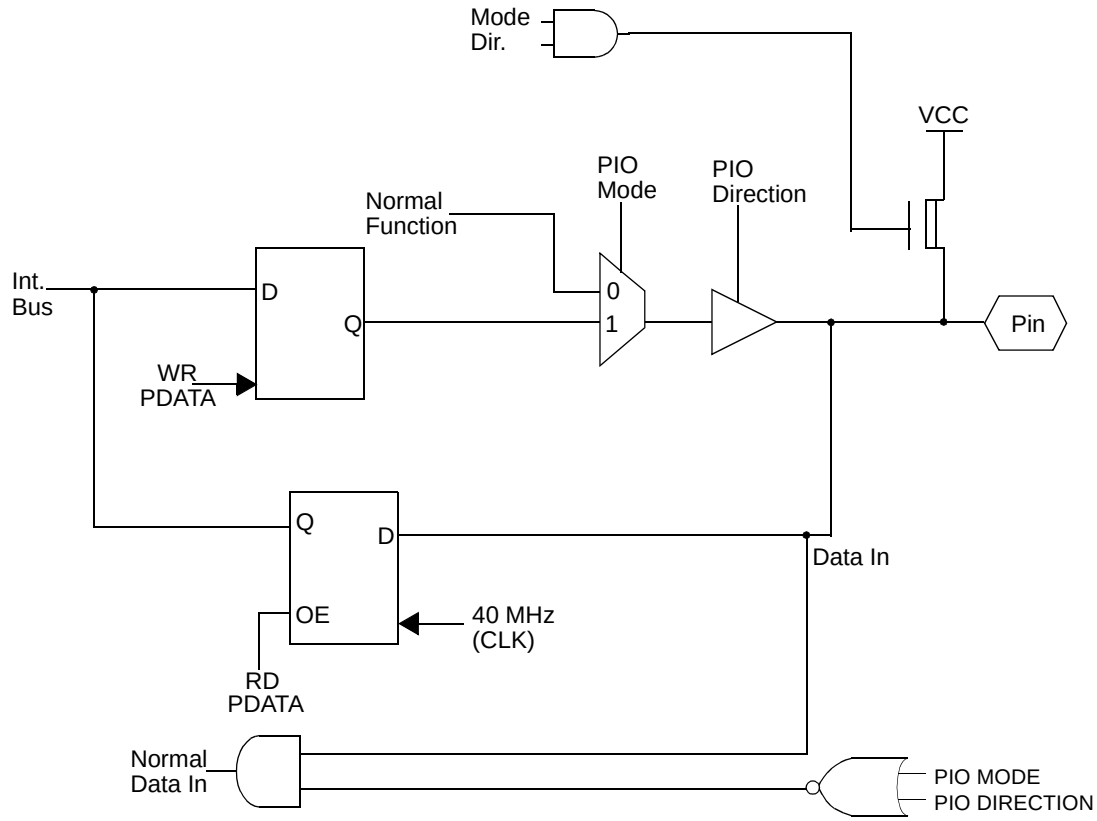


Table 11-1 PIO Pin Assignments

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	$\overline{\text{DEN}}$/$\overline{\text{DS}}$	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/INT5	Input with pullup
13	DRQ1/INT6	Input with pullup
14	$\overline{\text{MCS}}$0	Input with pullup
15	$\overline{\text{MCS}}$1/$\overline{\text{UCAS}}$	Input with pullup
16	$\overline{\text{PCS}}$0	Input with pullup
17	$\overline{\text{PCS}}$1	Input with pullup
18	$\overline{\text{PCS}}$2/$\overline{\text{CTS}}$1/$\overline{\text{ENRX}}$1	Input with pullup
19	$\overline{\text{PCS}}$3/$\overline{\text{RTS}}$1/$\overline{\text{RTR}}$1	Input with pullup
20	$\overline{\text{RTS}}$0/$\overline{\text{RTR}}$0	Input with pullup
21	$\overline{\text{CTS}}$0/$\overline{\text{ENRX}}$0	Input with pullup
22	TXD0	Input with pullup
23	RXD0	Input with pullup
24	$\overline{\text{MCS}}$2/$\overline{\text{LCAS}}$	Input with pullup
25	$\overline{\text{MCS}}$3/$\overline{\text{RAS}}$1	Input with pullup
26(1,2)	$\overline{\text{UZI}}$	Input with pullup
27	TXD1	Input with pullup
28	RXD1	Input with pullup
29(1,2)	S6/$\overline{\text{CLKDIV}}$2	Input with pullup
30	INT4	Input with pullup
31	INT2/$\overline{\text{INTA}}$0/PWD	Input with pullup

Notes:

1. *These pins are used by many 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 $\overline{BHE}$/$\overline{ADEN}$ 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.*

11.2 PIO MODE REGISTERS

Table 11-2 shows the possible settings for the PIO Mode and PIO Direction bits. The Am186ED/EDLV 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 11-2. The column titled *Power-On Reset State* in Table 11-1 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 11-2 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 w/o pullup/pulldown

Figure 11-2 PIO Mode 1 Register (PIOMODE1, Offset 76h)

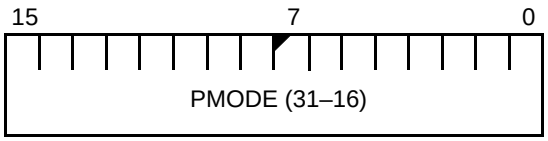
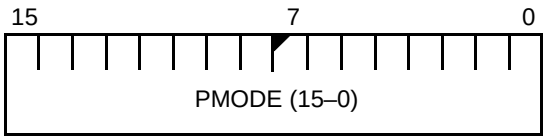


Figure 11-3 PIO Mode 0 Register (PIOMODE0, Offset 70h)



11.2.1 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, along with the PIO direction registers, determines whether each PIO pin performs its preassigned 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 11-2 shows the values that the PIO mode bits and the PIO direction bits can encode.

11.2.2 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.

11.3 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 11-4 and Figure 11-5). Table 11-2 on page 11-3 shows the values that the PIO mode bits and the PIO direction bits can encode. The column titled *Power-On Reset Status* in Table 11-1 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 11-4 PIO Direction 1 Register (PDIR1, Offset 78h)

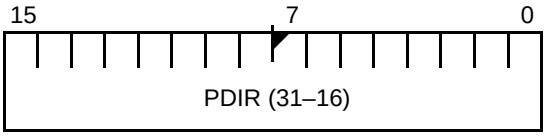
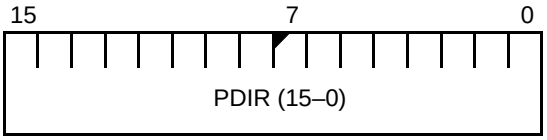


Figure 11-5 PIO Direction 0 Register (PDIR0, Offset 72h)



11.3.1 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.

11.3.2 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.

11.4 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 11-6 and Figure 11-7) 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 11-6 PIO Data 1 Register (PDATA1, offset 7Ah)

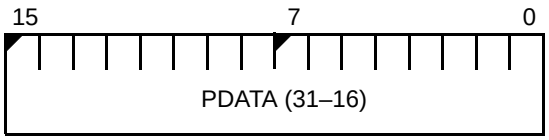
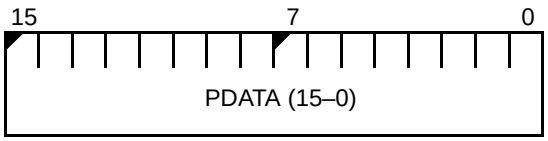


Figure 11-7 PIO Data 0 Register (PDATA0, offset 74h)



11.4.1 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.

11.4.2 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.

11.5 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.

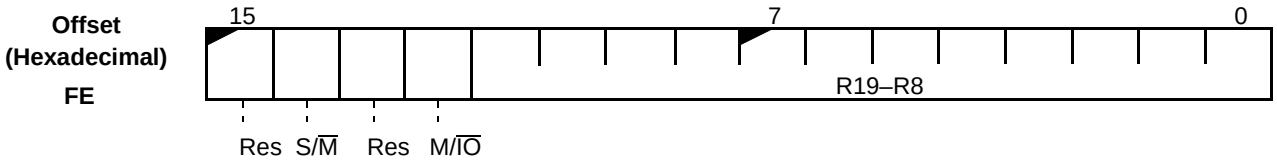
Table A-1 Internal Register Summary

Hex Offset	Mnemonic	Register Description	Comment
FE	RELREG	Peripheral control block relocation register	
F6	RESCON	Reset configuration register	
F4	PRL	Processor release level register	
F2	AUXCON	Auxiliary configuration	
F0	SYSCON	System configuration register	
E6	WDT	Watchdog timer control register	
E4	EDRAM	Enable RCU register	
E2	CDRAM	Clock prescalar register	
DA	D1CON	DMA 1 control 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	

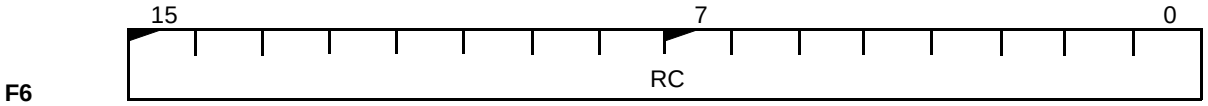
Hex Offset	Mnemonic	Register Description	Comment
A2	LMCS	Low memory chip select register	
A0	UMCS	Upper memory chip select register	
88	SP0BAUD	Serial port 0 baud rate divisor register	
86	SP0RD	Serial port 0 receive data register	
84	SP0TD	Serial port 0 transmit data register	
82	SP0STS	Serial port 0 status register	
80	SP0CT	Serial port 0 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	
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	SP0CON	Serial port 0 interrupt control register	Master mode
42	SP1CON	Serial port 1 interrupt control register	Master mode
40	INT4CON	INT4 control register	Master mode
3E	INT3CON	INT3 control register	Master mode
3C	INT2CON	INT2 control register	Master mode
3A	INT1CON	INT1 control register	Master mode
	T2INTCON	Timer 2 interrupt control register	Slave mode
38	INT0CON	INT0 control register	Master mode
	T1INTCON	Timer 1 interrupt control register	Slave mode

Hex Offset	Mnemonic	Register Description	Comment
36	DMA1CON/ INT6CON	DMA 1 interrupt control register/INT6	Slave & master
34	DMA0CON/ INT5CON	DMA 0 interrupt control register/INT5	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	SP1BAUD	Serial port 1 baud rate divisor register	
16	SP1RD	Serial port 1 receive register	
14	SP1TD	Serial port 1 transmit register	
12	SP1STS	Serial port 1 status register	
10	SP1CT	Serial port 1 control register	

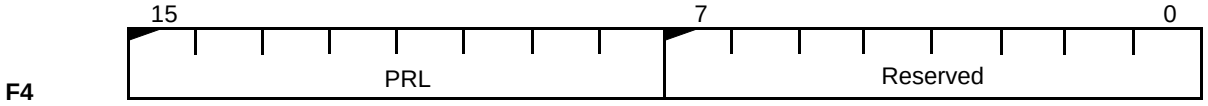
Figure A-1 Internal Register Summary



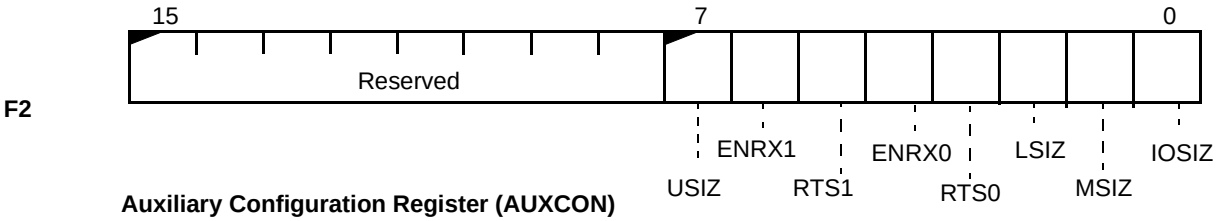
Peripheral Control Block Relocation Register (RELREG)
Page 4-3



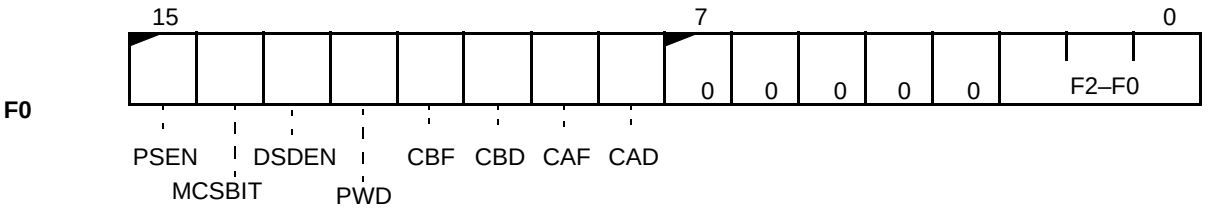
Reset Configuration Register (RESCON)
Page 4-4



Processor Release Level Register (PRL)
Page 4-5

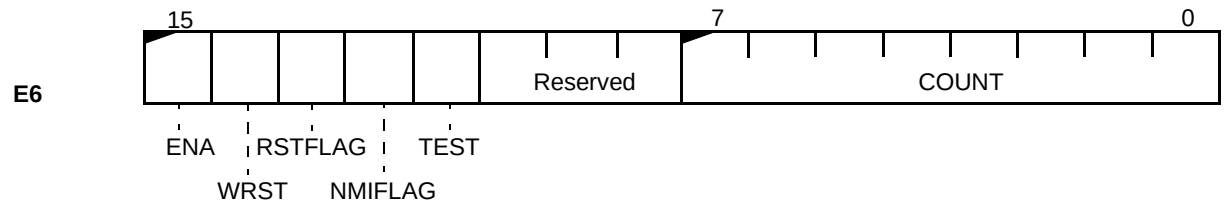


Auxiliary Configuration Register (AUXCON)
Page 4-6

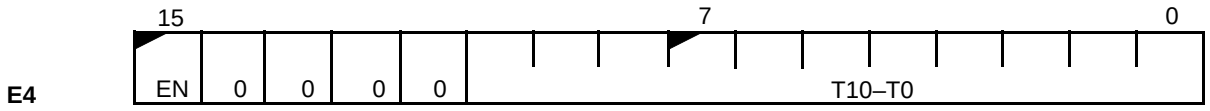


System Configuration Register (SYSCON)
Page 4-8

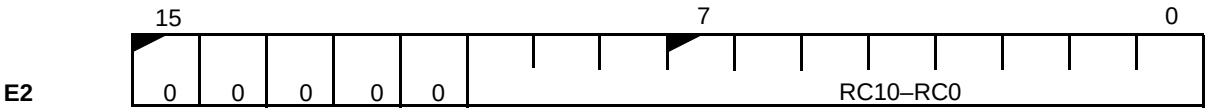
Figure A-1 Internal Register Summary (continued)



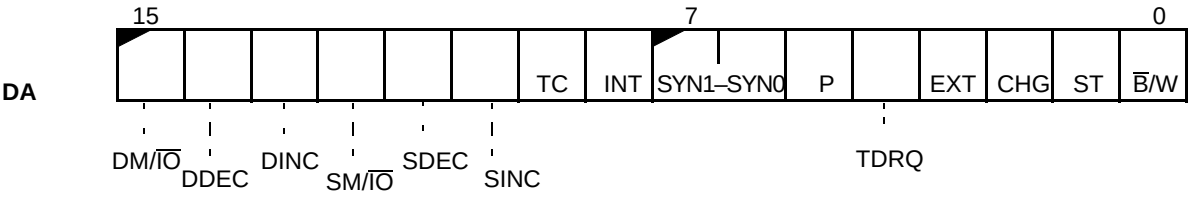
Watchdog Timer Control Register (WDTCON)
Page 6-8



Enable RCU Register (EDRAM)
Page 6-7



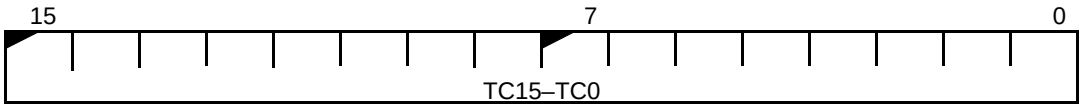
Clock Prescalar Register (CDRAM)
Page 6-6



DMA 1 Control Register (D1CON)
Page 9-3

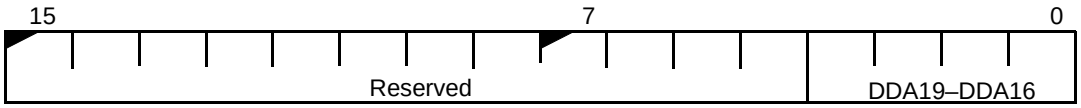
Figure A-1 Internal Register Summary (continued)

D8



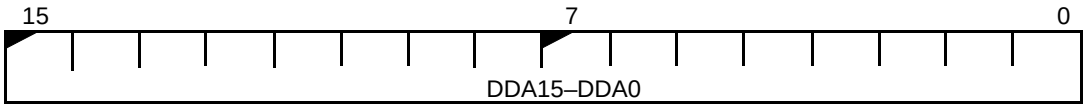
DMA 1 Transfer Count Register (D1TC)
Page 9-6

D6



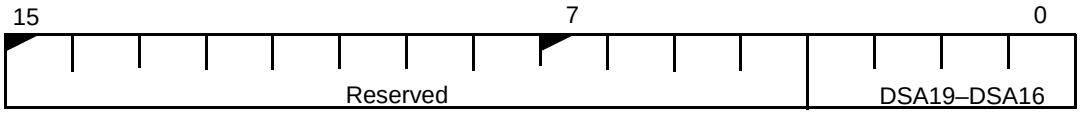
DMA 1 Destination Address High Register (D1DSTH)
Page 9-7

D4



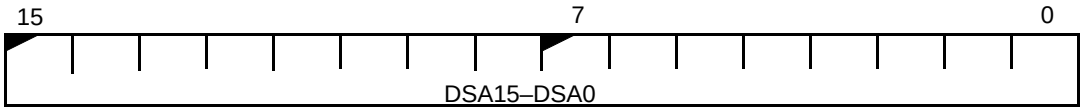
DMA 1 Destination Address Low Register (D1DSTL)
Page 9-8

D2



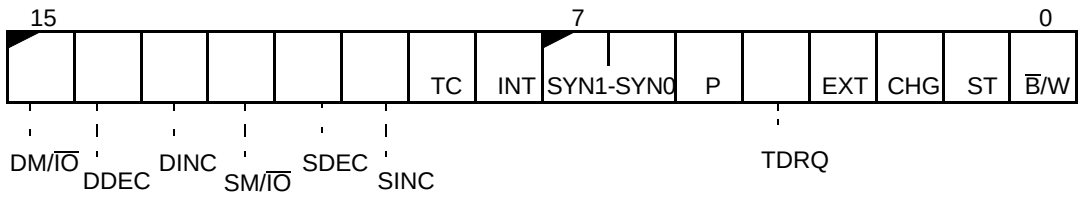
DMA 1 Source Address High Register (D1SRCH)
Page 9-9

D0



DMA 1 Source Address Low Register (D1SRCL)
Page 9-10

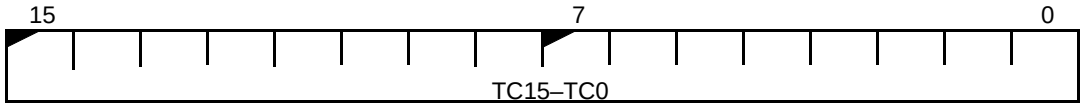
CA



DMA 0 Control Register (D0CON)
Page 9-3

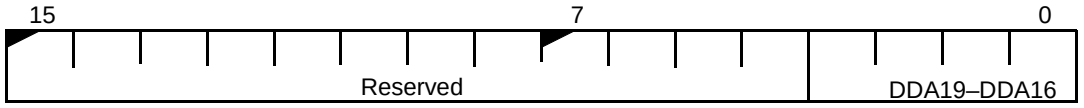
Figure A-1 Internal Register Summary (continued)

C8



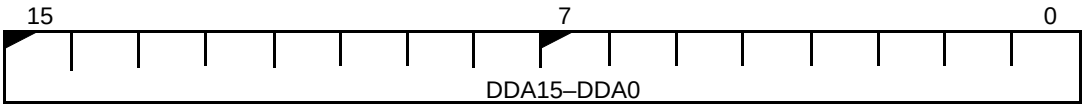
DMA 0 Transfer Count Register (D0TC)
Page 9-6

C6



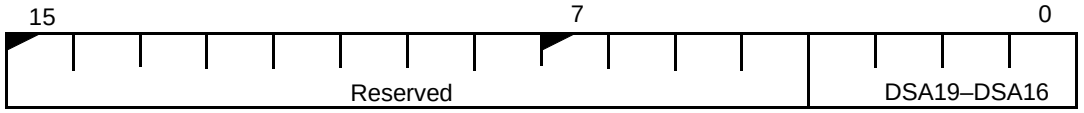
DMA 0 Destination Address High Register (D0DSTH)
Page 9-7

C4



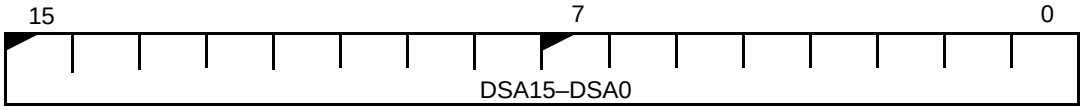
DMA 0 Destination Address Low Register (D0DSTL)
Page 9-8

C2



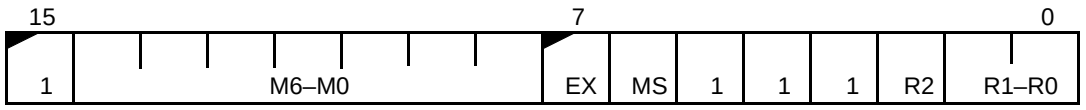
DMA 0 Source Address High Register (D0SRCH)
Page 9-9

C0



DMA 0 Source Address Low Register (D0SRCL)
Page 9-10

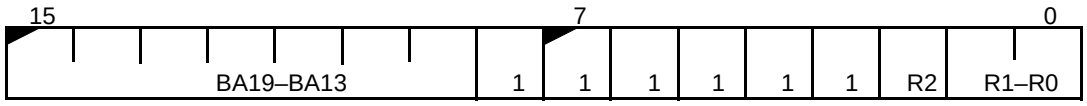
A8



$\overline{\text{PCS}}$ and $\overline{\text{MCS}}$ Auxiliary Register (MPCS)
Page 5-10

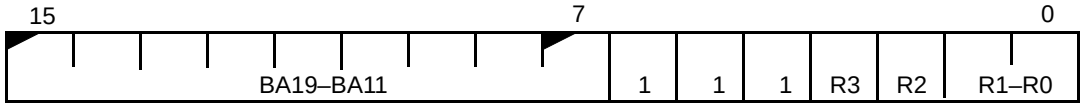
Figure A-1 Internal Register Summary (continued)

A6



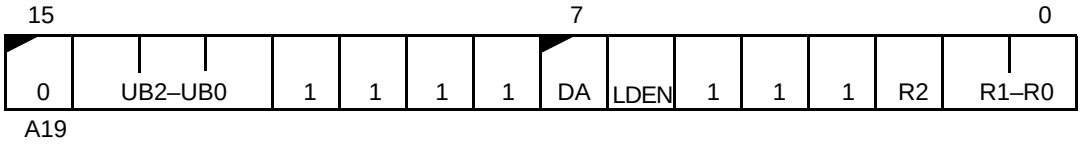
Midrange Memory Chip Select Register (MMCS)
Page 5-8

A4



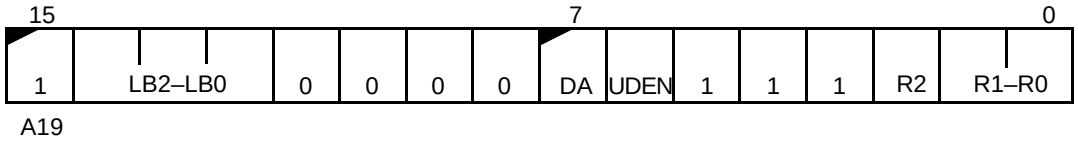
Peripheral Chip Select Register (PACS)
Page 5-12

A2



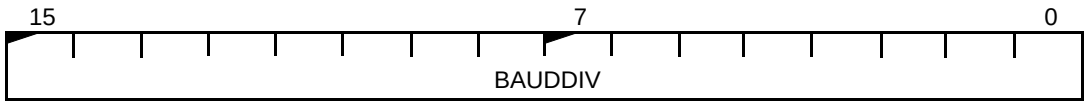
Low Memory Chip Select Register (LMCS)
Page 5-6

A0



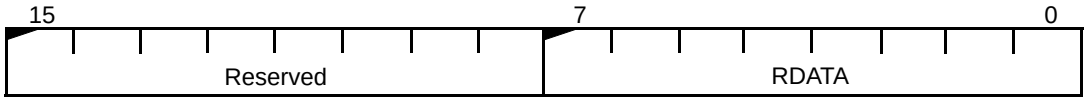
Upper Memory Chip Select Register (UMCS)
Page 5-4

88



Serial Port 0 Baud Rate Divisor Register (SP0BAUD)
Page 10-15

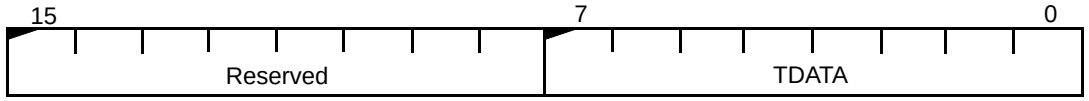
86



Serial Port 0 Receive Register (SP0RD)
Page 10-13

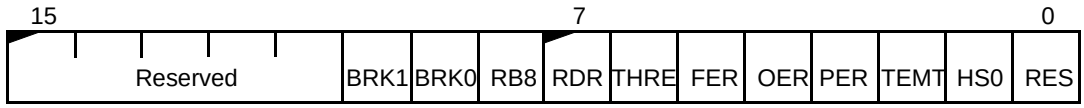
Figure A-1 Internal Register Summary (continued)

84



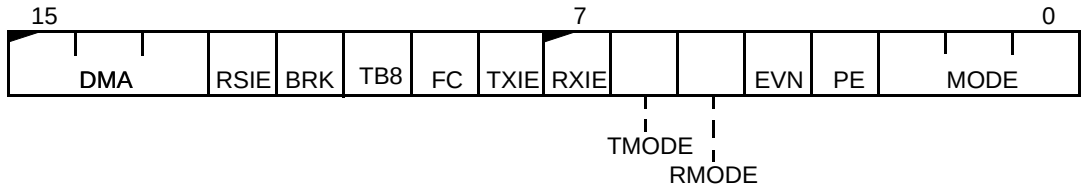
Serial Port 0 Transmit Register (SP0TD)
Page 10-12

82



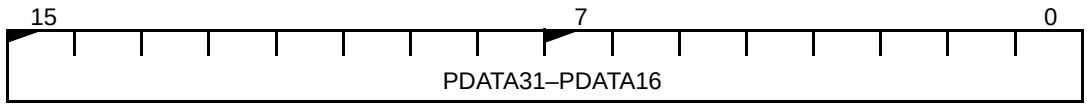
Serial Port 0 Status Register (SP0STS)
Page 10-10

80



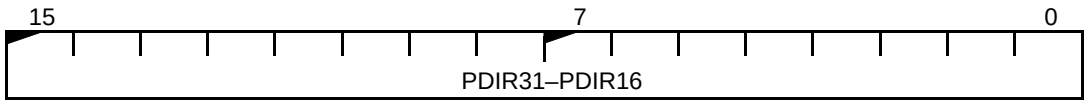
Serial Port 0 Control Register (SP0CT)
Page 10-5

7A



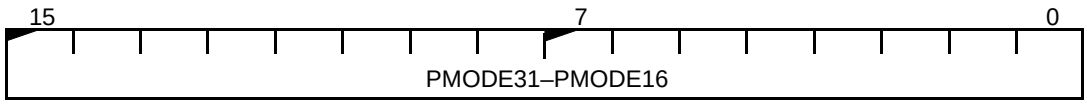
PIO Data 1 Register (PDATA1)
Page 11-5

78



PIO Direction 1 Register (PDIR1)
Page 11-4

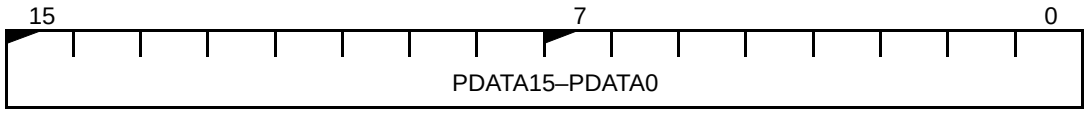
76



PIO Mode 1 Register (PIOMODE1)
Page 11-3

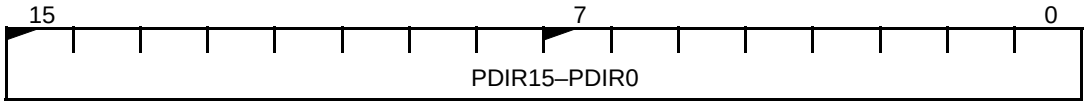
Figure A-1 Internal Register Summary (continued)

74



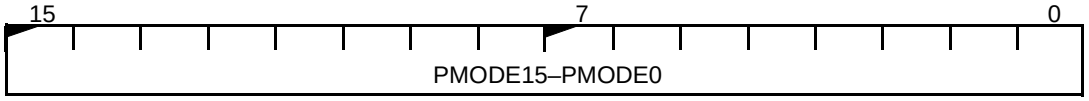
PIO Data 0 Register (PDATA0)
Page 11-5

72



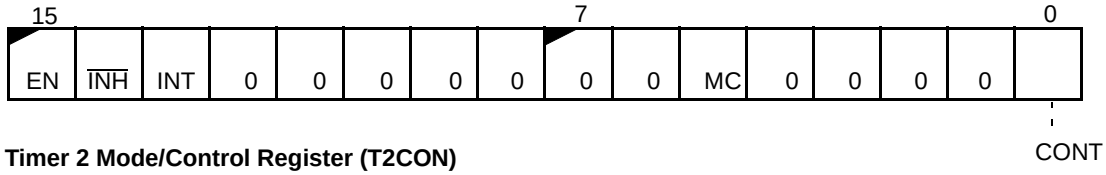
PIO Direction 0 Register (PDIR0)
Page 11-4

70



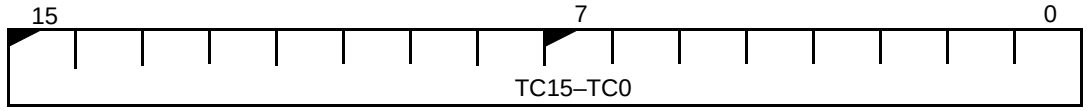
PIO Mode 0 Register (PIOMODE0)
Page 11-3

66



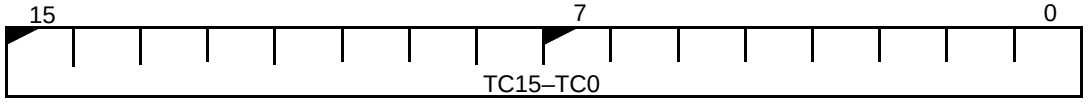
Timer 2 Mode/Control Register (T2CON)
Page 8-5

62



Timer 2 Maxcount Compare A Register (T2CMPA)
Page 8-7

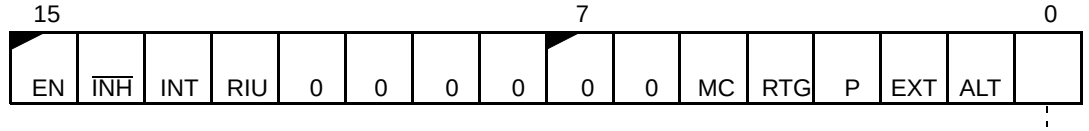
60



Timer 2 Count Register (T2CNT)
Page 8-6

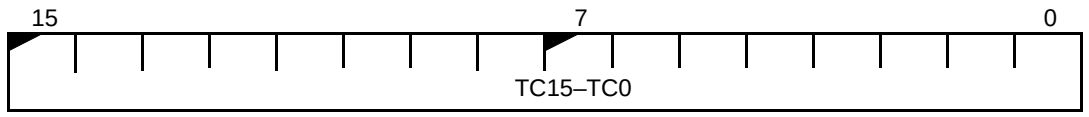
Figure A-1 Internal Register Summary (continued)

5E



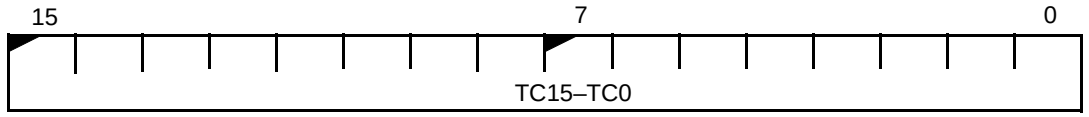
Timer 1 Mode/Control Register (T1CON)
Page 8-3

5C



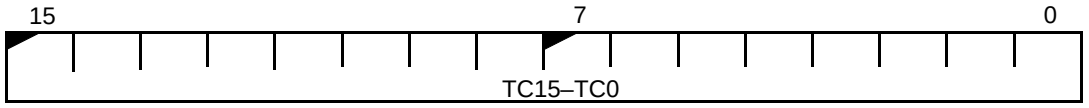
Timer 1 Maxcount Compare B Register (T1CMPB)
Page 8-7

5A



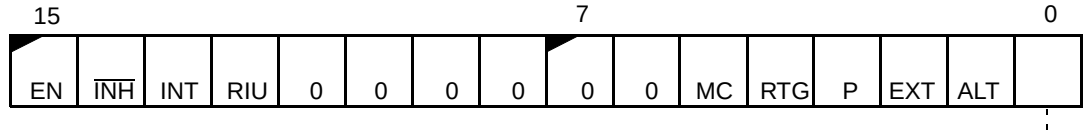
Timer 1 Maxcount Compare A Register (T1CMPA)
Page 8-7

58



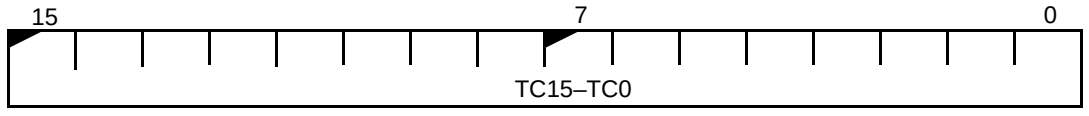
Timer 1 Count Register (T1CNT)
Page 8-6

56



Timer 0 Mode/Control Register (T0CON)
Page 8-3

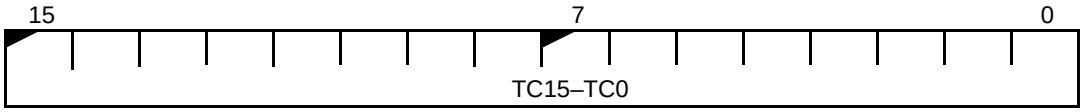
54



Timer 0 Maxcount Compare B Register (T0CMPB)
Page 8-7

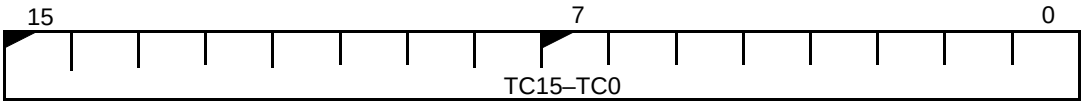
Figure A-1 Internal Register Summary (continued)

52



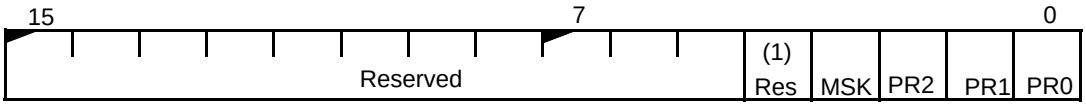
Timer 0 Maxcount Compare A Register (T0CMPA)
Page 8-7

50



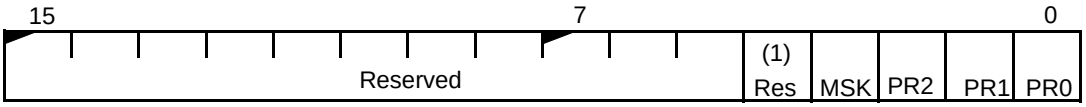
Timer 0 Count Register (T0CNT)
Page 8-6

44



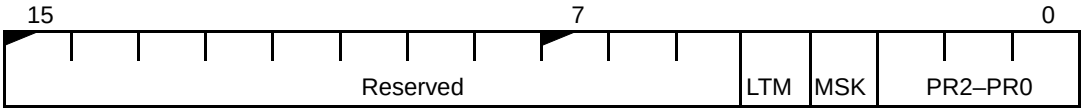
Serial Port 0 Interrupt Control Register (SP0CON)
Master Mode
Page 7-19

42



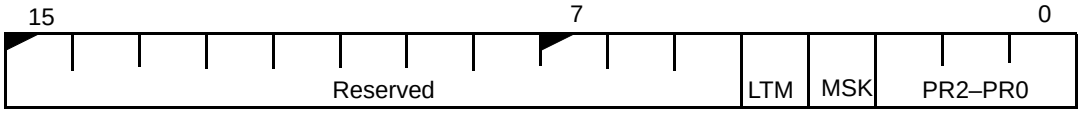
Serial Port 1 Interrupt Control Register (SP1CON)
Master Mode
Page 7-19

40



INT4 Control Register (I4CON)
Master Mode
Page 7-17

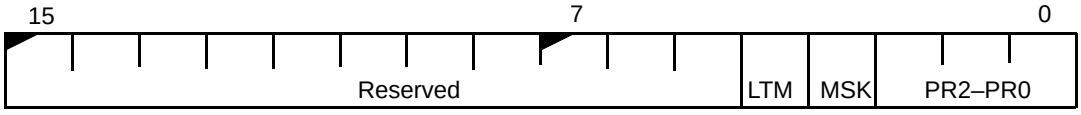
3E



INT3 Control Register (I3CON)
Master Mode
Page 7-16

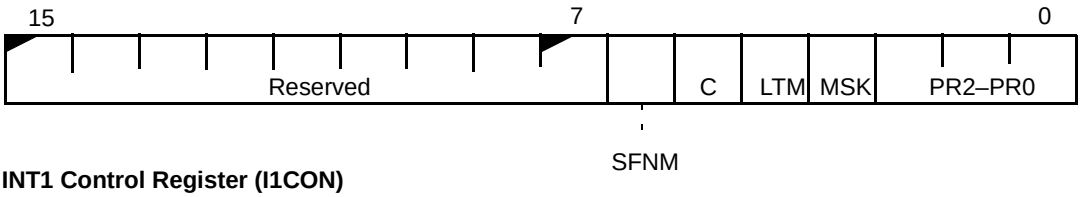
Figure A-1 Internal Register Summary (continued)

3C



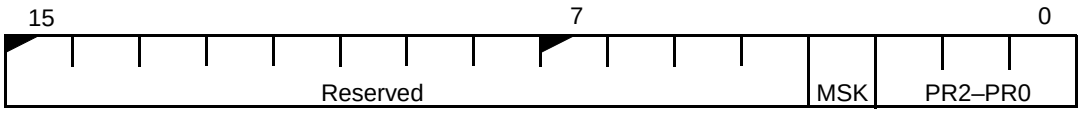
INT2 Control Register (I2CON)
Master Mode
Page 7-16

3A



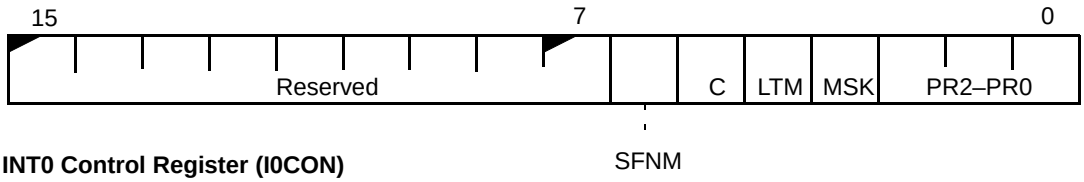
INT1 Control Register (I1CON)
Master Mode
Page 7-14

3A



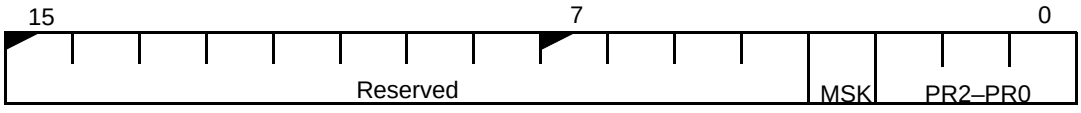
Timer 2 Interrupt Control Register (T2INTCON)
Slave Mode
Page 7-30

38



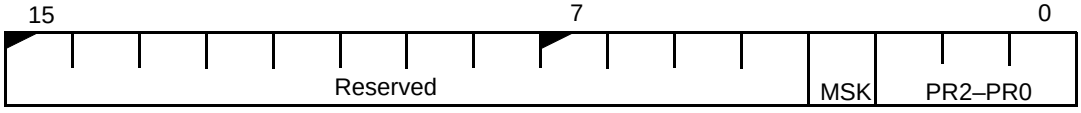
INT0 Control Register (I0CON)
Master Mode
Page 7-14

38



Timer 1 Interrupt Control Register (T1INTCON)
Slave Mode
Page 7-30

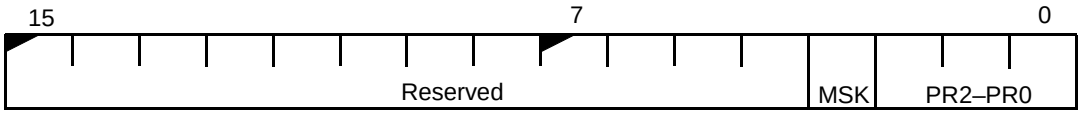
36



DMA 1 Interrupt Control Register (DMA1CON)/INT6CON
Master Mode—Page 7-18
Slave Mode—Page 7-30

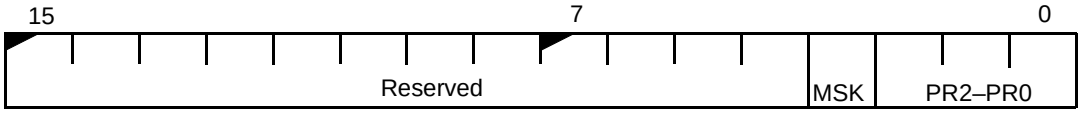
Figure A-1 Internal Register Summary (continued)

34



DMA 0 Interrupt Control Register (DMA0CON)/INT5CON
Master Mode—Page 7-18
Slave Mode—Page 7-30

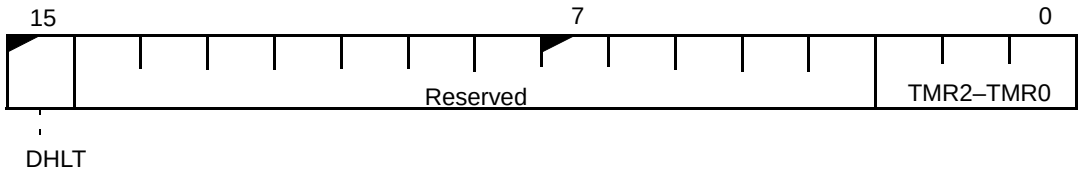
32



Timer Interrupt Control Register (TCUCON)
Master Mode—Page 7-18

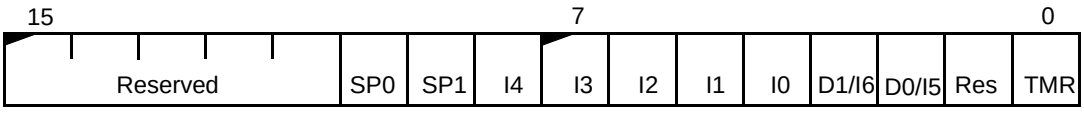
Timer 0 Interrupt Control Register (T0INTCON)
Slave Mode—Page 7-30

30



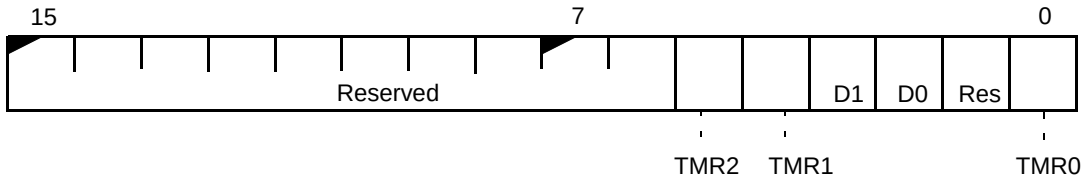
Interrupt Status Register (INTSTS)
Master Mode—Page 7-20
Slave Mode—Page 7-31

2E



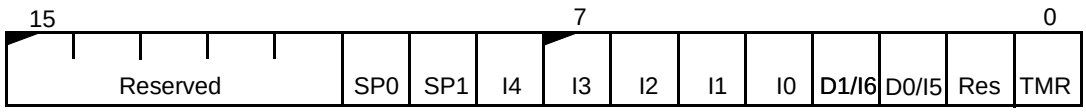
Interrupt Request Register (REQST)
Master Mode
Page 7-21

2E



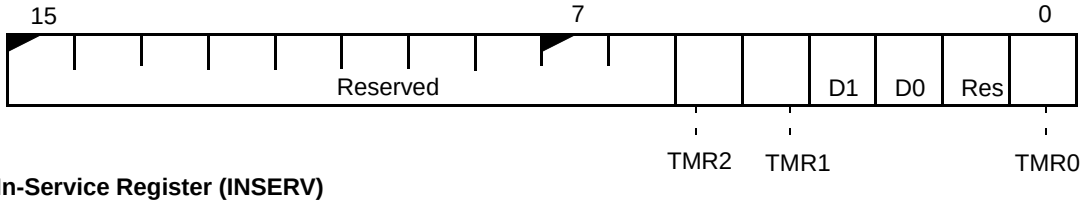
Interrupt Request Register (REQST)
Slave Mode
Page 7-32

Figure A-1 Internal Register Summary (continued)



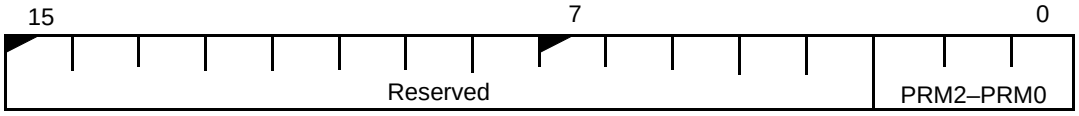
2C

Interrupt In-Service Register (INSERV)
Master Mode
Page 7-23



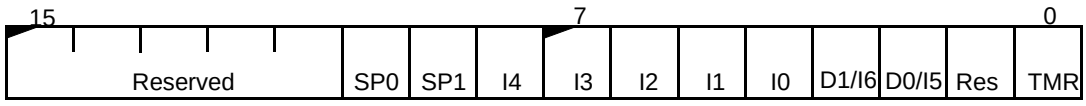
2C

In-Service Register (INSERV)
Slave Mode
Page 7-33



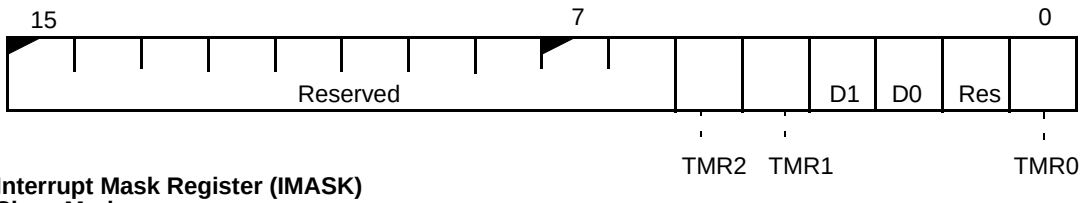
2A

Priority Mask Register (PRIMSK)
Master Mode—Page 7-24
Slave Mode—Page 7-34



28

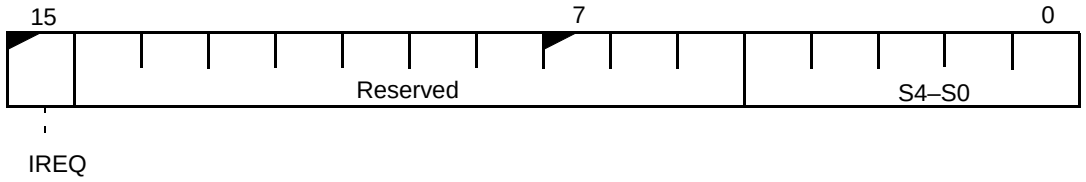
Interrupt Mask Register (IMASK)
Master Mode
Page 7-25



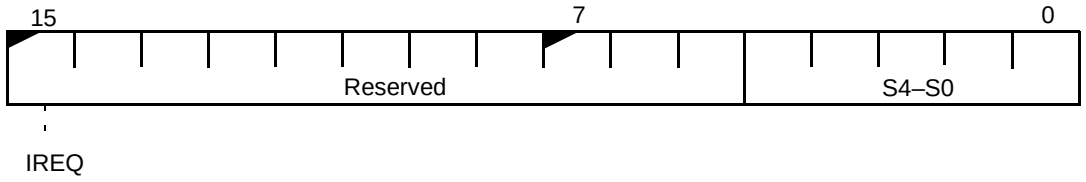
28

Interrupt Mask Register (IMASK)
Slave Mode
Page 7-35

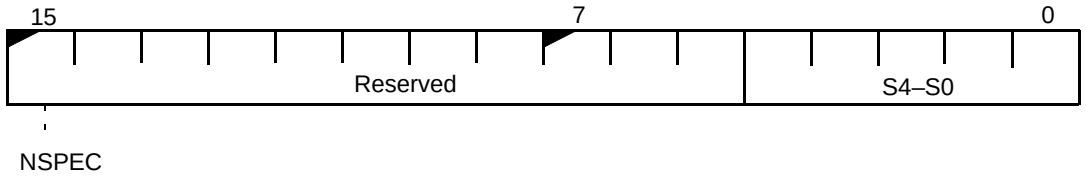
Figure A-1 Internal Register Summary (continued)



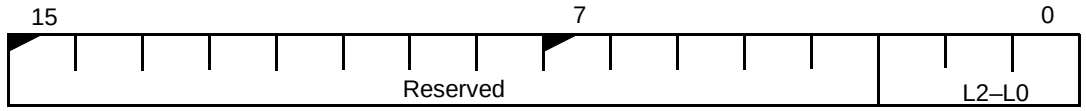
Poll Status Register (POLLST)
Master Mode
Page 7-26



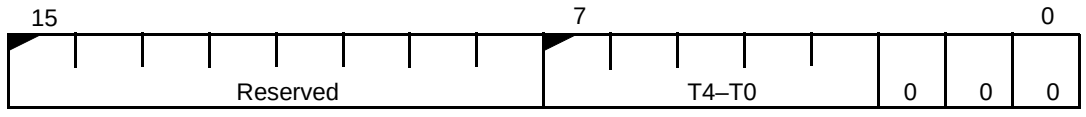
Poll Register (POLL)
Master Mode
Page 7-27



End-of-Interrupt Register (EOI)
Master Mode
Page 7-28



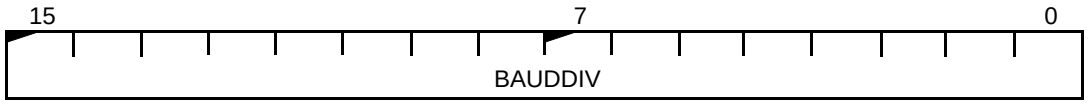
Specific End-of-Interrupt Register (EOI)
Slave Mode
Page 7-36



Interrupt Vector Register (INTVEC)
Slave Mode
Page 7-37

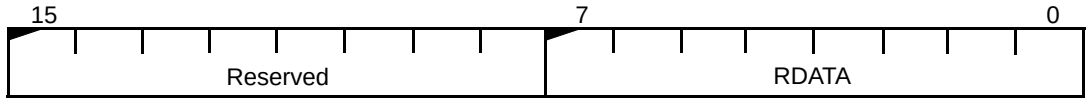
Figure A-1 Internal Register Summary (continued)

18



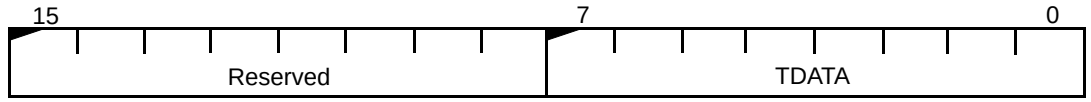
Serial Port 1 Baud Rate Divisor Register (SP1BAUD)
Page 10-15

16



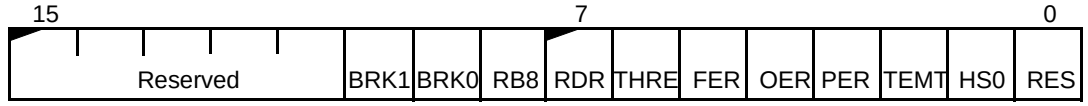
Serial Port 1 Receive Register (SP1RD)
Page 10-13

14



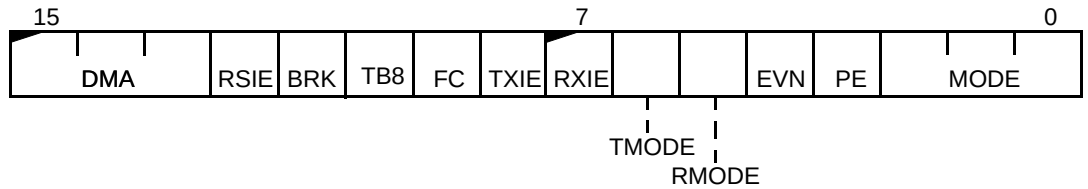
Serial Port 1 Transmit Register (SP1TD)
Page 10-12

12



Serial Port 1 Status Register (SP1STS)
Page 10-10

10



Serial Port 1 Control Register (SP1CT)
Page 10-5

INDEX

A

B

C

D

E

F

G

H

I

L

M

N

O

P

R

S

T

U

V

W

X

Z