

# Am186™ES and Am188™ES User's Manual

© 1997 Advanced Micro Devices, Inc. All rights reserved.

Advanced Micro Devices, Inc. ("AMD") reserves the right to make changes in its products without notice in order to improve design or performance characteristics.

The information in this publication is believed to be accurate at the time of publication, but AMD makes no representations or warranties with respect to the accuracy or completeness of the contents of this publication or the information contained herein, and reserves the right to make changes at any time, without notice. AMD disclaims responsibility for any consequences resulting from the use of the information included in this publication.

This publication neither states nor implies any representations or warranties of any kind, including but not limited to, any implied warranty of merchantability or fitness for a particular purpose. AMD products are not authorized for use as critical components in life support devices or systems without AMD's written approval. AMD assumes no liability whatsoever for claims associated with the sale or use (including the use of engineering samples) of AMD products except as provided in AMD's Terms and Conditions of Sale for such products.

## **Trademarks**

AMD, the AMD logo, and combinations thereof are trademarks of Advanced Micro Devices, Inc.

Am386 and Am486 are registered trademarks, and Am186, Am188, E86, AMD Facts-On-Demand, and K86 are trademarks of Advanced Micro Devices, Inc.

FusionE86 is a service mark of Advanced Micro Devices, Inc.

Product names used in this publication are for identification purposes only and may be trademarks of their respective companies.

---

## 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](mailto: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://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](mailto:webmaster@amd.com).

### Documentation and Literature

Free E86 family information such as data books, user's manuals, data sheets, application notes, the FusionE86<sup>SM</sup> Partner Solutions Catalog, and other literature is available with a simple phone call. Internationally, contact your local AMD sales office for complete E86 family literature.

### Literature Ordering

(800) 222-9323 toll-free for U.S. and Canada

(512) 602-5651 direct dial worldwide

(512) 602-7639 fax

(800) 222-9323 AMD Facts-On-Demand™ faxback service  
toll-free for U.S. and Canada



# TABLE OF CONTENTS

<b>PREFACE</b>	<b>INTRODUCTION AND OVERVIEW</b>	
	DESIGN PHILOSOPHY . . . . .	xiii
	PURPOSE OF THIS MANUAL . . . . .	xiii
	INTENDED AUDIENCE . . . . .	xiii
	USER'S MANUAL OVERVIEW . . . . .	xiii
	AMD DOCUMENTATION . . . . .	xiv
	E86 Family . . . . .	xiv
<b>CHAPTER 1</b>	<b>FEATURES AND PERFORMANCE</b>	
	1.1 KEY FEATURES AND BENEFITS . . . . .	1-1
	1.2 DISTINCTIVE CHARACTERISTICS . . . . .	1-2
	1.3 APPLICATION CONSIDERATIONS . . . . .	1-6
	1.3.1 Clock Generation . . . . .	1-6
	1.3.2 Memory Interface . . . . .	1-6
<b>CHAPTER 2</b>	<b>PROGRAMMING</b>	
	2.1 REGISTER SET . . . . .	2-1
	2.1.1 Processor Status Flags Register . . . . .	2-2
	2.2 MEMORY ORGANIZATION AND ADDRESS GENERATION . . . . .	2-3
	2.3 I/O SPACE . . . . .	2-4
	2.4 INSTRUCTION SET . . . . .	2-4
	2.5 SEGMENTS . . . . .	2-8
	2.6 DATA TYPES . . . . .	2-8
	2.7 ADDRESSING MODES . . . . .	2-10
<b>CHAPTER 3</b>	<b>SYSTEM OVERVIEW</b>	
	3.1 PIN DESCRIPTIONS . . . . .	3-1
	3.1.1 Pins That Are Used by Emulators . . . . .	3-19
	3.2 BUS OPERATION . . . . .	3-20
	3.3 BUS INTERFACE UNIT . . . . .	3-23
	3.3.1 Nonmultiplexed Address Bus . . . . .	3-23
	3.3.2 Static Bus Sizing . . . . .	3-23
	3.3.3 Byte Write Enables . . . . .	3-24
	3.3.4 Pseudo Static RAM (PSRAM) Support . . . . .	3-24
	3.4 CLOCK AND POWER MANAGEMENT UNIT . . . . .	3-25
	3.4.1 Phase-Locked Loop (PLL) . . . . .	3-25
	3.4.2 Crystal-Driven Clock Source . . . . .	3-25
	3.4.3 External Source Clock . . . . .	3-27
	3.4.4 System Clocks . . . . .	3-27
	3.4.5 Power-Save Operation . . . . .	3-27
<b>CHAPTER 4</b>	<b>PERIPHERAL CONTROL BLOCK</b>	
	4.1 OVERVIEW . . . . .	4-1
	4.1.1 Peripheral Control Block Relocation Register (RELREG, Offset FEh) . . . . .	4-3
	4.1.2 Reset Configuration Register (RESCON, Offset F6h) . . . . .	4-4
	4.1.3 Processor Release Level Register (PRL, Offset F4h) . . . . .	4-5
	4.1.4 Auxiliary Configuration Register (AUXCON, Offset F2h) . . . . .	4-6
	4.1.5 System Configuration Register (SYSCON, Offset F0h) . . . . .	4-7

	4.2	INITIALIZATION AND PROCESSOR RESET	4-8
<b>CHAPTER 5</b>		<b>CHIP SELECT UNIT</b>	
	5.1	OVERVIEW	5-1
	5.2	CHIP SELECT TIMING	5-2
	5.3	READY AND WAIT-STATE PROGRAMMING	5-2
	5.4	CHIP SELECT OVERLAP	5-2
	5.5	CHIP SELECT REGISTERS	5-3
	5.5.1	Upper Memory Chip Select Register (UMCS, Offset A0h)	5-4
	5.5.2	Low Memory Chip Select Register (LMCS, Offset A2h)	5-6
	5.5.3	Midrange Memory Chip Select Register (MMCS, Offset A6h)	5-8
	5.5.4	$\overline{\text{PCS}}$ and $\overline{\text{MCS}}$ Auxiliary Register (MPCS, Offset A8h)	5-10
	5.5.5	Peripheral Chip Select Register (PACS, Offset A4h)	5-12
<b>CHAPTER 6</b>		<b>REFRESH CONTROL UNIT</b>	
	6.1	OVERVIEW	6-1
	6.1.1	Memory Partition Register (MDRAM, Offset E0h)	6-1
	6.1.2	Clock Prescaler Register (CDRAM, Offset E2h)	6-2
	6.1.3	Enable RCU Register (EDRAM, Offset E4h)	6-2
	6.1.4	Watchdog Timer Control Register (WDTCON, Offset E6h)	6-3
<b>CHAPTER 7</b>		<b>INTERRUPT CONTROL UNIT</b>	
	7.1	OVERVIEW	7-1
	7.1.1	Definitions of Interrupt Terms	7-1
	7.1.2	Interrupt Conditions and Sequence	7-5
	7.1.3	Interrupt Priority	7-6
	7.1.4	Software Exceptions, Traps, and NMI	7-7
	7.1.5	Interrupt Acknowledge	7-8
	7.1.6	Interrupt Controller Reset Conditions	7-9
	7.2	MASTER MODE OPERATION	7-10
	7.2.1	Fully Nested Mode	7-10
	7.2.2	Cascade Mode	7-11
	7.2.3	Special Fully Nested Mode	7-12
	7.2.4	Operation in a Polled Environment	7-12
	7.2.5	End-of-Interrupt Write to the EOI Register	7-12
	7.3	MASTER MODE INTERRUPT CONTROLLER REGISTERS	7-13
	7.3.1	INT0 and INT1 Control Registers (I0CON, Offset 38h, I1CON, Offset 3Ah)	7-14
	7.3.2	INT2 and INT3 Control Registers (I2CON, Offset 3Ch, I3CON, Offset 3Eh)	7-15
	7.3.3	INT4 Control Register (I4CON, Offset 40h)	7-16
	7.3.4	Timer and DMA Interrupt Control Registers (TCUCON, Offset 32h, DMA0CON/INT5CON, Offset 34h, DMA1CON/INT6CON, Offset 36h)	7-17
	7.3.5	Serial Port 0/1 Interrupt Control Registers (SP0CON/SP1CON, Offset 44h/42h)	7-18
	7.3.6	Interrupt Status Register (INTSTS, Offset 30h)	7-19
	7.3.7	Interrupt Request Register (REQST, Offset 2Eh)	7-20
	7.3.8	Interrupt In-Service Register (INSERV, Offset 2Ch)	7-22
	7.3.9	Priority Mask Register (PRIMSK, Offset 2Ah)	7-23
	7.3.10	Interrupt Mask Register (IMASK, Offset 28h)	7-24
	7.3.11	Poll Status Register (POLLST, Offset 26h)	7-25
	7.3.12	Poll Register (POLL, Offset 24h)	7-26
	7.3.13	End-of-Interrupt Register (EOI, Offset 22h)	7-27
	7.4	SLAVE MODE OPERATION	7-28
	7.4.1	Slave Mode Interrupt Nesting	7-28
	7.4.2	Slave Mode Interrupt Controller Registers	7-28

	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) . . . . .	7-29
	7.4.4	Interrupt Status Register (INTSTS, Offset 30h) . . . . .	7-30
	7.4.5	Interrupt Request Register (REQST, Offset 2Eh) . . . . .	7-31
	7.4.6	Interrupt In-Service Register (INSERT, Offset 2Ch) . . . . .	7-32
	7.4.7	Priority Mask Register (PRIMSK, Offset 2Ah) . . . . .	7-33
	7.4.8	Interrupt Mask Register (IMASK, Offset 28h) . . . . .	7-34
	7.4.9	Specific End-of-Interrupt Register (EOI, Offset 22h) . . . . .	7-35
	7.4.10	Interrupt Vector Register (INTVEC, Offset 20h) . . . . .	7-36
<b>CHAPTER 8</b>	<b>TIMER CONTROL UNIT</b>		
	8.1	OVERVIEW . . . . .	8-1
	8.2	PULSE WIDTH DEMODULATION . . . . .	8-1
	8.3	PROGRAMMABLE REGISTERS . . . . .	8-2
	8.3.1	Timer Operating Frequency . . . . .	8-2
	8.3.2	Timer 0 and Timer 1 Mode and Control Registers (T0CON, Offset 56h, T1CON, Offset 5Eh) . . . . .	8-3
	8.3.3	Timer 2 Mode and Control Register (T2CON, Offset 66h) . . . . .	8-5
	8.3.4	Timer Count Registers (T0CNT, Offset 50h, T1CNT, Offset 58h, T2CNT, Offset 60h) . . . . .	8-6
	8.3.5	Timer Maxcount Compare Registers (T0CMPA, Offset 52h, T0CMPB, Offset 54h, T1CMPA, Offset 5Ah, T1CMPB, Offset 5Ch, T2CMPA, Offset 62h) . . . . .	8-7
<b>CHAPTER 9</b>	<b>DMA CONTROLLER</b>		
	9.1	OVERVIEW . . . . .	9-1
	9.2	DMA OPERATION . . . . .	9-1
	9.3	PROGRAMMABLE DMA REGISTERS . . . . .	9-3
	9.3.1	DMA Control Registers (D0CON, Offset CAh, D1CON, Offset DAh) . . . . .	9-3
	9.3.2	Serial Port/DMA Transfers . . . . .	9-5
	9.3.3	DMA Transfer Count Registers (D0TC, Offset C8h, D1TC, Offset D8h) . . . . .	9-6
	9.3.4	DMA Destination Address High Register (High Order Bits) (D0DSTH, Offset C6h, D1DSTH, Offset D6h) . . . . .	9-7
	9.3.5	DMA Destination Address Low Register (Low Order Bits) (D0DSTL, Offset C4h, D1DSTL, Offset D4h) . . . . .	9-8
	9.3.6	DMA Source Address High Register (High Order Bits) (D0SRCH, Offset C2h, D1SRCH, Offset D2h) . . . . .	9-9
	9.3.7	DMA Source Address Low Register (Low Order Bits) (D0SRCL, Offset C0h, D1SRCL, Offset D0h) . . . . .	9-10
	9.4	DMA REQUESTS . . . . .	9-11
	9.4.1	Synchronization Timing . . . . .	9-12
	9.4.2	DMA Acknowledge . . . . .	9-13
	9.4.3	DMA Priority . . . . .	9-13
	9.4.4	DMA Programming . . . . .	9-13
	9.4.5	DMA Channels on Reset . . . . .	9-14
<b>CHAPTER 10</b>	<b>ASYNCHRONOUS SERIAL PORTS</b>		
	10.1	OVERVIEW . . . . .	10-1
	10.1.1	Serial Port Flow Control . . . . .	10-1
	10.2	PROGRAMMABLE REGISTERS . . . . .	10-3
	10.2.1	Serial Port 0/1 Control Registers (SP0CT/SP1CT, Offset 80h/10h) . . . . .	10-5
	10.2.2	Serial Port 0/1 Status Registers (SP0STS/SP1STS, Offset 82h/12h) . . . . .	10-9

	10.2.3 Serial Port 0/1 Transmit Registers (SP0TD/SP1TD, Offset 84h/14h) . . . . .	10-11
	10.2.4 Serial Port 0/1 Receive Registers (SP0RD/SP1RD, Offset 86h/16h) . . . . .	10-12
	10.2.5 Serial Port 0/1 Baud Rate Divisor Registers (SP0BAUD/SP1BAUD, Offset 88h/18h) . . . . .	10-13
<b>CHAPTER 11</b>	<b>PROGRAMMABLE I/O PINS</b>	
	11.1 OVERVIEW . . . . .	11-1
	11.2 PIO MODE REGISTERS . . . . .	11-3
	11.2.1 PIO Mode 1 Register (PIOMODE1, Offset 76h) . . . . .	11-3
	11.2.2 PIO Mode 0 Register (PIOMODE0, Offset 70h) . . . . .	11-3
	11.3 PIO DIRECTION REGISTERS . . . . .	11-4
	11.3.1 PIO Direction 1 Register (PDIR1, Offset 78h) . . . . .	11-4
	11.3.2 PIO Direction 0 Register (PDIR0, Offset 72h) . . . . .	11-4
	11.4 PIO DATA REGISTERS . . . . .	11-5
	11.4.1 PIO Data Register 1 (PDATA1, Offset 7Ah) . . . . .	11-5
	11.4.2 PIO Data Register 0 (PDATA0, Offset 74h) . . . . .	11-5
	11.5 OPEN-DRAIN OUTPUTS . . . . .	11-5
<b>APPENDIX A</b>	<b>REGISTER SUMMARY</b>	
	<b>INDEX</b>	



**LIST OF FIGURES**

Figure 1-1	Am186ES Microcontroller Block Diagram . . . . .	1-4
Figure 1-2	Am188ES Microcontroller Block Diagram . . . . .	1-5
Figure 1-3	Basic Functional System Design . . . . .	1-6
Figure 2-1	Register Set . . . . .	2-2
Figure 2-2	Processor Status Flags Register (F) . . . . .	2-2
Figure 2-3	Physical Address Generation . . . . .	2-4
Figure 2-4	Memory and I/O Space . . . . .	2-4
Figure 2-5	Supported Data Types . . . . .	2-9
Figure 3-1	Am186ES Microcontroller Address Bus—Normal Read and Write Operation . . . . .	3-21
Figure 3-2	Am186ES Microcontroller—Read and Write with Address Bus Disable In Effect . . . . .	3-21
Figure 3-3	Am188ES Microcontroller Address Bus—Normal Read and Write Operation . . . . .	3-22
Figure 3-4	Am188ES Microcontroller—Read and Write with Address Bus Disable In Effect . . . . .	3-22
Figure 3-5	Oscillator Configurations . . . . .	3-26
Figure 3-6	Clock Organization . . . . .	3-27
Figure 4-1	Peripheral Control Block Relocation Register . . . . .	4-3
Figure 4-2	Reset Configuration Register . . . . .	4-4
Figure 4-3	Processor Release Level Register . . . . .	4-5
Figure 4-4	Auxiliary Configuration Register . . . . .	4-6
Figure 4-5	System Configuration Register . . . . .	4-7
Figure 5-1	Upper Memory Chip Select Register . . . . .	5-4
Figure 5-2	Low Memory Chip Select Register . . . . .	5-6
Figure 5-3	Midrange Memory Chip Select Register . . . . .	5-8
Figure 5-4	PCS and MCS Auxiliary Register . . . . .	5-10
Figure 5-5	Peripheral Chip Select Register . . . . .	5-12
Figure 6-1	Memory Partition Register . . . . .	6-1
Figure 6-2	Clock Prescaler Register . . . . .	6-2
Figure 6-3	Enable RCU Register . . . . .	6-2
Figure 6-4	Watchdog Timer Control Register . . . . .	6-3
Figure 7-1	External Interrupt Acknowledge Bus Cycles . . . . .	7-8
Figure 7-2	Fully Nested (Direct) Mode Interrupt Controller Connections . . . . .	7-10
Figure 7-3	Cascade Mode Interrupt Controller Connections . . . . .	7-11
Figure 7-4	INT0 and INT1 Control Registers . . . . .	7-14
Figure 7-5	INT2 and INT3 Control Registers . . . . .	7-15
Figure 7-6	INT4 Control Register . . . . .	7-16
Figure 7-7	Timer/DMA Interrupt Control Registers . . . . .	7-17
Figure 7-8	Serial Port 0/1 Interrupt Control Register . . . . .	7-18
Figure 7-9	Interrupt Status Register . . . . .	7-19
Figure 7-10	Interrupt Request Register . . . . .	7-20
Figure 7-11	Interrupt In-Service Register . . . . .	7-22
Figure 7-12	Priority Mask Register . . . . .	7-23
Figure 7-13	Interrupt Mask Register . . . . .	7-24
Figure 7-14	Poll Status Register . . . . .	7-25
Figure 7-15	Poll Register . . . . .	7-26
Figure 7-16	Example EOI Assembly Code . . . . .	7-27
Figure 7-17	End-of-Interrupt Register . . . . .	7-27
Figure 7-18	Timer and DMA Interrupt Control Registers . . . . .	7-29
Figure 7-19	Interrupt Status Register . . . . .	7-30
Figure 7-20	Interrupt Request Register . . . . .	7-31
Figure 7-21	Interrupt In-Service Register . . . . .	7-32
Figure 7-22	Priority Mask Register . . . . .	7-33
Figure 7-23	Interrupt Mask Register . . . . .	7-34
Figure 7-24	Specific End-of-Interrupt Register . . . . .	7-35
Figure 7-25	Interrupt Vector Register . . . . .	7-36

Figure 8-1	Typical Waveform Behavior . . . . .	8-1
Figure 8-2	Timer 0 and Timer 1 Mode and Control Registers . . . . .	8-3
Figure 8-3	Timer 2 Mode and Control Register . . . . .	8-5
Figure 8-4	Timer Count Registers . . . . .	8-6
Figure 8-5	Timer Maxcount Compare Registers . . . . .	8-7
Figure 9-1	DMA Unit Block Diagram . . . . .	9-2
Figure 9-2	DMA Control Registers . . . . .	9-3
Figure 9-3	DMA Transfer Count Registers . . . . .	9-6
Figure 9-4	DMA Destination Address High Register . . . . .	9-7
Figure 9-5	DMA Destination Address Low Register . . . . .	9-8
Figure 9-6	DMA Source Address High Register . . . . .	9-9
Figure 9-7	DMA Source Address Low Register . . . . .	9-10
Figure 9-8	Source-Synchronized DMA Transfers . . . . .	9-12
Figure 9-9	Destination Synchronized DMA Transfers . . . . .	9-13
Figure 10-1	DCE/DTE Protocol . . . . .	10-2
Figure 10-2	CTS/RTR Protocol . . . . .	10-3
Figure 10-3	Serial Port Control Register . . . . .	10-5
Figure 10-4	Serial Port 0/1 Status Register . . . . .	10-9
Figure 10-5	Serial Port 0/1 Transmit Registers . . . . .	10-11
Figure 10-6	Serial Port Receive 0/1 Registers . . . . .	10-12
Figure 10-7	Serial Port 0/1 Baud Rate Divisor Registers . . . . .	10-14
Figure 11-1	Programmable I/O Pin Operation . . . . .	11-1
Figure 11-2	PIO Mode 0 Register (PIOMODE0, offset 70h) . . . . .	11-3
Figure 11-3	PIO Mode 1 Register (PIOMODE1, offset 76h) . . . . .	11-3
Figure 11-4	PIO Direction 1 Register (PDIR1, offset 78h) . . . . .	11-4
Figure 11-5	PIO Direction 0 Register (PDIR0, offset 72h) . . . . .	11-4
Figure 11-6	PIO Data 1 Register (PDAT1, offset 7Ah) . . . . .	11-5
Figure 11-7	PIO Data 0 Register (PDAT0, offset 74h) . . . . .	11-5
Figure A-1	Internal Register Summary . . . . .	A-4

**LIST OF TABLES**

Table 2-1	Instruction Set . . . . .	2-5
Table 2-2	Segment Register Selection Rules . . . . .	2-8
Table 2-3	Memory Addressing Mode Examples . . . . .	2-10
Table 3-1	Numeric PIO Pin Designations. . . . .	3-13
Table 3-2	Alphabetic PIO Pin Designations . . . . .	3-14
Table 3-3	Programming Am186ES Microcontroller Bus Width . . . . .	3-24
Table 4-1	Peripheral Control Block Register Map . . . . .	4-2
Table 4-2	Processor Release Level (PRL) Values. . . . .	4-5
Table 4-3	Initial Register State After Reset . . . . .	4-9
Table 5-1	Chip Select Register Summary . . . . .	5-1
Table 5-2	UMCS Block Size Programming Values . . . . .	5-4
Table 5-3	LMCS Block Size Programming Values. . . . .	5-6
Table 5-4	MCS Block Size Programming. . . . .	5-10
Table 5-5	PCS Address Ranges . . . . .	5-13
Table 5-6	PCS3–PCS0 Wait-State Encoding. . . . .	5-13
Table 6-1	Watchdog Timer COUNT Settings. . . . .	6-4
Table 6-2	Watchdog Timer Duration . . . . .	6-4
Table 7-1	Am186ES and Am188ES Microcontroller Interrupt Types. . . . .	7-4
Table 7-2	Interrupt Controller Registers in Master Mode . . . . .	7-13
Table 7-3	Priority Level . . . . .	7-18
Table 7-4	Priority Level . . . . .	7-23
Table 7-5	Interrupt Controller Registers in Slave Mode . . . . .	7-28
Table 7-6	Priority Level . . . . .	7-33
Table 8-1	Timer Control Unit Register Summary . . . . .	8-2
Table 9-1	DMA Controller Register Summary . . . . .	9-1
Table 9-2	Synchronization Type. . . . .	9-4
Table 9-3	Maximum DMA Transfer Rates . . . . .	9-11
Table 10-1	Serial Port External Pins . . . . .	10-2
Table 10-2	Asynchronous Serial Port Register Summary . . . . .	10-4
Table 10-3	DMA Control Bits . . . . .	10-5
Table 10-4	Serial Port MODE Settings. . . . .	10-7
Table 10-5	Common Baud Rates. . . . .	10-13
Table 11-1	PIO Pin Assignments . . . . .	11-2
Table 11-2	PIO Mode and PIO Direction Settings . . . . .	11-3
Table A-1	Internal Register Summary . . . . .	A-1



## INTRODUCTION AND OVERVIEW

---

### DESIGN PHILOSOPHY

The Am186™ES and Am188™ES 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 Am186ES and Am188ES microcontrollers eliminate the need for off-chip system-interface logic. It is possible to implement a fully functional system with ROM and RAM, serial interfaces, and custom I/O capability without additional system-interface logic.

The Am186ES and Am188ES microcontrollers can operate at frequencies up to 40 MHz. And, the microcontrollers include an on-board PLL so that the input clock can be one-to-one with the internal processor clock. The Am186ES and Am188ES microcontrollers are available in versions operating at 20, 25, 33, and 40 MHz.

The Am186ES and Am188ES microcontrollers provide a natural migration path for 80C186/188 designs that need performance and cost enhancements.

### PURPOSE OF THIS MANUAL

This manual describes the technical features and programming interface of the Am186ES and Am188ES 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 Am186ES and Am188ES microcontrollers.

### USER'S MANUAL OVERVIEW

This manual contains information on the Am186ES and Am188ES 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 11 chapters and 1 appendix.

- Chapter 1 introduces the **features and performance** aspects of the Am186ES and Am188ES 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 **refresh control unit**.
- Chapter 7 provides a description of the **on-chip interrupt controller**.

- Chapter 8 describes the **timer control unit**.
- Chapter 9 describes the **DMA controller**.
- Chapter 10 describes the **asynchronous serial 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 Am186ES and Am188ES microcontroller pin lists, timing, thermal characteristics, and physical dimensions, please refer to the *Am186ES/ESLV and Am188ES/ESLV Microcontrollers Data Sheet* (order# 20002).

## AMD DOCUMENTATION

### E86 Family

ORDER NO.	DOCUMENT TITLE
-----------	----------------

<b>20002</b>	<b>Am186ES/ESLV and Am188ES/ESLV Microcontrollers Data Sheet</b> Hardware documentation: pin descriptions, functional descriptions, absolute maximum ratings, operating ranges, switching characteristics and waveforms, connection diagrams and pinouts, and package physical dimensions.
<b>21267</b>	<b>Am186 and Am188 Family Instruction Set Manual</b> Provides a detailed description and examples for each instruction included in the Am186 and Am188 Family Instruction Set.
<b>19255</b>	<b>FusionE86<sup>SM</sup> Catalog</b> Provides information on tools that speed an E86 family embedded product to market. Includes products from expert suppliers of embedded development solutions.
<b>20071</b>	<b>E86 Family Support Tools Brief</b> Lists available E86 family software and hardware development tools, as well as contact information for suppliers.
<b>21058</b>	<b>FusionE86 Development Tools Reference CD</b> Provides a single-source multimedia tool for customer evaluation of AMD products, as well as Fusion partner tools and technologies that support the E86 family of microcontrollers and microprocessors. Technical documentation for the E86 family is included on the CD in PDF format.

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

Literature is also available in postscript and PDF formats on the AMD web site. To access the AMD home page, go to <http://www.amd.com>. To download documents and software, ftp to <ftp://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>.

Compared to the 80C186/188 microcontrollers, the Am186<sup>TM</sup>ES and Am188<sup>TM</sup>ES microcontrollers enable designers to reduce the cost, size, and power consumption of embedded systems, while increasing performance and functionality. The Am186ES and Am188ES microcontrollers are cost-effective, enhanced versions of the AMD 80C186/188.

The Am186ES and Am188ES microcontrollers are the ideal upgrade for 80C186/188 designs requiring 80C186/188-compatibility, increased performance, serial communications, and a glueless bus interface. Developed exclusively for the embedded marketplace, the Am186ES and Am188ES microcontrollers increase the performance of existing 80C186/188 systems while decreasing their cost.

Because the Am186ES and Am188ES microcontrollers integrate on-chip peripherals and offer up to twice the performance of an 80C186/188, they are ideal upgrade solutions for customers requiring more integration and performance than their present x86 solution delivers.

## 1.1 KEY FEATURES AND BENEFITS

The Am186ES and Am188ES microcontrollers extend the AMD family of microcontrollers based on the industry-standard x86 architecture. The Am186ES and Am188ES microcontrollers are higher-performance, more integrated versions of the 80C186/188 microprocessors, offering an attractive migration path. In addition, the Am186ES and Am188ES microcontrollers offer application-specific features that can enhance the system functionality of the Am186EM and Am188EM microcontrollers. Upgrading to the Am186ES and Am188ES microcontrollers is an attractive solution for several reasons:

- **Minimized total system cost**—New peripherals and on-chip system interface logic on the Am186ES and Am188ES microcontrollers reduce the cost of existing 80C186/188 designs.
- **x86 software compatibility**—80C186/188-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 Am186ES and Am188ES microcontrollers increase the performance of 80C186/188 systems, and the nonmultiplexed address bus offers faster, unbuffered access to memory.
- **Enhanced functionality**—The new and enhanced on-chip peripherals of the Am186ES and Am188ES microcontrollers include 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 PSRAM controller, a 16-bit reset configuration register, and enhanced chip-select functionality.

The Am186ES and Am188ES 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, Am186EMLV, Am188EMLV, Am186EM, Am188EM, Am186ESLV, Am188ESLV, Am186ER, Am188ER, ÉlanSC300, ÉlanSC310, ÉlanSC400, Am386<sup>®</sup>SX, Am386DX, and Am486<sup>®</sup>DX microprocessors and microcontrollers.

## DISTINCTIVE CHARACTERISTICS

A block diagram of each microcontroller is shown in Figure 1-1 and Figure 1-2. The Am186ES microcontroller uses a 16-bit external bus, while the Am188ES microcontroller has an 8-bit external bus.

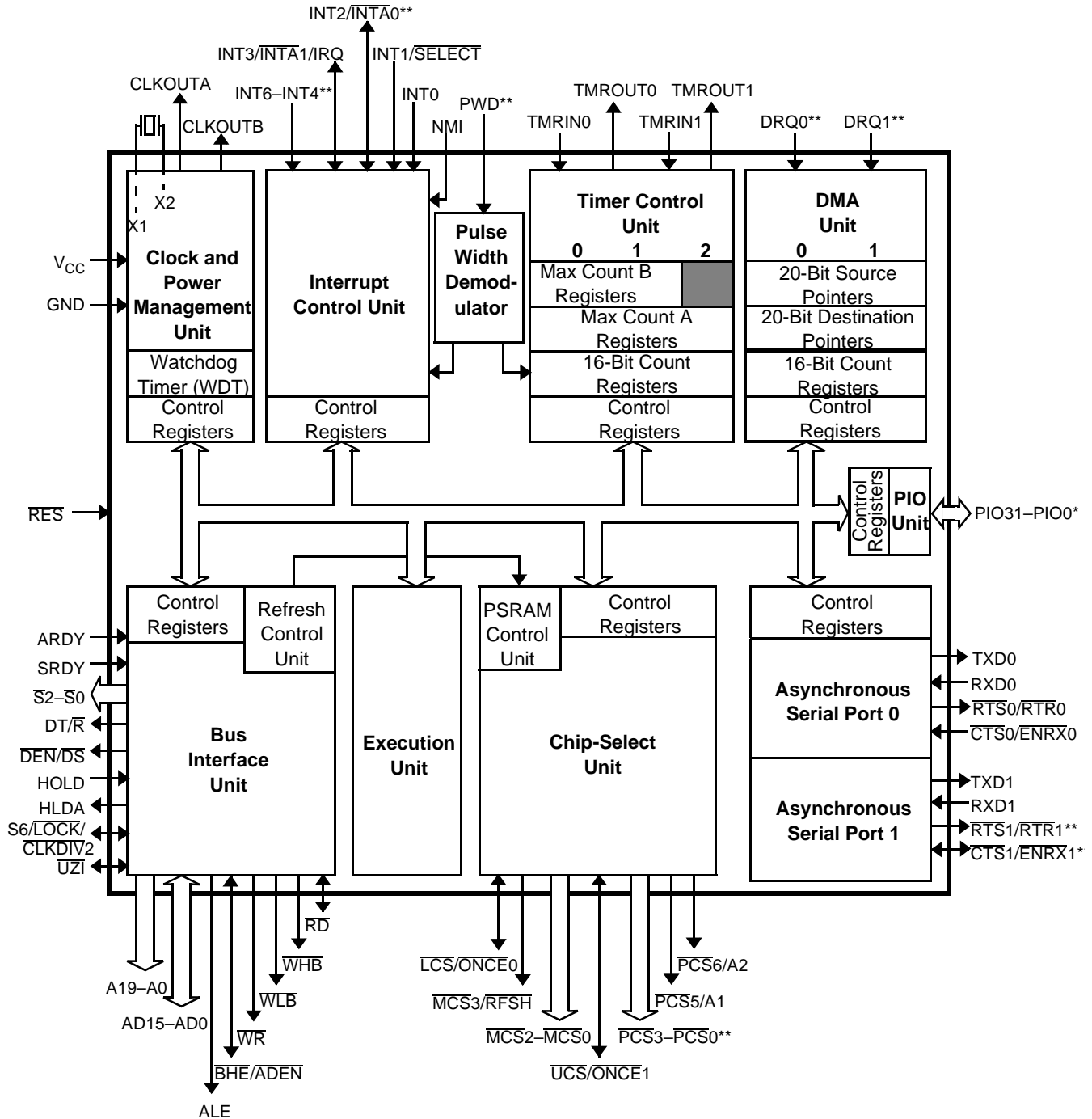
The Am186ES and Am188ES microcontrollers include the following features:

- E86™ family 80C186- and 80C188-compatible microcontrollers with enhanced bus interface
  - Lower system cost with higher performance
- 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
  - On the Am186ES microcontroller, 8-bit or 16-bit memory and I/O static bus option
- 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  $\overline{\text{CTS}}$ ,  $\text{RTS}$ ,  $\overline{\text{ENRX}}$ , 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 included for  $\overline{\text{DEN}}$
  - Pseudo Static RAM (PSRAM) controller includes auto refresh capability
  - Reset configuration register
- 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™EM and Am188™EM microcontrollers
  - Available in the following packages:
    - 100-pin, thin quad flat pack (TQFP)
    - 100-pin, plastic quad flat pack (PQFP)

Figure 1-1 Am186ES Microcontroller Block Diagram

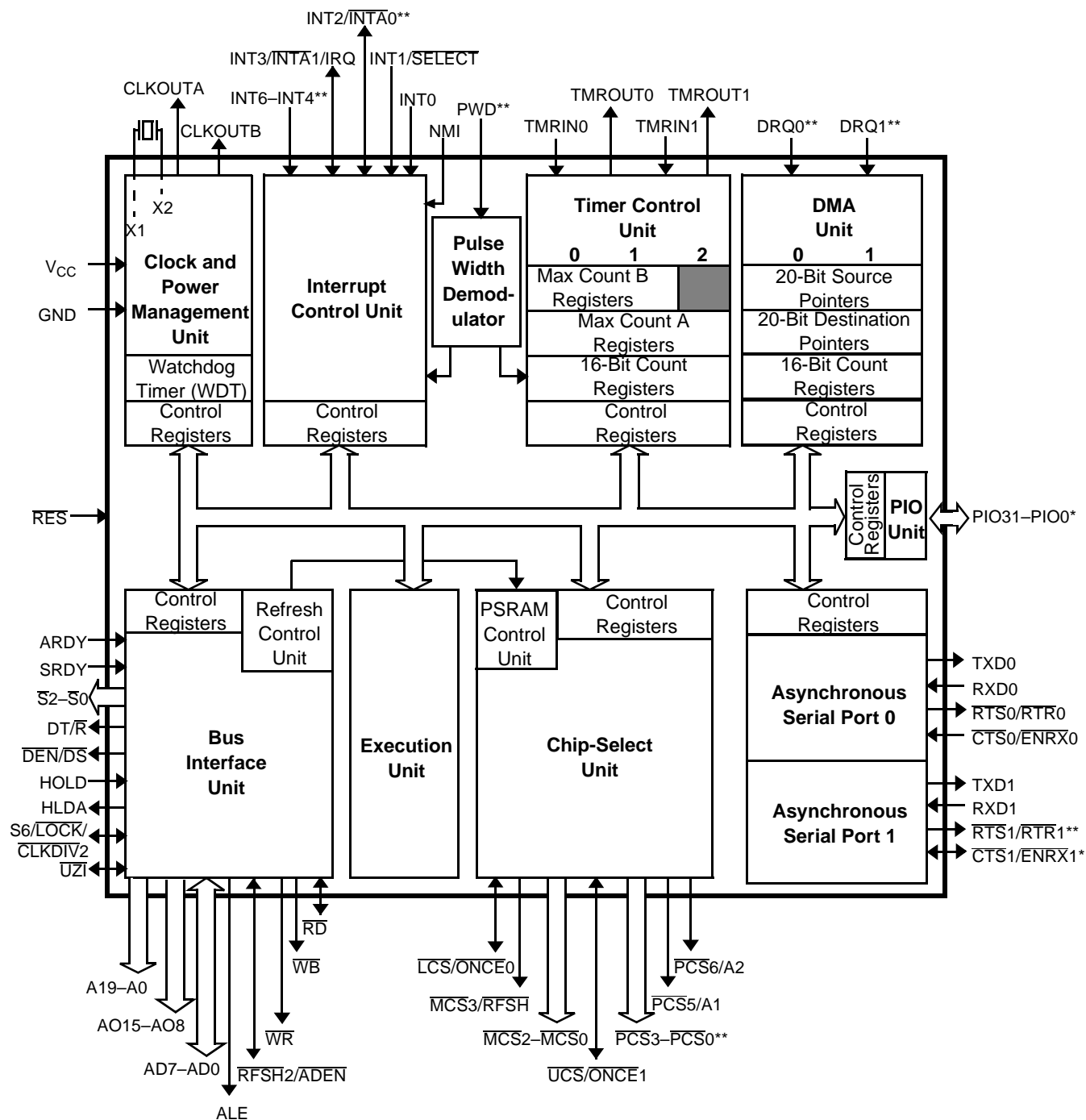


**Notes:**

\*All PIO signals are shared with other physical pins. See the pin descriptions in Chapter 3 and Table 3-1 on page 3-13 for information on shared functions.

\*\* PWD, INT5, INT6, RTS1/RTR1, and CTS1/ENRX1 are multiplexed with INT2/INTA0, DRQ0, DRQ1, PCS3, and PCS2 respectively. See the pin descriptions in Chapter 3.

Figure 1-2 Am188ES Microcontroller 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 3-13 for information on shared functions.

\*\* PWD, INT5, INT6, RTS1/RTR1, and CTS1/ENRX1 are multiplexed with INT2/INTA0, DRQ0, DRQ1, PCS3, and PCS2 respectively. See the pin descriptions in Chapter 3.

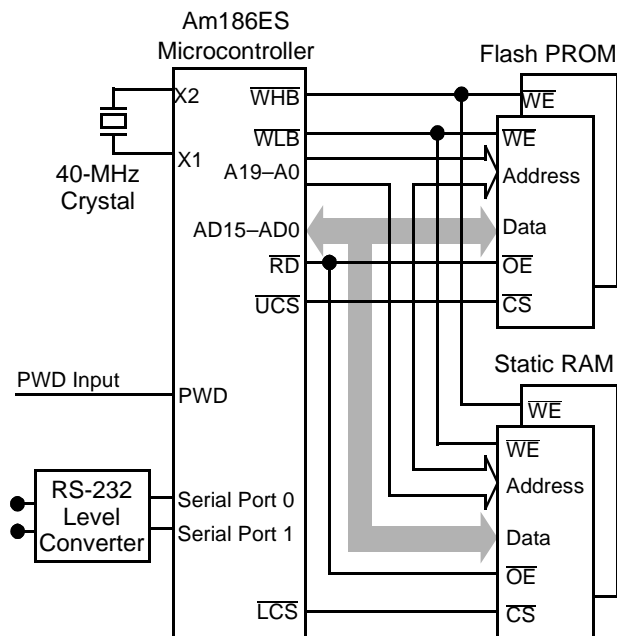
### 1.3 APPLICATION CONSIDERATIONS

The integration enhancements of the Am186ES and Am188ES 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.

The nonmultiplexed address bus is available in addition to the 186/188 multiplexed address/data bus (AD15–AD0). The two buses can operate simultaneously or the AD15–AD0 bus can be configured to operate only during the data phase of a bus cycle. See the  $\overline{\text{BHE}}/\text{ADEN}$  and  $\overline{\text{RFSH2}}/\text{ADEN}$  pin descriptions in Chapter 3, and see section 5.5.1 on page 5-4 for additional information regarding the AD15–AD0 address enabling and disabling.

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

Figure 1-3 Basic Functional System Design



#### 1.3.1 Clock Generation

The integrated clock generation circuitry of the Am186ES and Am188ES microcontrollers allows the use of a times-one crystal frequency. The design in Figure 1-3 achieves 40-MHz CPU operation, while using a 40-MHz crystal.

#### 1.3.2 Memory Interface

The integrated memory controller logic of the Am186ES and Am188ES microcontrollers provides a direct address bus interface to memory devices. The use of an external address latch controlled by the address latch enable (ALE) signal is not required.

Individual byte write-enable signals are provided to eliminate the need for external high/low-byte, write-enable circuitry. The maximum bank size programmable for the memory

chip-select signals is increased to 512 Kbytes to facilitate the use of high-density memory devices.

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

Figure 1-3 illustrates an Am186ES microcontroller-based SRAM configuration. The memory interface requires the following:

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

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

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



All members of the Am186 and Am188 family of microcontrollers, including the Am186ES and Am188ES, contain the same basic set of registers, instructions, and addressing modes, and are compatible with the original industry-standard 186/188 parts.

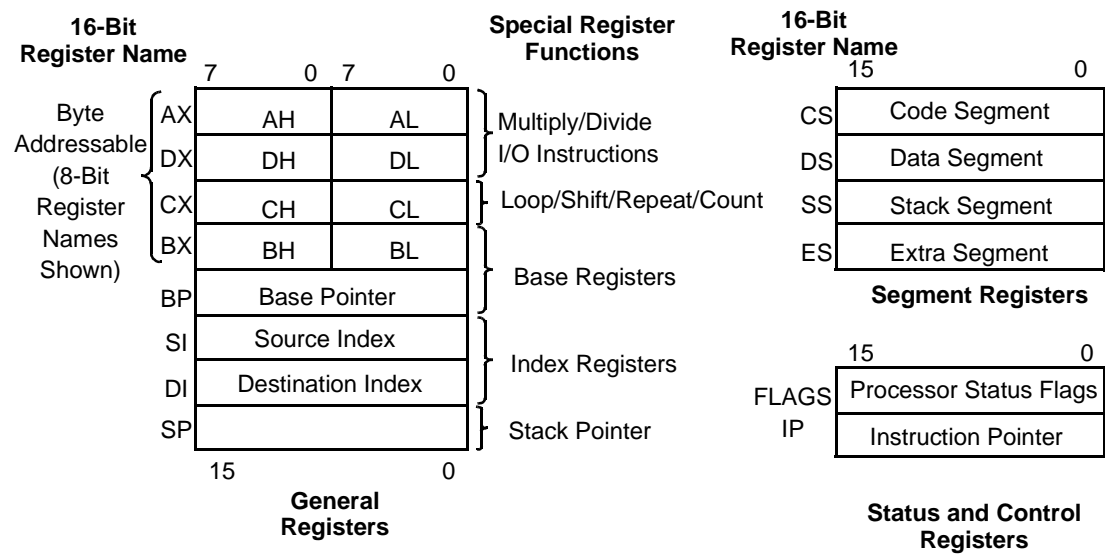
## 2.1 REGISTER SET

The base architecture of the Am186ES and Am188ES 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 Am186ES and Am188ES 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

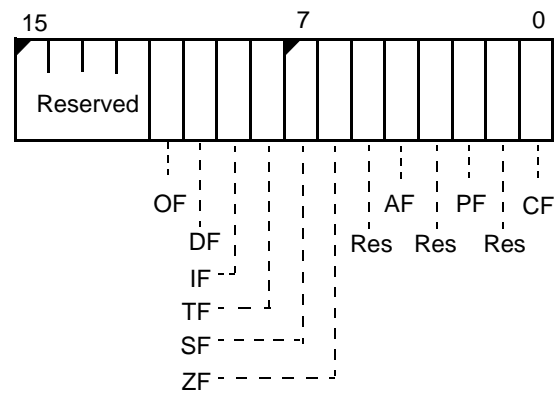


### 2.1.1 Processor Status Flags Register

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.

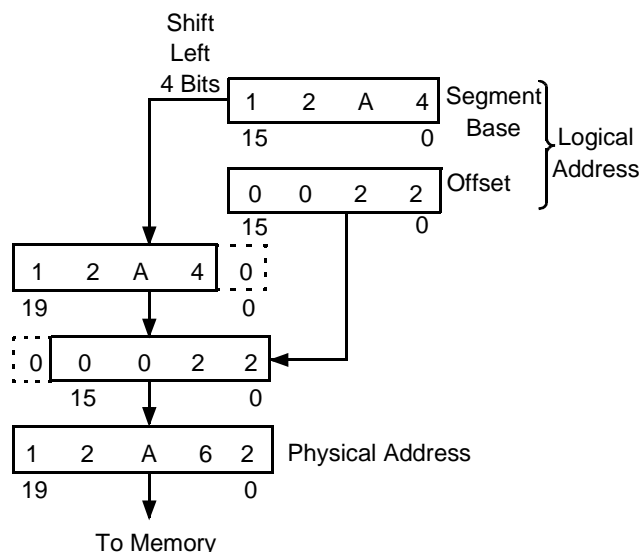
For example, if the segment register is loaded with 12A4h and the offset is 0022h, the resultant address is 12A62h (see Figure 2-3). To find the result:

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).

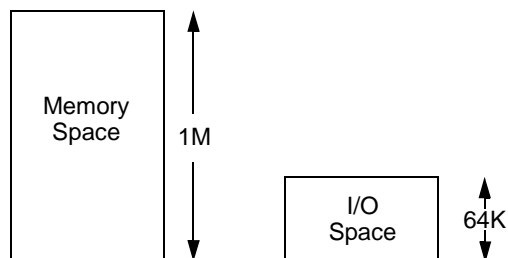
Figure 2-3 Physical Address Generation



## 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 Am186ES and Am188ES microcontrollers provide specific instructions for addressing I/O space.

Figure 2-4 Memory and I/O Space



## 2.4 INSTRUCTION SET

The instruction set used by the Am186ES and Am188ES microcontrollers is identical to the 80C186/188 instruction set. An instruction can reference from zero to several operands. An operand can reside in a register, in the instruction itself, or in memory. Specific operand addressing modes are discussed later in this section.

Table 2-1 lists the instructions for the Am186ES and Am188ES microcontrollers in alphabetical order.

Table 2-1 Instruction Set

<b>Mnemonic</b>	<b>Instruction Name</b>
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
JNBE	Jump if above/not below or equal
JAE/JNB	Jump if above or equal/not below

<b>Mnemonic</b>	<b>Instruction Name</b>
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

<b>Mnemonic</b>	<b>Instruction Name</b>
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 Am186ES and Am188ES 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

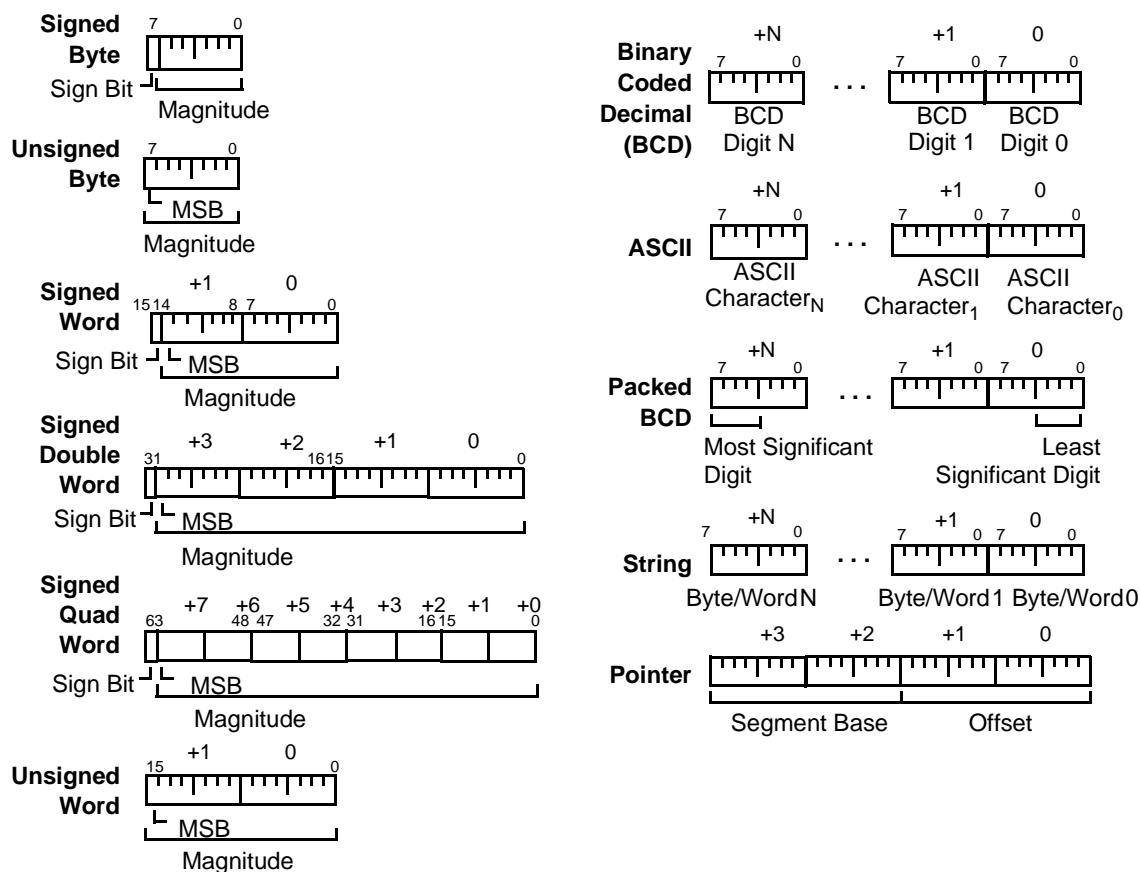
The Am186ES and Am188ES microcontrollers directly support the following data types:

- **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 Am186ES and Am188ES microcontrollers.

Figure 2-5 Supported Data Types



## ADDRESSING MODES

The Am186ES and Am188ES 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	<code>mov ax, ds:4</code>
Register Indirect	<code>mov ax, [si]</code>
Based	<code>mov ax, [bx]4</code>
Indexed	<code>mov ax, [si]4</code>
Based Indexed	<code>mov ax, [si][bx]</code>
Based Indexed with Displacement	<code>mov ax, [si][bx]4</code>



This chapter contains descriptions of the Am186ES and Am188ES 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 on the 186 or AO15–AO8 and AD7–AD0 on the 188). During a bus hold or reset condition, the address bus is in a high-impedance state.

**AD15–AD8****Address and Data Bus, Am186ES Microcontroller Only  
(input/output, three-state, synchronous, level-sensitive)**

**AD15–AD8**—On the Am186ES microcontroller, 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 for the 186, AO15–AO8 and AD7–AD0 for the 188) can also be used to load system configuration information into the internal reset configuration register.

**AO15–AO8****Address-Only Bus, Am188ES Microcontroller  
(output, three-state, synchronous, level-sensitive)**

**AO15–AO8**—On the Am188ES microcontroller, the address-only bus (AO15–AO8) contains valid high-order address bits from bus cycles  $t_1$ – $t_4$ . When address generation is disabled ( $AD = 1$ ), the address on this bus is not valid during  $t_1$ . These outputs are floated during a bus hold or reset.

On the Am188ES microcontroller, AO15–AO8 combine with AD7–AD0 to form a complete multiplexed address bus while AD7–AD0 is the 8-bit data bus.

**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 on the Am188ES microcontroller, AD7–AD0 supplies the data.

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

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

During a power-on reset, the address and data bus pins (AD15–AD0 for the 186, AO15–AO8 and AD7–AD0 for the 188) 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 for the 186 or AO15–AO8 and AD7–AD0 for the 188). The address is guaranteed to be valid on the trailing edge of ALE. This pin is three-stated during ONCE mode. This pin is not three-stated during a bus hold or 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 may 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.

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

**Bus High Enable, Am186ES Microcontroller Only  
(three-state, output, synchronous)**  
**Address Enable, Am188ES Microcontroller Only  
(input, internal pullup)**

**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  $\overline{\text{BHE/ADEN}}$  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	Refresh

$\overline{\text{BHE}}$  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.

On the Am186ES microcontroller,  $\overline{\text{WLB}}$  and  $\overline{\text{WHB}}$  implement the functionality of  $\overline{\text{BHE}}$  and AD0 for high and low byte write enables.

$\overline{\text{BHE/ADEN}}$  also signals DRAM refresh cycles when using the multiplexed address and data (AD) bus. A refresh cycle is indicated when both  $\overline{\text{BHE/ADEN}}$  and AD0 are High. During refresh cycles, the A bus and the AD bus are not guaranteed to provide the same address during the address phase of the AD bus cycle. For this reason, the A0 signal cannot be used in place of the AD0 signal to determine refresh cycles. PSRAM refreshes also provide an additional  $\overline{\text{RFSH}}$  signal (see the  $\overline{\text{MCS3/RFSH}}$  pin description on page 3-9).

**ADEN**—If  $\overline{\text{BHE/ADEN}}$  is held High or left floating during power-on reset, the address portion of the AD bus (AD15–AD0 for the 186 or AO15–AO8 and AD7–AD0 for the 188) is enabled or disabled during LCS and UCS bus cycles based on the DA bit in the LMCS and UMCS registers. If the DA bit is set, the memory address is accessed on the A19–A0 pins. There is a weak internal pullup resistor on  $\overline{\text{BHE/ADEN}}$  so no external pullup is required. This mode of operation reduces power consumption.

If  $\overline{\text{BHE/ADEN}}$  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 13.)

**Note:** On the Am188ES microcontroller, AO15–AO8 are driven 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 crystal input frequency (X1), the power-save frequency, or is three-stated. 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 crystal input frequency (X1), the power-save frequency, or is three-stated. CLKOUTB remains active during reset and bus hold conditions.

CLKOUTB is not used for AC timing specs.

## CTS0/ENRX0/PIO21

### **Clear-to-Send 0 (input, asynchronous)**

### **Enable-Receiver-Request 0 (input, asynchronous)**

**CTS0**—This pin provides the Clear to Send signal for asynchronous serial port 0 when the  $\overline{\text{ENRX0}}$  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  $\overline{\text{CTS0}}$  signal gates the transmission of data from the associated serial port transmit register. When  $\overline{\text{CTS0}}$  is asserted, the transmitter will begin transmission of a frame of data, if any is available. If  $\overline{\text{CTS0}}$  is deasserted, the transmitter holds the data in the serial port transmit register. The value of  $\overline{\text{CTS0}}$  is checked only at the beginning of the transmission of the frame.

**ENRX0**—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  $\overline{\text{ENRX0}}$  signal enables the receiver for the associated serial port.

## DEN/ $\overline{\text{DS}}$ /PIO5

### **Data Enable (output, three-state, synchronous)**

### **Data Strobe (output, three-state, synchronous)**

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

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

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

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

This pin indicates to the microcontroller that an external bus master needs control of the local bus.

The Am186ES and Am188ES microcontrollers' HOLD latency time, that is, the time between HOLD request and HOLD acknowledge, is a function of the activity occurring in the processor when the HOLD request is received. A HOLD request is second only to DRAM refresh

requests in priority of activity requests received by the processor. This implies that if a HOLD request is received just as a DMA transfer begins, the HOLD latency can be as great as 4 bus cycles. This occurs if a DMA word transfer operation is taking place (Am186ES microcontroller only) from an odd address to an odd address. This is a total of 16 clock cycles or more if wait states are required. In addition, if locked transfers are performed, the HOLD latency time is increased by the length of the locked transfer.

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/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.

**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 **SELECT** pin indicates to the microcontroller that the interrupt type appears on the bus.

## INT2/INTA0/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 INTA0 when INT0 is configured in cascade mode.

**INTA0**—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/ $\overline{\text{INTA0}}$ /PWD/PIO31 is reflected in the PIO data register for PIO31 as if it was a PIO.

#### INT3/ $\overline{\text{INTA1}}$ /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  $\overline{\text{INTA1}}$  when INT1 is configured in cascade mode.

**INTA1**—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.

#### **$\overline{\text{LCS}}/\overline{\text{ONCE0}}$**

#### **Lower Memory Chip Select (output, synchronous, internal pullup) ONCE Mode Request 0 (input)**

**$\overline{\text{LCS}}$** —This pin indicates to the system that a memory access is in progress to the lower memory block. The base address and size of the lower memory block are programmable up to 512 Kbytes. On the Am186ES microcontroller,  $\overline{\text{LCS}}$  is configured for 8-bit or 16-bit bus size by the auxiliary configuration register.  $\overline{\text{LCS}}$  is held High during a bus hold condition.

**$\overline{\text{ONCE0}}$** —During reset, this pin and  $\overline{\text{ONCE1}}$  indicate to the microcontroller the mode in which it should operate.  $\overline{\text{ONCE0}}$  and  $\overline{\text{ONCE1}}$  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,  $\overline{\text{ONCE0}}$  has a weak internal pullup resistor that is active only during reset. This pin is not three-stated during a bus hold condition.

#### **$\overline{\text{MCS0}}$**

#### **( $\overline{\text{MCS0}}$ /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 mid-range memory block are programmable. On the Am186ES microcontroller,  $\overline{\text{MCS0}}$  is configured for 8-bit or 16-bit bus size by the auxiliary configuration register.  $\overline{\text{MCS0}}$  is held High during a bus hold condition. In addition, it has weak internal pullup resistors that are active during reset.

This signal functions like the corresponding signal in the Am186EM and Am188EM microcontrollers except that  $\overline{\text{MCS0}}$  can be programmed as the chip select for the entire middle chip select address range.

#### **$\overline{\text{MCS2}}\text{--}\overline{\text{MCS1}}$**

#### **( $\overline{\text{MCS2}}$ /PIO24, $\overline{\text{MCS1}}$ /PIO15)**

#### **Midrange Memory Chip Selects (output, synchronous, internal pullup)**

These pins indicate to the system that a memory access is in progress to the corresponding region of the midrange memory block. The base address and size of the midrange memory block are programmable. On the Am186ES microcontroller,  $\overline{\text{MCS2}}\text{--}\overline{\text{MCS1}}$  are configured for 8-bit or 16-bit bus size by the auxiliary configuration register.  $\overline{\text{MCS2}}\text{--}\overline{\text{MCS1}}$  are held High during a bus hold condition. In addition, they have weak internal pullup resistors that are active during reset.

These signals function like the signals in the Am186EM and Am188EM microcontrollers except that if  $\overline{\text{MCS0}}$  is programmed to be active for the entire middle chip-select range, then these signals are available as PIOs.



If they are not programmed as PIOs and if  $\overline{MCS0}$  is programmed for the whole middle chip-select range, these signals operate normally.

### **$\overline{MCS3}/\overline{RFSH}/\text{PIO25}$**

#### **Midrange Memory Chip Select 3 (output, synchronous, internal pullup) Automatic Refresh (output, synchronous)**

**$\overline{MCS3}$** —This pin indicates to the system that a memory access is in progress to the fourth region of the midrange memory block. The base address and size of the midrange memory block are programmable. On the Am186ES microcontroller,  $\overline{MCS3}$  is configured for 8-bit or 16-bit bus size by the auxiliary configuration register.  $\overline{MCS3}$  is held High during a bus hold condition. In addition, this pin has a weak internal pullup resistor that is active during reset.

This signal functions like the corresponding signal in the Am186EM and Am188EM microcontrollers except that if  $\overline{MCS0}$  is programmed for the entire middle chip-select range, then this signal is available as a PIO. If  $\overline{MCS3}$  is not programmed as a PIO and if  $\overline{MCS0}$  is programmed for the entire middle chip-select range, this signal operates normally. Depending on chip configuration, this signal can serve as a memory  $\overline{RFSH}$ .

**$\overline{RFSH}$** —This pin provides a signal timed for auto refresh to PSRAM or DRAM devices. It is only enabled to function as a refresh pulse when the PSRAM or DRAM mode bit is set. An active Low pulse is generated for 1.5 clock cycles with an adequate deassertion period to ensure that overall auto refresh cycle time is met.

This signal functions like the  $\overline{RFSH}$  signal in the Am186EM and Am188EM microcontrollers except that the DRAM row address is not driven on DRAM refreshes. This pin is not three-stated during a bus hold condition.

### **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 reenabled by software in the NMI interrupt service routine, via the STI instruction for example, the fact that an NMI is currently in service will not have any effect on the priority resolution of maskable interrupt requests. For this reason, it is strongly advised that the interrupt service routine for NMI does not enable the maskable interrupts.

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

**$\overline{PCS1}$ – $\overline{PCS0}$**

**( $\overline{PCS1}$ /PIO17,  $\overline{PCS0}$ /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.  $\overline{PCS3}$ – $\overline{PCS0}$  are held High during a bus hold condition. They are also held High 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.

**$\overline{PCS2}$ /CTS1/ENRX1/PIO18**

**Peripheral Chip Select 2 (output, synchronous)**

**Clear-to-Send 1 (input, asynchronous)**

**Enable-Receiver-Request 1 (input, asynchronous)**

**$\overline{PCS2}$** —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  $\overline{PCS2}$  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.  $\overline{PCS2}$  is held High during a bus hold or reset condition.

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.

**$\overline{CTS1}$** —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  $\overline{CTS1}$  signal gates the transmission of data from the associated serial port transmit register. When  $\overline{CTS1}$  is asserted, the transmitter will begin transmission of a frame of data, if any is available. If  $\overline{CTS1}$  is deasserted, the transmitter holds the data in the serial port transmit register. The value of  $\overline{CTS1}$  is checked only at the beginning of the transmission of the frame.

**$\overline{ENRX1}$** —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  $\overline{ENRX1}$  signal enables the receiver for the associated serial port.

**PCS3/RTS1/RTR1/PIO19****Peripheral Chip Select 3 (output, synchronous)****Ready-to-Send 1 (output, asynchronous)****Ready-to-Receive 1 (output, asynchronous)**

**PCS3**—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  $\overline{\text{PCS3}}$  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.  $\overline{\text{PCS3}}$  is held High during a bus hold or reset condition.

Unlike the  $\overline{\text{UCS}}$  and  $\overline{\text{LCS}}$  chip selects, the  $\overline{\text{PCS}}$  outputs assert with the multiplexed AD address bus. Note 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.

**RTS1**—This pin provides the Ready to Send signal for asynchronous serial port 1 when the RTS1 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  $\overline{\text{RTS1}}$  signal is asserted when the associated serial port transmit register contains data which has not been transmitted.

**RTR1**—This pin provides the Ready to Receive signal for asynchronous serial port 1 when the RTS1 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  $\overline{\text{RTR1}}$  signal is asserted when the associated serial port receive register does not contain valid, unread data.

**PCS5/A1/PIO3****Peripheral Chip Select 5 (output, synchronous)****Latched Address Bit 1 (output, synchronous)**

**PCS5**—This pin indicates to the system that a memory access is in progress to the sixth region of the peripheral memory block (either I/O or memory address space). The base address of the peripheral memory block is programmable.  $\overline{\text{PCS5}}$  is held High during a bus hold condition. It is also held High 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.

**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.

**PCS6/A2/PIO2****Peripheral Chip Select 6 (output, synchronous)****Latched Address Bit 2 (output, synchronous)**

**PCS6**—This pin indicates to the system that a memory access is in progress to the seventh region of the peripheral memory block (either I/O or memory address space). The base address of the peripheral memory block is programmable.  $\overline{\text{PCS6}}$  is held High during a bus hold condition or 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.

**A2**—When the EX bit in the MCS and 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 Am186ES and Am188ES 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  $\overline{DT/R}$ ,  $\overline{DEN}$ , and SRDY pins also default to normal operation on power-on reset.

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	$\overline{\text{PCS6/A2}}$	Input with pullup
3	$\overline{\text{PCS5/A1}}$	Input with pullup
4	DT/ $\overline{\text{R}}$	Normal operation <sup>(3)</sup>
5	$\overline{\text{DEN/DS}}$	Normal operation <sup>(3)</sup>
6	SRDY	Normal operation <sup>(4)</sup>
7 <sup>(1)</sup>	A17	Normal operation <sup>(3)</sup>
8 <sup>(1)</sup>	A18	Normal operation <sup>(3)</sup>
9 <sup>(1)</sup>	A19	Normal operation <sup>(3)</sup>
10	TMROUT0	Input with pulldown
11	TMRIN0	Input with pullup
12	DRQ0/INT5	Input with pullup
13	DRQ1/INT6	Input with pullup
14	$\overline{\text{MCS0}}$	Input with pullup
15	$\overline{\text{MCS1}}$	Input with pullup
16	$\overline{\text{PCS0}}$	Input with pullup
17	$\overline{\text{PCS1}}$	Input with pullup
18	$\overline{\text{PCS2/CTS1/ENRX1}}$	Input with pullup
19	$\overline{\text{PCS3/RTS1/RTR1}}$	Input with pullup
20	$\overline{\text{RTS0/RTR0}}$	Input with pullup
21	$\overline{\text{CTS0/ENRX0}}$	Input with pullup
22	TXD0	Input with pullup
23	RXD0	Input with pullup
24	$\overline{\text{MCS2}}$	Input with pullup
25	$\overline{\text{MCS3/RFSH}}$	Input with pullup
26 <sup>(1,2)</sup>	UZI	Input with pullup
27	TXD1	Input with pullup
28	RXD1	Input with pullup
29 <sup>(1,2)</sup>	S6/LOCK/CLKDIV2	Input with pullup
30	INT4	Input with pullup
31	INT2/INTA0/PWD	Input with pullup

**Notes:**

1. These pins are used by emulators. (Emulators also use  $\overline{\text{S2-S0}}$ ,  $\overline{\text{RES}}$ , NMI, CLKOUTA,  $\overline{\text{BHE}}$ , ALE, AD15–AD0, and A16–A0.)
2. These pins revert to normal operation if  $\overline{\text{BHE/ADEN}}$  (186) or  $\overline{\text{RFSH2/ADEN}}$  (188) 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 <sup>(1)</sup>	7	Normal operation <sup>(3)</sup>
A18 <sup>(1)</sup>	8	Normal operation <sup>(3)</sup>
A19 <sup>(1)</sup>	9	Normal operation <sup>(3)</sup>
CTS0/ENRX0	21	Input with pullup
DEN/DS	5	Normal operation <sup>(3)</sup>
DRQ0/INT5	12	Input with pullup
DRQ1/INT6	13	Input with pullup
DT/ $\overline{R}$	4	Normal operation <sup>(3)</sup>
INT2/INTA0/PWD	31	Input with pullup
INT4	30	Input with pullup
MCS0	14	Input with pullup
MCS1	15	Input with pullup
MCS2	24	Input with pullup
MCS3/RFSH	25	Input with pullup
$\overline{PCS}$ 0	16	Input with pullup
$\overline{PCS}$ 1	17	Input with pullup
PCS2/CTS1/ENRX1	18	Input with pullup
$\overline{PCS}$ 3/RTS1/ $\overline{RTR}$ 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/LOCK/CLKDIV2 <sup>(1,2)</sup>	29	Input with pullup
SRDY	6	Normal operation <sup>(4)</sup>
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
$\overline{UZI}$ <sup>(1,2)</sup>	26	Input with pullup

Notes:

1. These pins are used by emulators. (Emulators also use  $\overline{S}2$ – $\overline{S}0$ ,  $\overline{RES}$ , NMI, CLKOUTA,  $\overline{BHE}$ , ALE, AD15–AD0, and A16–A0.)
2. These pins revert to normal operation if  $\overline{BHE}/\overline{ADEN}$  (186) or  $\overline{RFSH2}/\overline{ADEN}$  (188) 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.

<b><math>\overline{RD}</math></b>	<p><b>Read Strobe (output, synchronous, three-state)</b></p> <p><b><math>\overline{RD}</math></b>—This pin indicates to the system that the microcontroller is performing a memory or I/O read cycle. <math>\overline{RD}</math> is guaranteed to not be asserted before the address and data bus is floated during the address-to-data transition. <math>\overline{RD}</math> floats during a bus hold condition.</p>
<b><math>\overline{RES}</math></b>	<p><b>Reset (input, asynchronous, level-sensitive)</b></p> <p>This pin requires the microcontroller to perform a reset. When <math>\overline{RES}</math> is asserted, the microcontroller immediately terminates its present activity, clears its internal logic, and CPU control is transferred to the reset address, FFFF0h.</p> <p><math>\overline{RES}</math> must be held Low for at least 1 ms.</p> <p><math>\overline{RES}</math> can be asserted asynchronously to CLKOUTA because <math>\overline{RES}</math> is synchronized internally. For proper initialization, <math>V_{CC}</math> must be within specifications, and CLKOUTA must be stable for more than four CLKOUTA periods during which <math>\overline{RES}</math> is asserted.</p> <p>The microcontroller begins fetching instructions approximately 6.5 CLKOUTA periods after <math>\overline{RES}</math> is deasserted. This input is provided with a Schmitt trigger to facilitate power-on <math>\overline{RES}</math> generation via an RC network.</p>
<b><math>\overline{RFSH2/ADEN}</math></b>	<p><b>(Am188ES Microcontroller Only)</b></p> <p><b>Refresh 2 (three-state, output, synchronous)</b></p> <p><b>Address Enable (input, internal pullup)</b></p> <p><b><math>\overline{RFSH2}</math></b>—Asserted Low to signify a DRAM refresh bus cycle. The use of <math>\overline{RFSH2/ADEN}</math> to signal a refresh is not valid when PSRAM mode is selected. Instead, the <math>\overline{MCS3/RFSH}</math> signal is provided to the PSRAM.</p> <p><b><math>\overline{ADEN}</math></b>—If <math>\overline{RFSH2/ADEN}</math> is held High or left floating on power-on reset, the AD bus (AO15–AO8 and AD7–AD0) is enabled or disabled during the address portion of LCS and UCS bus cycles based on the DA bit in the LMCS and UMCS registers. If the DA bit is set, the memory address is accessed on the A19–A0 pins. This mode of operation reduces power consumption. There is a weak internal pullup resistor on <math>\overline{RFSH2/ADEN}</math> so no external pullup is required.</p> <p>If <math>\overline{RFSH2/ADEN}</math> 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 one crystal clock cycle after the rising edge of <math>\overline{RES}</math>. <math>\overline{RFSH2/ADEN}</math> is three-stated during bus holds and ONCE mode.</p>
<b><math>\overline{RTS0/RTR0}/PIO20</math></b>	<p><b>Ready-to-Send 0 (output, asynchronous)</b></p> <p><b>Ready-to-Receive 0 (output, asynchronous)</b></p> <p><b><math>\overline{RTS0}</math></b>—This pin provides the Ready to Send signal for asynchronous serial port 0 when the RTS0 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 <math>\overline{RTS0}</math> signal is asserted when the associated serial port transmit register contains data which has not been transmitted.</p> <p><b><math>\overline{RTR0}</math></b>—This pin provides the Ready to Receive signal for asynchronous serial port 0 when the RTS0 bit in the AUXCON register is 0 and</p>

hardware flow control is enabled for the port (FC bit in the serial port 0 control register is set). The RTR0 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.

**$\overline{S}2\text{--}\overline{S}0$**

**Bus Cycle Status (output, three-state, synchronous)**

These pins indicate to the system the type of bus cycle in progress.  $\overline{S}2$  can be used as a logical memory or I/O indicator, and  $\overline{S}1$  can be used as a data transmit or receive indicator.  $\overline{S}2\text{--}\overline{S}0$  float during bus hold and hold acknowledge conditions. The  $\overline{S}2\text{--}\overline{S}0$  pins are encoded as shown.

$\overline{S}2$	$\overline{S}1$	$\overline{S}0$	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/LOCK/CLKDIV2/PIO29**

**Bus Cycle Status Bit 6 (output, synchronous)**

**Bus Lock (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.

**LOCK**—This signal is asserted Low to indicate to other system bus masters that they are not to gain control of the system bus. This signal is only available during  $t_1$ .

$\overline{LOCK}$  on the Am186ES and Am188ES microcontrollers does not conform to the timing of the  $\overline{LOCK}$  signal on the 80C186/188 microcontrollers. This signal is primarily intended for use by emulators.

**CLKDIV2**—If S6/CLKDIV2/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  $\overline{RES}$ .

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/CLKDIV2/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/ $\overline{\text{INTA0}}$ /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/ $\overline{\text{INTA0}}$ /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.

**$\overline{UCS}/\overline{ONCE1}$** 
**Upper Memory Chip Select (output, synchronous)  
ONCE Mode Request 1 (input, internal pullup)**

**$\overline{UCS}$** —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.  $\overline{UCS}$  is held High during a bus hold condition.

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

**$\overline{ONCE1}$** —During reset this pin and  $\overline{LCS}/\overline{ONCE0}$  indicate to the microcontroller the mode in which it should operate.  $\overline{ONCE0}$  and  $\overline{ONCE1}$  are sampled on the rising edge of  $\overline{RES}$ . 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,  $\overline{ONCE1}$  has a weak internal pullup resistor that is active only during a reset. This pin is not three-stated during a bus hold condition.

 **$\overline{UZI}/\text{PIO26}$** 
**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 on the 186 and AO15–AO10 on the 188).  $\overline{UZI}$  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, Am186ES Microcontroller Only  
(output, three-state, synchronous)**

This pin and  $\overline{WLB}$  indicate to the system which bytes of the data bus (upper, lower, or both) participate in a write cycle. In 80C186 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.

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

 **$\overline{WLB}/\overline{WB}$** 
**Write Low Byte, Am186ES Microcontroller  
(output, three-state, synchronous)**
**Write Byte, Am188ES Microcontroller  
(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.

**$\overline{WB}$** —On the Am188ES microcontroller, this pin indicates a write to the bus.  $\overline{WB}$  uses the same early timing as the nonmultiplexed address bus.  $\overline{WB}$  is associated with AD7–AD0. This pin floats during reset.

**$\overline{WR}$**

**Write Strobe (output, synchronous)**

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

**X1**

**Crystal Input (input)**

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

**X2**

**Crystal Output (output)**

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

### 3.1.1

#### Pins That Are Used by Emulators

The following pins are used by emulators: A19–A0, AO15–AO8, AD7–AD0, ALE,  $\overline{BHE}/\overline{ADEN}$  (on the 186), CLKOUTA,  $\overline{RFSH2}/\overline{ADEN}$  (on the 188),  $\overline{RD}$ ,  $\overline{S2}$ – $\overline{S0}$ , S6/ $\overline{LOCK}/\overline{CLKDIV2}$ , and  $\overline{UZI}$ .

Emulators require S6/ $\overline{LOCK}/\overline{CLKDIV2}$  and  $\overline{UZI}$  be configured in their normal functionality, that is as S6 and  $\overline{UZI}$ .

If  $\overline{BHE}/\overline{ADEN}$  (on the 186) or  $\overline{RFSH2}/\overline{ADEN}$  (on the 188) is held Low during the rising edge of RES, S6 and  $\overline{UZI}$  are configured in their normal functionality, instead of as PIOs, at reset.

## 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 Am186ES and Am188ES microcontrollers continue to provide the multiplexed AD bus and, in addition, provide a nonmultiplexed address (A) bus. The A bus provides an address to the system for the complete bus cycle ( $t_1$ – $t_4$ ).

For systems where power consumption is a concern, it is possible to disable the address from being driven on the AD bus on the Am186ES microcontroller and on the AD and AO buses on the Am188ES microcontroller during the normal address portion of the bus cycle for accesses to UCS and/or LCS address spaces. In this mode, the affected bus is placed in a high impedance state during the address portion of the bus cycle. This feature is enabled through the DA bits in the UMCS and LMCS registers. When address disable is in effect, the number of signals that assert on the bus during all normal bus cycles to the associated address space is reduced, thus decreasing power consumption, reducing processor switching noise, and preventing bus contention with memory devices and peripherals when operating at high clock rates. On the Am188ES microcontroller, the address is driven on A015–A08 during the data portion of the bus cycle, regardless of the setting of the DA bits.

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

Figure 3-1 shows the affected signals during a normal read or write operation for an Am186ES microcontroller. The address and data will be multiplexed onto the AD bus.

Figure 3-2 shows an Am186ES microcontroller bus cycle when address bus disable is in effect. This results in the AD bus operating in a nonmultiplexed data-only mode. The A bus will have the address during a read or write operation.

Figure 3-3 shows the affected signals during a normal read or write operation for an Am188ES microcontroller. The multiplexed address/data mode is compatible with the 80C188 microcontrollers and might be used to take advantage of existing logic or peripherals.

Figure 3-4 shows an Am188ES microcontroller bus cycle when address bus disable is in effect. The address and data are not multiplexed. The AD7–AD0 signals will have only data on the bus, while the AO bus will have the address during a read or write operation.

Figure 3-1 Am186ES Microcontroller Address Bus—Normal Read and Write Operation

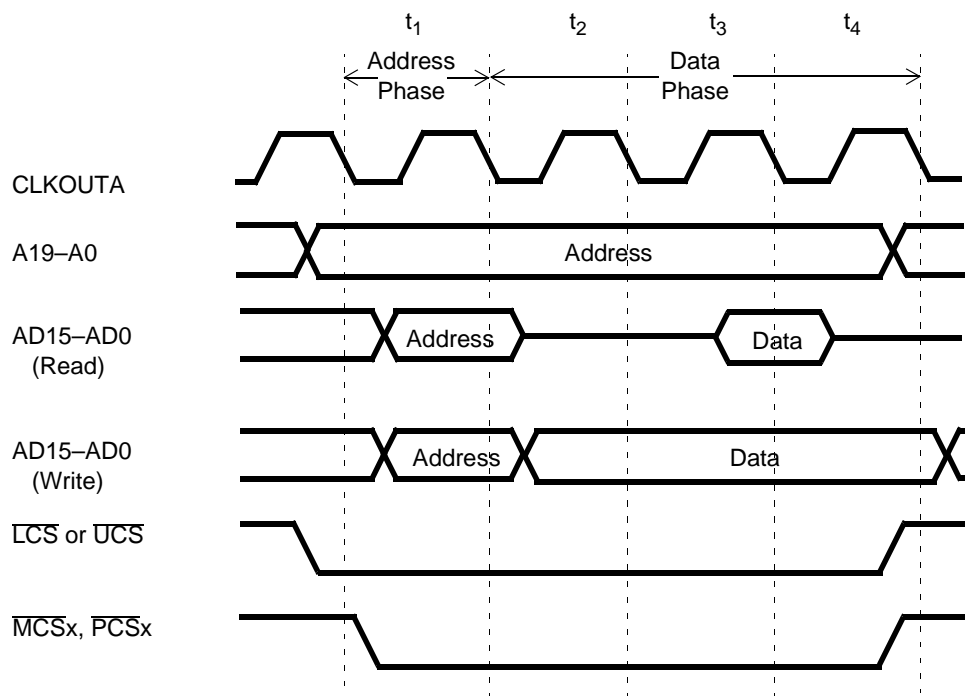


Figure 3-2 Am186ES Microcontroller—Read and Write with Address Bus Disable In Effect

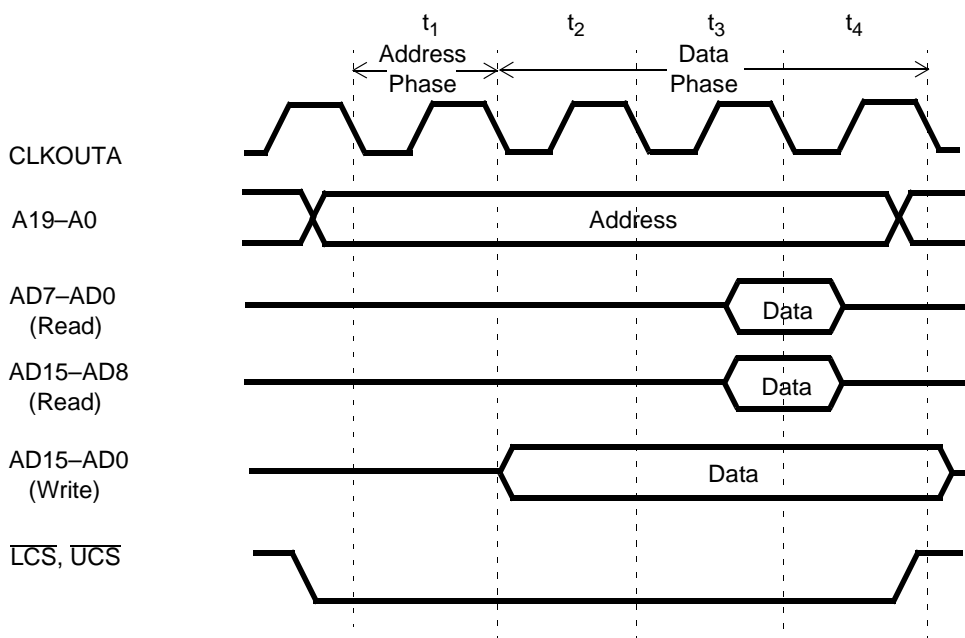


Figure 3-3 Am188ES Microcontroller Address Bus—Normal Read and Write Operation

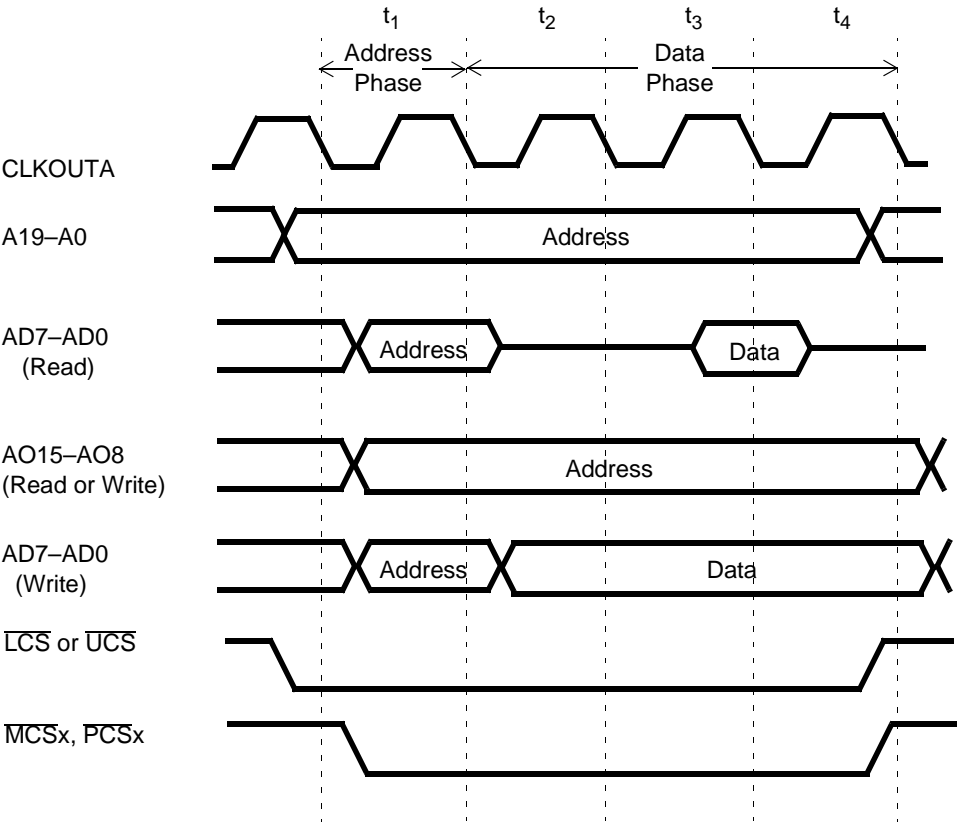
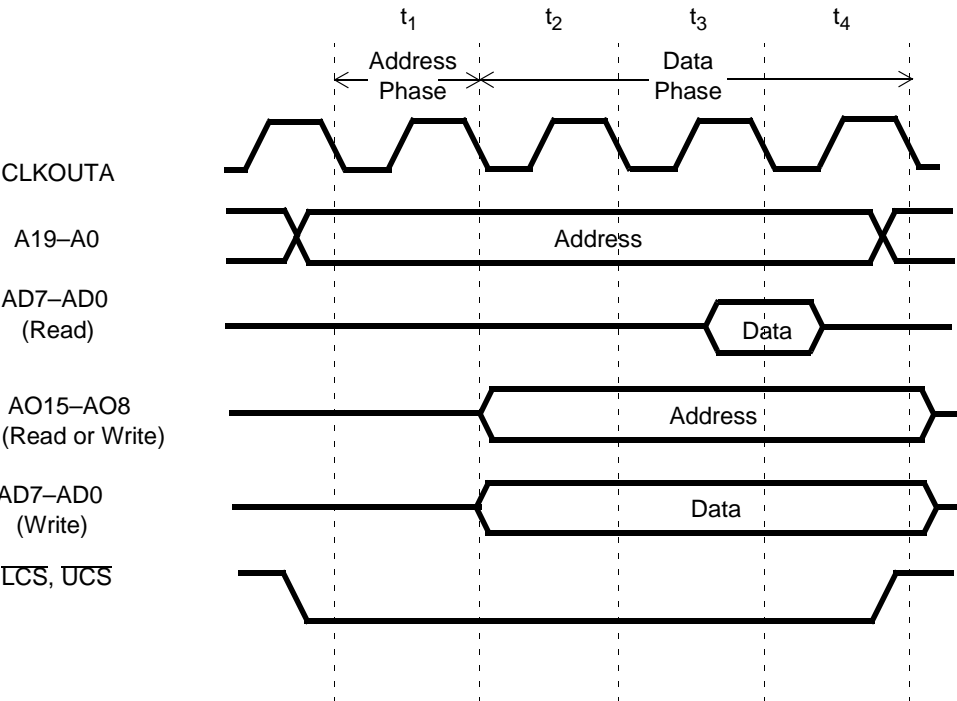


Figure 3-4 Am188ES Microcontroller—Read and Write with Address Bus Disable In Effect



### 3.3 BUS INTERFACE UNIT

The bus interface unit controls all accesses to external peripherals and memory devices. External accesses include those to memory devices, as well as those to memory-mapped and I/O-mapped peripherals and the peripheral control block. The Am186ES and Am188ES microcontrollers provide an enhanced bus interface unit with the following features:

- A nonmultiplexed address bus
- On the Am186ES microcontroller, a static bus sizing option for 8-bit and 16-bit memory and I/O
- On the Am186ES microcontroller, separate byte write enables for high and low bytes
- Pseudo Static RAM (PSRAM) support

The standard 80C186/188 microcontroller multiplexed address and data bus requires system-interface logic and an external address latch. On the Am186ES and Am188ES microcontrollers, new byte write enables, PSRAM control logic, and a new nonmultiplexed address bus can reduce design costs by eliminating external logic.

Timing diagrams for the operations described in this chapter appear in the *Am186ES and Am188ES Microcontrollers Data Sheet*, order# 20002.

#### 3.3.1 Nonmultiplexed Address Bus

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

#### 3.3.2 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. The 80C188 microcontroller provided a lower-cost interface by reducing the data bus width to 8 bits, again over the entire address range. The Am188ES microcontroller follows the 80C188 microcontroller in providing an 8-bit data bus to all memory and peripherals. However, the Am186ES microcontroller differs 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 Am186ES Microcontroller Bus Width

Space	AUXCON Field	Value	Bus Width	Comments
UCS	—	—	16 bits	not configurable
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	

### 3.3.3 Byte Write Enables

The Am186ES microcontroller provides 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 Am188ES microcontroller provides one signal for byte write enables— $\overline{WB}$  (Write Byte).  $\overline{WB}$  is the logical AND of  $\overline{WHB}$  and  $\overline{WLB}$ , which are not present on the Am188ES microcontroller.

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

### 3.3.4 Pseudo Static RAM (PSRAM) Support

The Am186ES and Am188ES microcontrollers support the use of PSRAM devices in low memory chip-select (LCS) space only. When PSRAM mode is enabled, the timing for the  $\overline{LCS}$  signal is modified by the chip select control unit to provide a  $\overline{CS}$  precharge period during PSRAM accesses. The 40-MHz timing of the Am186ES and Am188ES microcontrollers is appropriate to allow 70-ns PSRAM to run with one wait state. PSRAM mode is enabled through a bit in the Low Memory Chip Select (LMCS) register. (See section 5.5.2 on page 5-6.) The PSRAM feature is disabled on CPU reset.

In addition to the  $\overline{LCS}$  timing changes for PSRAM precharge, the PSRAM devices also require periodic refresh of all internal row addresses to retain their data. Although refresh of PSRAM can be accomplished several ways, the Am186ES and Am188ES microcontrollers implement auto refresh only. The microcontrollers generate a refresh signal,  $\overline{RFSH}$ , to the PSRAM devices when PSRAM mode and the refresh control unit are enabled. No refresh address is required by the PSRAM when using the auto refresh mechanism. The  $\overline{RFSH}$  signal is multiplexed with the  $\overline{MCS3}$  signal pin. When PSRAM mode is enabled,  $\overline{MCS3}$  is not available for use as a chip select signal.

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

The refresh counter reload value in the CDRAM register should not be set to less than 18 (12h) in order to provide time for processor cycles within refresh. The refresh address counter must be set to 0000h to prevent another chip select from asserting.  $\overline{LCS}$  is held



High during a refresh cycle. The A19–A0 bus is not used during refresh cycles. The LMCS register must be configured to external Ready ignored (R2=1) with one wait state (R1–R0=01b), and the PSRAM mode enable bit (PSE) must be set to 1. See section 5.5.2 on page 5-6.

### 3.4 CLOCK AND POWER MANAGEMENT UNIT

The clock and power management unit of the Am186ES and Am188ES 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 Am186ES and Am188ES microcontrollers, the internal clock generated by the microcontroller (CLKOUTA) is the same frequency as the crystal. The PLL takes the crystal inputs (X1 and X2) and generates a 45/55% (worst case) duty cycle intermediate system clock of the same frequency. This feature removes the need for an external 2x oscillator, thereby reducing system cost. The PLL is reset during power-on reset by an on-chip power-on reset (POR) circuit.

#### 3.4.2 Crystal-Driven Clock Source

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

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

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

### 3.4.2.1 Selecting a Crystal

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

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

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

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

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

Drive Level ..... 1 mW Max

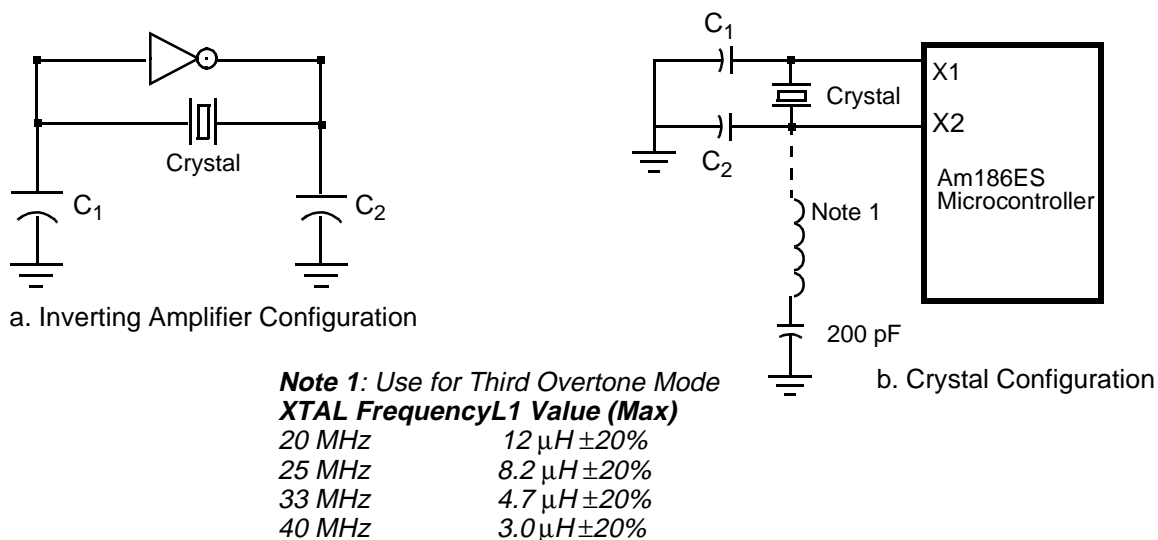
The recommended range of values for  $C_1$  and  $C_2$  are as follows:

$C_1$  ..... 15 pF  $\pm$  20%

$C_2$  ..... 22 pF  $\pm$  20%

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

Figure 3-5 Oscillator Configurations



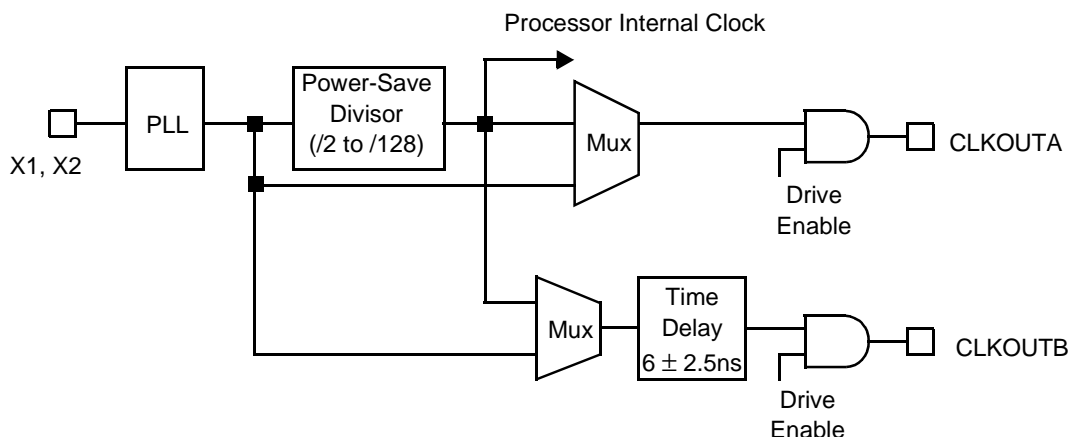
### 3.4.3 External Source Clock

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

### 3.4.4 System Clocks

Figure 3-6 shows the organization of the clocks. The original 80C186/188 microcontroller system clock has been renamed CLKOUTA. CLKOUTB is provided as an additional output.

Figure 3-6 Clock Organization



CLKOUTA and CLKOUTB operate at either the 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.



## 4.1

## OVERVIEW

The Am186ES and Am188ES microcontroller integrated peripherals are controlled by 16-bit read/write registers. The peripheral registers are contained within an internal 256-byte control block—the peripheral control block. Registers are physically located in the peripheral devices they control, but they are addressed as a single 256-byte block. Table 4-1 shows a map of the peripheral control block registers.

Code that is intended to execute on the Am188ES microcontroller should perform all writes to the PCB registers as byte writes. These writes will transfer 16 bits of data to the PCB register even if an 8-bit register is named in the instruction. For example, `out dx, al` results in the value of `ax` being written to the port address in `dx`. Reads to the PCB should be done as word reads. Code written in this manner will run correctly on the Am188ES microcontroller and on the Am186ES microcontroller. Unaligned reads and writes to the PCB result in unpredictable behavior on both the Am186ES and Am188ES microcontrollers.

The peripheral control block can be mapped into either memory or I/O space. The base address of the control block must be on an even 256-byte boundary (i.e., the lower eight bits of the base address are 00h). Internal logic recognizes control block addresses and responds to bus cycles. During bus cycles to internal registers, the bus controller signals the operation externally (i.e., the  $\overline{RD}$ ,  $\overline{WR}$ , 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	Register Name	Offset	Page
<b>Processor Control Registers: Chapters 4 and 6</b>			<b>Timer Registers: Chapter 8</b>		
Peripheral control block relocation register	FEh	4-3	Timer 2 mode/control register	66h	8-5
Reset configuration register	F6h	4-4	Timer 2 max count compare A register	62h	8-7
Processor release level register	F4h	4-5	Timer 2 count register	60h	8-6
Auxiliary configuration register	F2h	4-6	Timer 1 mode/control register	5Eh	8-3
System configuration register	F0h	4-7	Timer 1 max count compare B register	5Ch	8-7
Watchdog timer control register	E6h	6-3	Timer 1 max count compare A register	5Ah	8-7
Enable RCU register	E4h	6-2	Timer 1 count register	58h	8-6
Clock prescaler register	E2h	6-2	Timer 0 mode/control register	56h	8-3
Memory partition register	E0h	6-1	Timer 0 max count compare B register	54h	8-7
<b>DMA Registers: Chapter 9</b>			Timer 0 max count compare A register	52h	8-7
DMA 1 control register	DAh	9-3	Timer 0 count register	50h	8-6
DMA 1 transfer count register	D8h	9-6	<b>Interrupt Registers: Chapter 7</b>		
DMA 1 destination address high register	D6h	9-7	Serial port 0 interrupt control register	44h	7-18
DMA 1 destination address low register	D4h	9-8	Serial port 1 interrupt control register	42h	7-18
DMA 1 source address high register	D2h	9-9	INT4 interrupt control register	40h	7-16
DMA 1 source address low register	D0h	9-10	INT3 control register	3Eh	7-15
DMA 0 control register	CAh	9-3	INT2 control register	3Ch	7-15
DMA 0 transfer count register	C8h	9-6	INT1 control register	3Ah	7-14
DMA 0 destination address high register	C6h	9-7	INT0 control register	38h	7-14
DMA 0 destination address low register	C4h	9-8	DMA1/INT6 interrupt control register	36h	7-17
DMA 0 source address high register	C2h	9-9	DMA0/INT5 interrupt control register	34h	7-17
DMA 0 source address low register	C0h	9-10	Timer interrupt control register	32h	7-29
<b>Chip-Select Registers: Chapter 5</b>			Interrupt status register	30h	7-19
PCS and MCS auxiliary register	A8h	5-10	Interrupt request register	2Eh	7-20
Midrange memory chip-select register	A6h	5-8	Interrupt in-service register	2Ch	7-22
Peripheral chip-select register	A4h	5-12	Interrupt priority mask register	2Ah	7-23
Low memory chip-select register	A2h	5-6	Interrupt mask register	28h	7-34
Upper memory chip-select register	A0h	5-4	Interrupt poll status register	26h	7-25
<b>Serial Port 0 Registers: Chapter 10</b>			Interrupt poll register	24h	7-26
Serial port 0 baud rate divisor register	88h	10-13	End-of-interrupt register	22h	7-27
Serial port 0 receive register	86h	10-12	Interrupt vector register	20h	7-36
Serial port 0 transmit register	84h	10-11	<b>Serial Port 1 Registers: Chapter 10</b>		
Serial port 0 status register	82h	10-9	Serial port 1 baud rate divisor register	18h	10-13
Serial port 0 control register	80h	10-5	Serial port 1 receive register	16h	10-12
<b>PIO Registers: Chapter 11</b>			Serial port 1 transmit register	14h	10-11
PIO data 1 register	7Ah	11-5	Serial port 1 status register	12h	10-9
PIO direction 1 register	78h	11-4	Serial port 1 control register	10h	10-5
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			

**Note:** All unused addresses are reserved and should not be accessed.

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

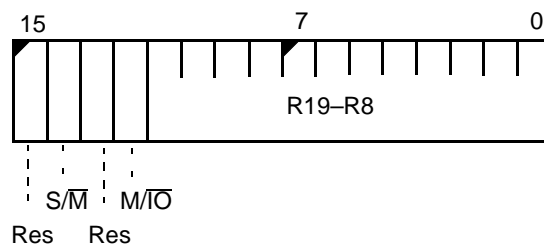
The peripheral control block is mapped into either memory or I/O space by programming the Peripheral Control Block Relocation (RELREG) register (see Figure 4-1). This register is a 16-bit register at offset FEh from the control block base address. The Peripheral Control Block Relocation register provides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

**Bit 14: Slave/Master (S/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/I $\bar{O}$ )**—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.

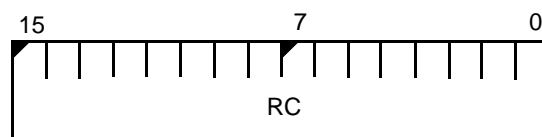
### 4.1.2 Reset Configuration Register (RESCON, Offset F6h)

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–AD0 for the 186 or AO15–AO8 and AD7–AD1 for the 188) 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  $\overline{RES}$  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 Am186ES microcontroller, AD15 corresponds to bit 15 of the Reset Configuration register, and so on. On the Am188ES microcontroller, AO15 corresponds to register bit 15 and AD7 corresponds to bit 7. Once  $\overline{RES}$  is deasserted, the Reset Configuration register holds its value. This value can be read by software to determine the configuration information.

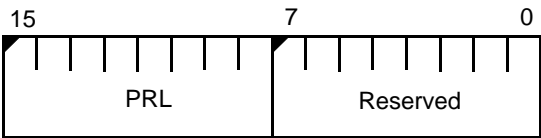
The contents of the Reset Configuration register are read-only and remain valid until the next processor reset.



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. The Am186ES and Am188ES microcontrollers’ revision A PRL is 10h.

**Bits 7–0: Reserved**

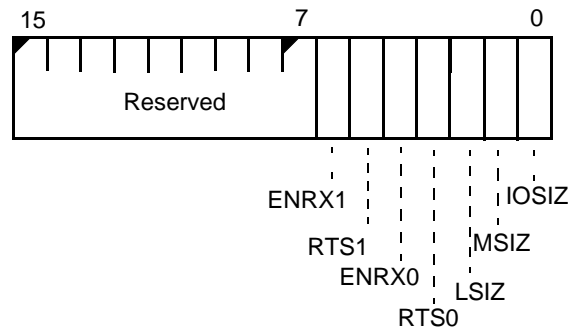
Table 4-2 Processor Release Level (PRL) Values

PRL Value	Processor Release Level
10h	A
11h	B
12h	C
13h	D
14h	E

#### 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 15–7: Reserved

**Bit 6: Serial Port 1 Enable Receiver Request (ENRX1)**—When this bit is 1, the  $\overline{\text{CTS1}}$ / $\overline{\text{ENRX1}}$  pin is configured as  $\overline{\text{ENRX1}}$ . When this bit is 0, the  $\overline{\text{CTS1}}$ / $\overline{\text{ENRX1}}$  pin is configured as  $\overline{\text{CTS1}}$ . This bit is 0 after processor reset.

**Bit 5: Serial Port 1 Request to Send (RTS1)**—When this bit is 1, the  $\overline{\text{RTR1}}$ / $\overline{\text{RTS1}}$  pin is configured as  $\overline{\text{RTS1}}$ . When this bit is 0, the  $\overline{\text{RTR1}}$ / $\overline{\text{RTS1}}$  pin is configured as  $\overline{\text{RTR1}}$ . This bit is 0 after processor reset.

**Bit 4: Serial Port 0 Enable Receiver Request (ENRX0)**—When this bit is 1, the  $\overline{\text{CTS0}}$ / $\overline{\text{ENRX0}}$  pin is configured as  $\overline{\text{ENRX0}}$ . When this bit is 0, the  $\overline{\text{CTS0}}$ / $\overline{\text{ENRX0}}$  pin is configured as  $\overline{\text{CTS0}}$ . This bit is 0 after processor reset.

**Bit 3: Serial Port 0 Request to Send (RTS0)**—When this bit is 1, the  $\overline{\text{RTR0}}$ / $\overline{\text{RTS0}}$  pin is configured as  $\overline{\text{RTS0}}$ . When this bit is 0, the  $\overline{\text{RTR0}}$ / $\overline{\text{RTS0}}$  pin is configured as  $\overline{\text{RTR0}}$ . This bit is 0 after processor reset.

**Bit 2: LCS Data Bus Size (LSIZ)**—(Am186ES microcontroller only) This bit determines the width of the data bus for accesses to 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 LCS space or while the PCB is overlaid with LCS space. This bit is 0 after processor reset.

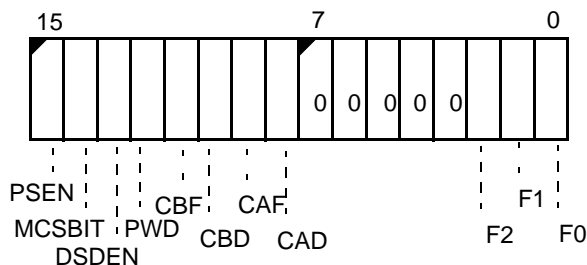
**Bit 1: Midrange Data Bus Size (MSIZ)**—(Am186ES microcontroller only) This bit determines the width of the data bus for memory accesses which do not fall into the UCS or LCS address spaces, including MCS address space and PCS address space, if mapped to memory. If this bit is 1, 8-bit accesses are 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)**—(Am186ES microcontroller only) 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 PCS 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, including those generated by on-chip peripheral devices, occurs. The value of the PSEN bit is not restored by the execution of an IRET instruction. Software interrupts (INT instruction) and exceptions do not clear the PSEN bit, and interrupt service routines for these conditions should do so, if desired. This bit is 0 after processor reset.

**Bit 14: MCS0 Only Mode Bit (MCSBIT)**—This bit controls the MCS0 only mode. When set to 0, the middle chip selects operate normally. When set to 1,  $\overline{\text{MCS0}}$  is active over the entire 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 Am186ES and Am188ES 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	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
INT2 Control	I2CON	000Fh	INT2 interrupt masked, edge-triggered, priority 7

Register Name	Mnemonic	Value at Reset	Comments
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	FFF9h	
DMA 0 Control	D0CON	FFF9h	
Watchdog Timer Control	WDTCON	C080h	

*Note:*

*Registers not listed in this table are undefined at reset.*

## 5.1

## OVERVIEW

The Am186ES and Am188ES 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 Am186ES and Am188ES microcontrollers provide six chip select outputs for use with memory devices and six more for use with peripherals in either memory space or I/O space. The six memory chip selects can be used to address three memory ranges. Each peripheral chip select addresses a 256-byte block offset from a programmable base address (see Section 4.1.1 on page 4-3).

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 ( $\overline{MCS3}$ – $\overline{MCS0}$ ) 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 ( $\overline{PCS6}$ – $\overline{PCS5}$  and  $\overline{PCS3}$ – $\overline{PCS0}$ ).

**Note:** The SYSCON register contains the MCSBIT field which determines behavior of the  $\overline{MCS0}$  chip select. The  $\overline{PCS4}$  chip select is not implemented on the Am186ES and Am188ES 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	$\overline{PCS6}$ – $\overline{PCS5}$ $\overline{PCS3}$ – $\overline{PCS0}$	Block size is fixed at 256 bytes
A6h	MMCS	Midrange Chip Select	$\overline{MCS3}$ – $\overline{MCS0}$	Starting address and block size are programmable
A8h	MPCS	$\overline{PCS}$ and $\overline{MCS}$ Auxiliary	$\overline{PCS6}$ – $\overline{PCS5}$ $\overline{PCS3}$ – $\overline{PCS0}$ $\overline{MCS3}$ – $\overline{MCS0}$	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 accessed. (An access is any write operation and only write operations activate.) For this reason, the chip select registers should not be read by the processor initialization code until after they have been written with valid data. The  $\overline{LCS}$  chip select is activated when the LMCS register is accessed, the  $\overline{MCS}$  chip selects are activated after both the MMCS and MPCS registers have been accessed, and the  $\overline{PCS}$  chip selects are activated after both the PACS and MPCS registers have been accessed.

## 5.2 CHIP SELECT TIMING

The timing for the  $\overline{UCS}$  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  $\overline{MCS3}$ – $\overline{MCS0}$  and  $\overline{PCS}$  chip selects assert with the AD bus.

## 5.3 READY AND WAIT-STATE PROGRAMMING

The Am186ES and Am188ES 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  $\overline{PCS3}$ – $\overline{PCS0}$  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  $\overline{PCS3}$ – $\overline{PCS0}$  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 Am186ES and Am188ES 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.

## 5.4 CHIP SELECT OVERLAP

Although programming the various chip selects on the Am186ES microcontroller so that multiple chip select signals are asserted for the same physical address is not recommended, it may be unavoidable in some systems. In such systems, the chip selects whose assertions



overlap must have the same configuration for ready (external ready required or not required) and the number of wait states to be inserted into the cycle by the processor.

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

When overlapping an additional chip select with either the  $\overline{\text{LCS}}$  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  $\overline{\text{PCS4}}$  signal is not available on an external pin, the ready and wait state logic for this signal still exists internal to the part. For this reason, the  $\overline{\text{PCS4}}$  address space must follow the rules for overlapping chip selects. The ready and wait-state logic for  $\overline{\text{PCS6}}$ – $\overline{\text{PCS5}}$  is disabled when these signals are configured as address bits A2–A1.

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

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

## 5.5

### CHIP SELECT REGISTERS

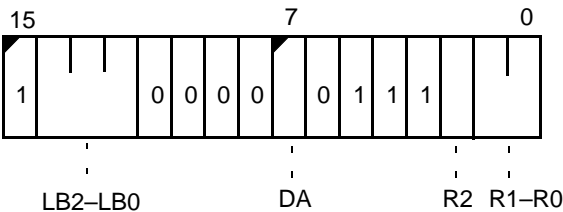
The following sections describe the chip select registers.

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

The Am186ES and Am188ES 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  $\overline{UCS}$  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 access through the  $\overline{UCS}$  chip selects. The number of programmable bits has been reduced from the eight bits in the 80C186 and 80C188 microcontrollers to three bits in the Am186ES and Am188ES microcontrollers.

The Am186ES and Am188ES microcontrollers provide an additional block size of 512K, which is not available on the 80C186 and 80C188 microcontrollers. Table 5-2 outlines the possible configurations and differences with the 80C186 and 80C188 microcontrollers.

Table 5-2 UMCS Block Size Programming Values

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

**Bits 11–8: Reserved**

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

**Note:** On the Am188ES microcontroller, the AO15–AO8 address pins are driven during the data phase of the bus cycles, even when DA is set to 1 in either the Upper Memory Chip Select register (UMCS) or the Low Memory Chip Select register (LMCS).

If  $\overline{BHE}/\overline{ADEN}$  (on the 186) or  $\overline{RFSH2}/\overline{ADEN}$  (on the 188) is held Low on the rising edge of  $\overline{RES}$ , 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  $\overline{BHE}/\overline{ADEN}$  (on the 186) or  $\overline{RFSH2}/\overline{ADEN}$  (on the 188) is High on the rising edge of  $\overline{RES}$ , then DA in the Upper Memory Chip Select (UMCS) register and DA in the Low Memory Chip Select (LMCS) register control the AD15–AD0 disabling.

See the descriptions of the  $\overline{BHE}/\overline{ADEN}$  and  $\overline{RFSH2}/\overline{ADEN}$  pins in Chapter 3.

**Bit 6: Reserved**—Set to 0.

**Bits 5–3: Reserved**—Set to 1.

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

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

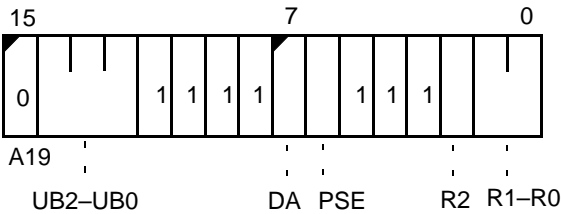
5.5.2 Low Memory Chip Select Register (LMCS, Offset A2h)

The Am186ES and Am188ES microcontrollers provide the  $\overline{\text{LCS}}$  chip select pin for the bottom of memory. Since the interrupt vector table is located at 00000h at the bottom of memory, the  $\overline{\text{LCS}}$  pin has been provided to facilitate this usage. The  $\overline{\text{LCS}}$  pin is not active on reset, but any write access 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 Low Memory Chip Select is configured through the LMCS register (see Figure 5-2).

Figure 5-2 Low Memory Chip Select Register



The value of the LMCS register at reset is undefined except DA is set to 0.

**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  $\overline{\text{LCS}}$  chip select. Because of the timing requirements of the  $\overline{\text{LCS}}$  output and the nonmultiplexed address bus, the number of programmable memory sizes for the LMCS register is reduced compared to the 80C186 and 80C188 microcontrollers. Consequently, the number of programmable bits has been reduced from eight bits in the 80C186 and 80C188 microcontrollers to three bits in the Am186ES and Am188ES microcontrollers.

The Am186ES and Am188ES microcontrollers have a block size of 512 Kbytes, which is not available on the 80C186 and 80C188 microcontrollers. Table 5-3 outlines the possible configurations.

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  $\overline{\text{LCS}}$  is asserted. 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.

**Note:** On the Am188ES microcontroller, the AO15–AO8 address pins are driven during the data phase of the bus cycles, even when DA is set to 1 in either the Upper Memory Chip Select register (UMCS) or the Low Memory Chip Select register (LMCS).

If  $\overline{\text{BHE}}/\overline{\text{ADEN}}$  (on the 186) or  $\overline{\text{RFSH2}}/\overline{\text{ADEN}}$  (on the 188) is held Low on the rising edge of  $\overline{\text{RES}}$ , then AD15–AD0 is always driven regardless of the DA setting.

If  $\overline{\text{BHE}}/\overline{\text{ADEN}}$  (on the 186) or  $\overline{\text{RFSH2}}/\overline{\text{ADEN}}$  (on the 188) 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 descriptions of the  $\overline{\text{BHE}}/\overline{\text{ADEN}}$  and  $\overline{\text{RFSH2}}/\overline{\text{ADEN}}$  pins in Chapter 3.

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

$\overline{\text{MCS3}}/\overline{\text{RFSH}}$  is configured as  $\overline{\text{RFSH}}$  by setting the enable bit (EN) in the enable RCU register (EDRAM, offset E4h).

**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.

**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)

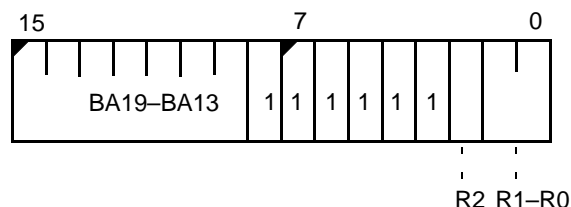
The Am186ES and Am188ES microcontrollers provide four chip select pins,  $\overline{MCS3}$ – $\overline{MCS0}$ , for use within a user-locatable memory block. The base address of the memory block can be located anywhere within the 1-Mbyte memory address space, exclusive of the areas associated with the  $\overline{UCS}$  and  $\overline{LCS}$  chip selects (and, if they are mapped to memory, the address range of the Peripheral Chip Selects,  $\overline{PCS6}$ – $\overline{PCS5}$  and  $\overline{PCS3}$ – $\overline{PCS0}$ ). The  $\overline{MCS}$  address range can overlap the  $\overline{PCS}$  address range if the  $\overline{PCS}$  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  $\overline{MCS}$  pins. The  $\overline{PCS}$  and  $\overline{MCS}$  Auxiliary (MPCS) register is used to configure the block size. The  $\overline{MCS3}$ – $\overline{MCS0}$  pins are not active on reset. Both the MMCS and MPCS registers must be accessed with a read to activate these chip selects.

Unlike the  $\overline{UCS}$  and  $\overline{LCS}$  chip selects, the  $\overline{MCS3}$ – $\overline{MCS0}$  outputs assert with the multiplexed AD address bus (AD15–AD0 or AO15–AO8 and AD7–AD0) rather than the earlier timing of the A19–A0 bus. The A19–A0 bus can still be used for address selection, but the timing is delayed for a half cycle later than that for  $\overline{UCS}$  and  $\overline{LCS}$ .

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  $\overline{MCS}$  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  $\overline{LCS}$  chip select is not active. This is because the  $\overline{LCS}$  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  $\overline{LCS}$  chip selects must not be active in this case. Use of the  $\overline{MCS}$  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  $\overline{UCS}$  chip select and is not allowed.

**Bits 8–3: Reserved**—Set to 1.

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

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

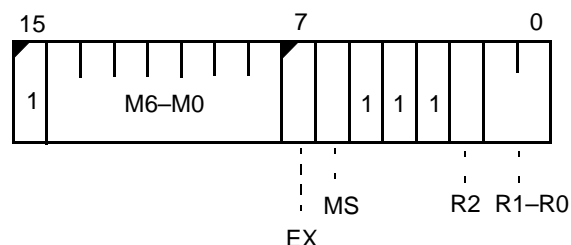
#### 5.5.4 $\overline{\text{PCS}}$ and $\overline{\text{MCS}}$ Auxiliary Register (MPCS, Offset A8h)

The  $\overline{\text{PCS}}$  and  $\overline{\text{MCS}}$  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  $\overline{\text{MCS}}3\text{--}\overline{\text{MCS}}0$  as well as  $\overline{\text{PCS}}6\text{--}\overline{\text{PCS}}5$  and  $\overline{\text{PCS}}3\text{--}\overline{\text{PCS}}0$ .

In addition to its function as a chip select control register, the MPCS register contains a field that configures the  $\overline{\text{PCS}}6\text{--}\overline{\text{PCS}}5$  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,  $\overline{\text{PCS}}6\text{--}\overline{\text{PCS}}5$  cannot be used as peripheral chip selects. These outputs can be used to provide latched address bits for A2 and A1.

On reset,  $\overline{\text{PCS}}6\text{--}\overline{\text{PCS}}5$  are not active. If  $\overline{\text{PCS}}6\text{--}\overline{\text{PCS}}5$  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  $\overline{\text{PCS}}6\text{--}\overline{\text{PCS}}5$  pins as addresses.

Figure 5-4  $\overline{\text{PCS}}$  and  $\overline{\text{MCS}}$  Auxiliary Register



The value of the MPCS register at reset is undefined.

**Bit 15: Reserved**—Set to 1.

**Bits 14–8:  $\overline{\text{MCS}}$  Block Size (M6–M0)**—This field determines the total block size for the  $\overline{\text{MCS}}3\text{--}\overline{\text{MCS}}0$  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  $\overline{\text{MCS}}$  lines occurs.

If the MCSBIT in the SYSCON register is set,  $\overline{\text{MCS}}0$  asserts over the entire programmed block size.  $\overline{\text{MCS}}3\text{--}\overline{\text{MCS}}1$  will continue to assert over their programmed range but are typically used as PIOs in the 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 and 80C188 microcontrollers.

The Am186ES and Am188ES microcontrollers provide six chip selects,  $\overline{PCS6}$ – $\overline{PCS5}$  and  $\overline{PCS3}$ – $\overline{PCS0}$ , for use within a user-locatable memory or I/O block. ( $\overline{PCS4}$  is not implemented on the Am186ES and Am 188ES microcontrollers). The base address of the memory block can be located anywhere within the 1-Mbyte memory address space, exclusive of the areas associated with the  $\overline{UCS}$ ,  $\overline{LCS}$ , and  $\overline{MCS}$  chip selects, or they can be configured to access the 64-Kbyte I/O space.

The Peripheral Chip Selects are programmed through two registers—the Peripheral Chip Select (PACS) register and the  $\overline{PCS}$  and  $\overline{MCS}$  Auxiliary (MPCS) register. The Peripheral Chip Select (PACS) register (Figure 5-5) determines the base address, the ready condition, and the wait states for the  $\overline{PCS3}$ – $\overline{PCS0}$  outputs.

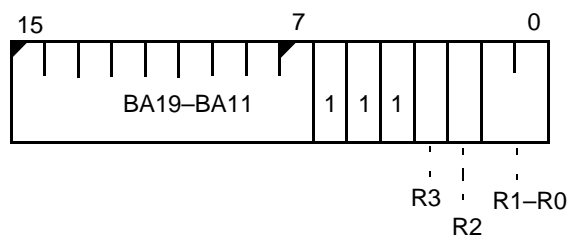
The  $\overline{PCS}$  and  $\overline{MCS}$  Auxiliary (MPCS) register (see Figure 5-4) contains bits that configure the  $\overline{PCS6}$ – $\overline{PCS5}$  pins as either chip selects or address pins A1 and A2. When the  $\overline{PCS6}$ – $\overline{PCS5}$  pins are chip selects, the MPCS register 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  $\overline{PCS6}$ – $\overline{PCS5}$  outputs.

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

$\overline{PCS6}$ – $\overline{PCS5}$  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.

$\overline{PCS3}$ – $\overline{PCS0}$  can be configured for zero wait states to 15 wait states.  $\overline{PCS6}$ – $\overline{PCS5}$  can be configured for zero wait states to three wait states.

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 and 80C188 microcontrollers and is not implemented. Thus, code previously written for the 80C186 microcontroller in which bit 6 was set with a meaningful value would not produce the address expected on the Am186ES microcontroller.

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

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

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

**Bits 6–4: Reserved**—Set to 1.

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

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

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

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

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

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

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

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



## 6.1 OVERVIEW

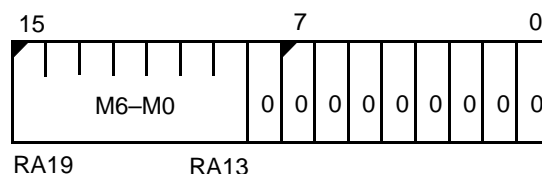
The Refresh Control Unit (RCU) automatically generates refresh bus cycles. After a programmable period of time, the RCU generates a memory read request to the bus interface unit. The RCU is fixed to three wait states for the PSRAM auto refresh mode.

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

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

### 6.1.1 Memory Partition Register (MDRAM, Offset E0h)

Figure 6-1 Memory Partition Register



The MDRAM register is set to 0000h on reset.

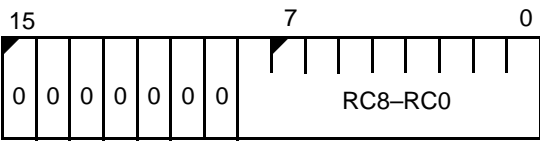
**Bits 15–9: Refresh Base (M6–M0)**—Upper bits corresponding to address bits A19–A13 of the 20-bit memory refresh address. These bits are available only on the AD bus, they are not available on the A19–A0 bus. When using PSRAM mode, M6–M0 must be programmed to 0000000b.

These bits are cleared to 0 at reset.

**Bits 8–0: Reserved**—Read back as 0.

6.1.2 Clock Prescaler Register  
(CDRAM, Offset E2h)

Figure 6-2 Clock Prescaler Register



The CDRAM register is undefined on reset.

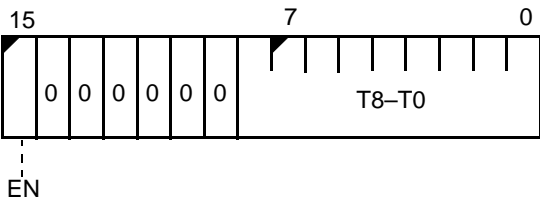
**Bits 15–9: Reserved**—Read back as 0.

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

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

6.1.3 Enable RCU Register  
(EDRAM, Offset E4h)

Figure 6-3 Enable RCU Register



The EDRAM register is set to 0000h on reset.

**Bit 15: Enable RCU (EN)**—Enables the refresh counter unit and changes  $\overline{MCS3}$  to  $\overline{RFSH}$  when set to 1. Clearing the EN bit at any time clears the refresh counter and stops refresh requests, but it does not reset the refresh address. This bit is 0 after processor reset.

**Bits 14–9: Reserved**—Read back as 0.

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

### 6.1.4 Watchdog Timer Control Register (WDTCON, Offset E6h)

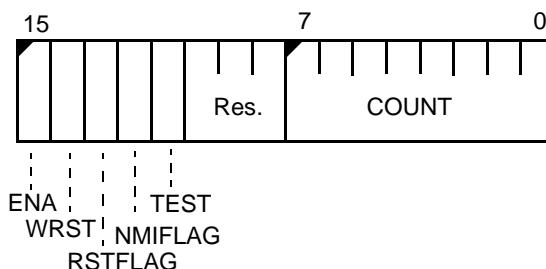
The Watchdog Timer Control register is a combined status and control register through which all watchdog timer functionality is implemented. The format of the watchdog timer control register is shown in Figure 6-4.

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 determined by using the value from the column titled Exponent in Table 6-1 in the following equation.

$$\text{Duration} = 2^{\text{Exponent}} / \text{Frequency}$$

Where Duration is the timeout period in seconds, Exponent is the value from Table 6-2, 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.

$$\begin{aligned} \text{Duration} &= (2^{24} \text{ cycles}) / (40,000,000 \text{ Hz}) \\ &= (16,777,216 \text{ cycles}) / (40,000,000 \text{ cycles/second}) \\ &= .4194 \text{ seconds} \end{aligned}$$

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

Table 6-1 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 6-2 Watchdog Timer Duration

Exponent	20 MHz	25 MHz	33 MHz	40 MHz
10	51 $\mu$ s	40 $\mu$ s	30 $\mu$ s	25 $\mu$ s
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



## 7.1 OVERVIEW

The Am186ES and Am188ES 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 Am186ES and Am188ES 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 Am186ES and Am188ES microcontrollers provide up to six interrupt sources that are not present on the 80C186 and 80C188 microcontrollers:

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

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 Am186ES and Am188ES 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 Am186ES and Am188ES 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
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 Am186ES and Am188ES 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 7-3 on page 7-18).

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 should 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 Am186ES and Am188ES microcontrollers do not support the numeric coprocessor interface.

#### 7.1.5 Interrupt Acknowledge

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

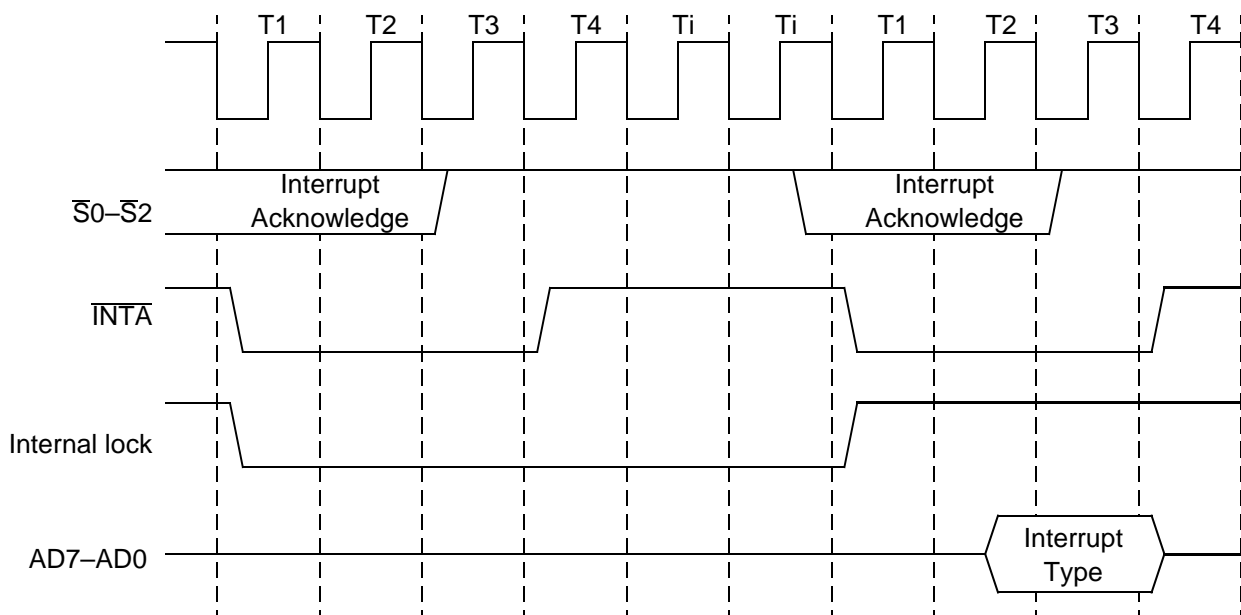
When the internal interrupt controller is supplying the interrupt type, no interrupt acknowledge bus cycles are generated. The only external indication that an interrupt is being serviced is the processor reading the interrupt vector table.

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.

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 7-28 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

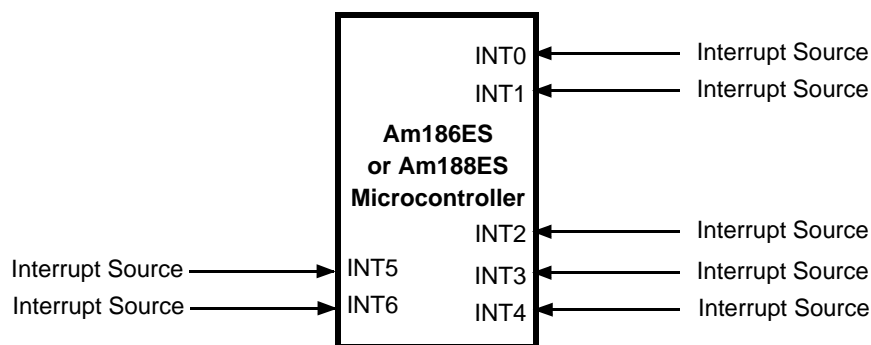
The basic modes of operation of the interrupt controller in master mode are similar to the 82C59A. The interrupt controller responds identically to internal interrupts in all three modes, the difference is only in the interpretation of function of the five external interrupt pins. The interrupt controller 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

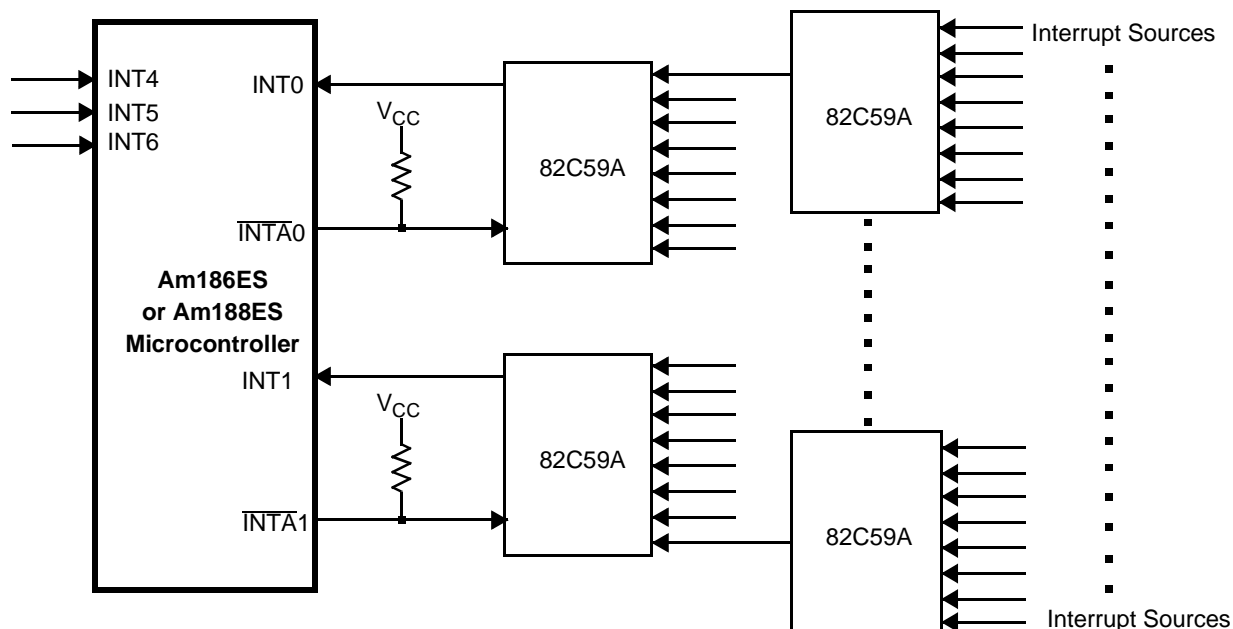
The Am186ES and Am188ES microcontrollers have seven interrupt pins, two of which (INT2 and INT3) have dual functions. In fully nested mode, the seven pins are used as direct interrupt inputs and the corresponding interrupt types are generated internally. In cascade mode, four of the seven pins can be configured into interrupt input and dedicated acknowledge signal pairs. INT0 can be configured with interrupt acknowledge  $\overline{\text{INTA0}}$  (INT2). INT1 can be configured with interrupt acknowledge  $\overline{\text{INTA1}}$  (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/ $\overline{\text{INTA0}}$  serves as the dedicated interrupt acknowledge signal to that peripheral. INT1 and INT3/ $\overline{\text{INTA1}}$  are also interfaced to an 82C59A. Each interrupt and acknowledge pair can be selectively placed in the cascade or non-cascade mode by programming the proper value into the INT0 and INT1 control registers. The dedicated acknowledge signals eliminate the need for external logic to generate  $\overline{\text{INTA}}$  and device select signals.

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

Figure 7-3 Cascade Mode Interrupt Controller Connections



### 7.2.3 Special Fully Nested Mode

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 Am186ES or Am188ES microcontroller interrupt request pin. As a result, if the external interrupt controller receives a higher-priority interrupt, its interrupt is not recognized by the microcontroller until the in-service bit is reset.

In special fully nested mode, the microcontroller 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 Am186ES or Am188ES microcontroller interrupt sources.

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

### 7.2.4 Operation in a Polled Environment

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 Am186ES and Am188ES 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 7-27).

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

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

## 7.3

## MASTER MODE INTERRUPT CONTROLLER REGISTERS

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

Registers can be redefined in slave mode. See Section 7.4 on page 7-28 for detailed information regarding slave mode register usage. On reset, the microcontroller is in master mode. Bit 14 of the 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

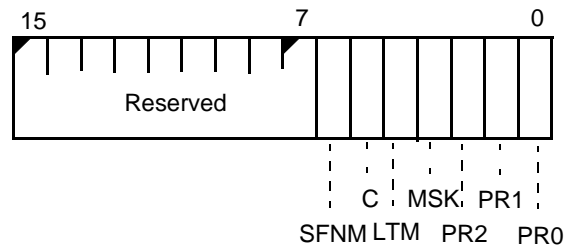
7.3.1 INT0 and INT1 Control Registers  
(IOCON, Offset 38h, I1CON, Offset 3Ah)  
(Master Mode)

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 IOCON 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 IOCON and I1CON at reset is 000Fh.

**Bits 15–7: Reserved**—Set to 0.

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

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

**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 7-24.

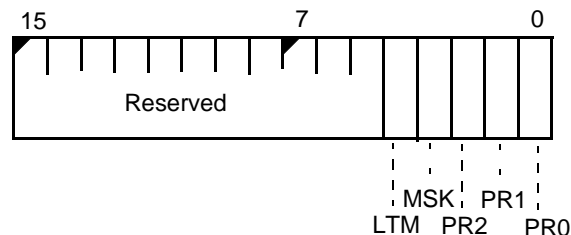
**Bits 2–0: Priority Level (PR2–PR0)**—This field determines the priority of INT0 or INT1 relative to the other interrupt signals, as shown in Table 7-3 on page 7-18.

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

The INT2 interrupt is assigned to interrupt type 0Eh. 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 7-24.

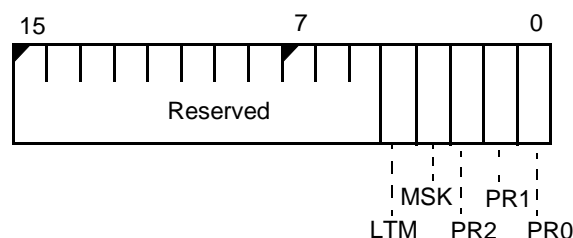
**Bits 2–0: Priority Level (PR2–PR0)**—This field determines the priority of INT2 or INT3 relative to the other interrupt signals, as shown in Table 7-3 on page 7-18.

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

The Am186ES and Am188ES 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 7-24.

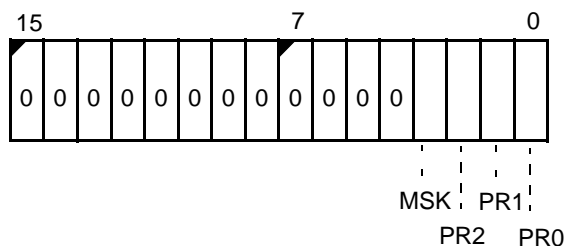
**Bits 2–0: Priority (PR)**—This field determines the priority of INT4 relative to the other interrupt signals, as shown in Table 7-3 on page 7-18.



#### 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.

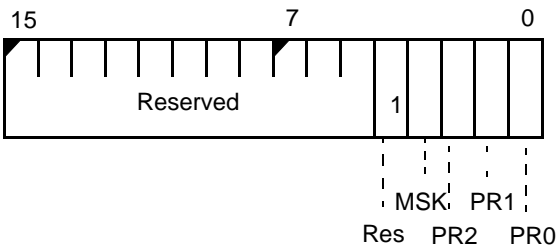
This bit is duplicated in the Interrupt Mask register. See the Interrupt Mask register in Section 7.3.10 on page 7-24.

**Bits 2–0: Priority Level (PR2–PR0)**—Sets the priority level for its corresponding source. See Table 7-3 on page 7-18.

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 SP1CON 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 7-24.

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

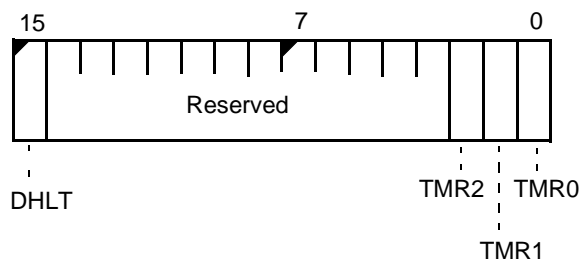
Table 7-3 Priority Level

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

### 7.3.6 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)

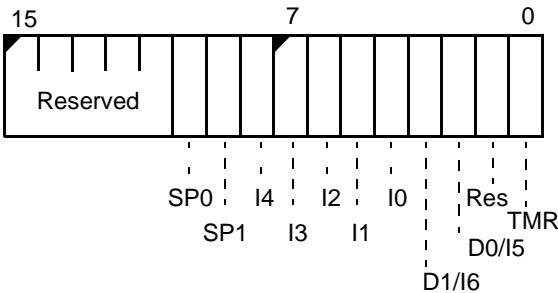
The hardware interrupt sources have interrupt request bits inside the interrupt controller. A read from this register yields the status of these bits. The Interrupt Request register is a read-only register. The format of the Interrupt Request register is shown in Figure 7-10.

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**

---

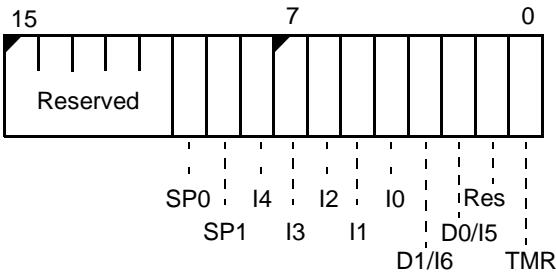
**Bit 0: Timer Interrupt Request (TMR)**—This bit indicates the state of the timer interrupts. This bit is the logical OR of the timer interrupt requests. When set to a 1, this bit indicates that the timer control unit has an interrupt pending.

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  
(INSERTV, 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 INSERTV 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.

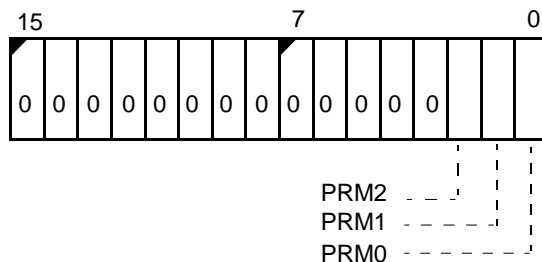
**Bit 1: Reserved**

**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 7-4 Priority Level

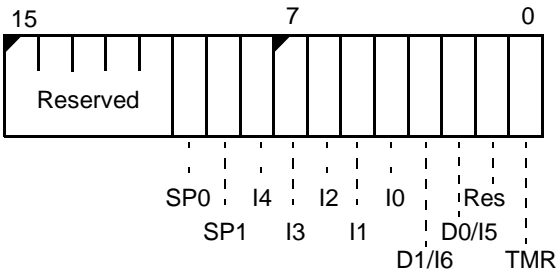
Priority	PR2–PR0
(High) 0	0 0 0b
1	0 0 1b
2	0 1 0b
3	0 1 1b
4	1 0 0b
5	1 0 1b
6	1 1 0b
(Low) 7	1 1 1b

7.3.10 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**

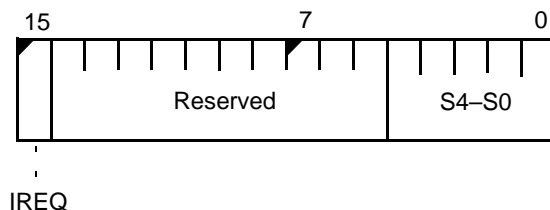
**Bit 0: Timer Interrupt Mask (TMR)**—When set to 1, this bit indicates that interrupt requests from the timer control unit are masked.



### 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.

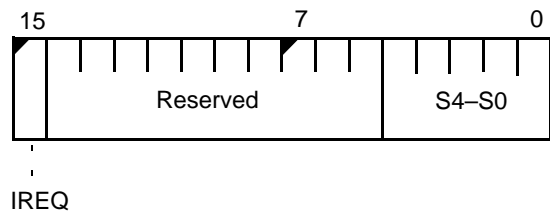
**Bits 4–0: Poll Status (S4–S0)**—Indicates the interrupt type of the highest priority pending interrupt (see Table 7-1 on page 7-4).

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 7-22) are reset by writing to the EOI register. Before executing the IRET instruction that ends an interrupt service routine (ISR), the ISR should write to the EOI register to reset the IS bit for the interrupt.

The specific EOI reset is the most secure method to use for resetting IS bits. Figure 7-16 shows example code for a specific EOI reset. See Table 7-1 on page 7-4 for specific EOI values.

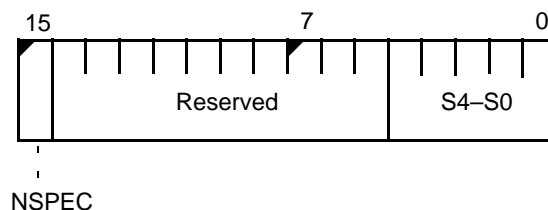
Figure 7-16 Example EOI Assembly Code

```

...
...                ;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 7-5. All registers can be read and written, unless specified otherwise.

Table 7-5 Interrupt Controller Registers in Slave Mode

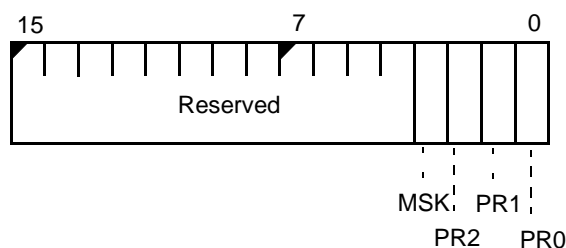
Offset	Register Mnemonic	Register Name	Affected Pins	Comments
3Ah	T2INTCON	Timer 2 Interrupt Control		Interrupt Type XXXXX101
38h	T1INTCON	Timer 1 Interrupt Control	TMRIN1 TMROUT1	Interrupt Type XXXXX100
36h	DMA1CON	DMA 1 Interrupt Control/INT6	DRQ1/INT6	Interrupt Type XXXXX011
34h	DMA0CON	DMA 0 Interrupt 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.

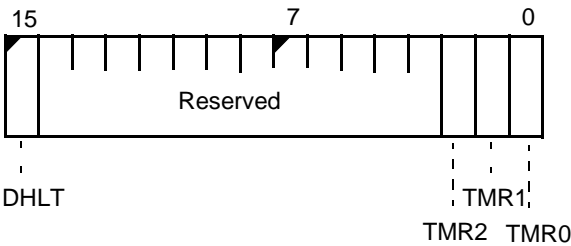
This bit is duplicated in the Interrupt Mask register. See the Interrupt Mask register in Section 7.4.8 on page 7-34.

**Bits 2–0: Priority Level (PR2–PR0)**—This field determines the priority of the interrupt source relative to the other interrupt signals, as shown in Table 7-3 on page 7-18.

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**

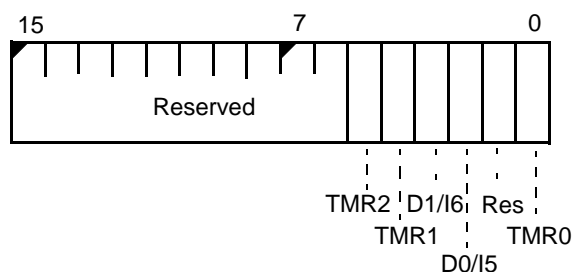
**Bits 2–0: Timer Interrupt Request (TMR2–TMR0)**—When set to 1, indicates the corresponding timer has an interrupt request pending.

### 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.

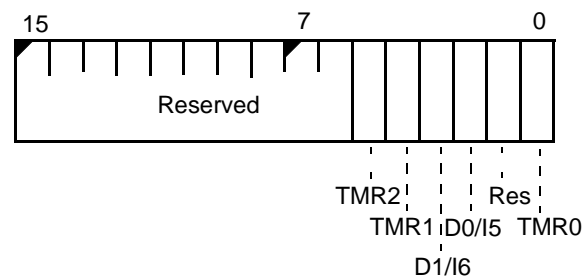
#### Bit 1: Reserved

**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  
(INSERTV, 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 INSERTV 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**

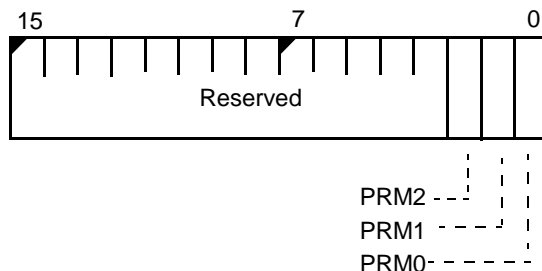
**Bit 0: Timer 0 Interrupt In-Service (TMR0)**—When set to 1, this bit indicates timer 0 is currently being serviced.



### 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.

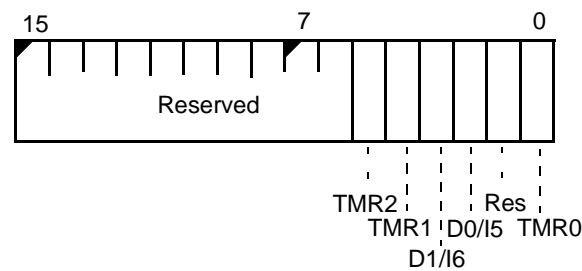
Table 7-6 Priority Level

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

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

The format of the Interrupt Mask register is shown in Figure 7-23. 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.

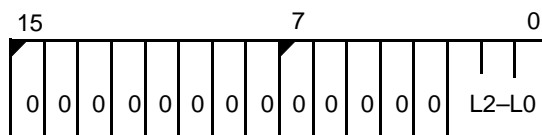
**Bit 1: Reserved**

**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.

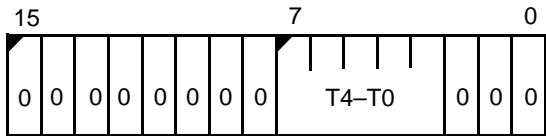
**Bits 15–3: Reserved**—Write as 0.

**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 7-5 on page 7-28.

**Bits 2–0: Reserved**—Read as 0.

### 8.1 OVERVIEW

There are three 16-bit programmable timers in the Am186ES and Am188ES 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.

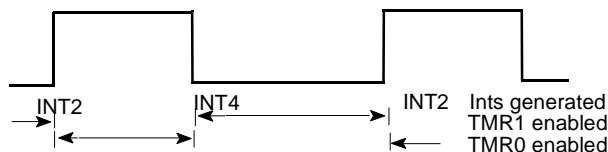
### 8.2 PULSE WIDTH DEMODULATION

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 Am186ES and Am188ES 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 Am186ES and Am188ES 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/INTA0/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.

### 8.3.1 Timer Operating Frequency

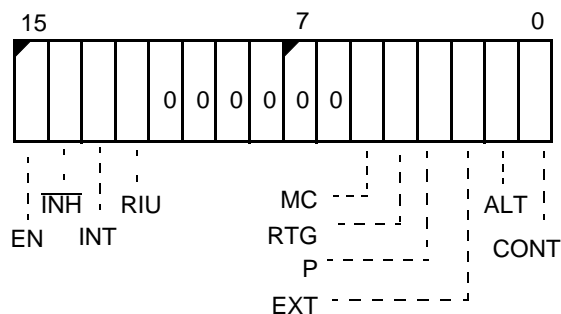
Each timer is serviced on every fourth clock cycle. Therefore, a timer can operate at a maximum speed of one-quarter of the internal clock frequency. A timer can be clocked externally at the same maximum frequency of one-fourth of the internal clock frequency. However, because of internal synchronization 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: Prescaler Bit (P)**—When set to 1, the timer is prescaled by timer 2. When set to 0, the timer counts up every fourth CLKOUT period. This bit is ignored when external clocking is enabled (EXT=1).

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

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

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

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

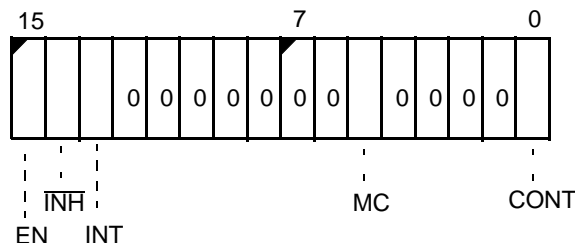
When CONT is set to 0, EN is cleared after each timer count sequence and the timer clears and then halts on reaching the maximum count. If CONT=0 and ALT=1, the timer counts to the maxcount compare A register value and resets, then it counts to the B register value and resets and halts.



### 8.3.3 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 (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

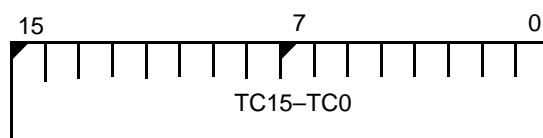
## Timer Count Registers

(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)

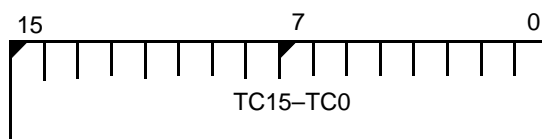
These registers serve as comparators for their associated count registers. Timer 0 and timer 1 each have two maximum count compare registers. See Figure 8-5.

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.



## 9.1 OVERVIEW

Direct memory access (DMA) permits transfer of data between memory and peripherals without CPU involvement. The DMA unit in the Am186ES and Am188ES microcontrollers provides two high-speed DMA channels. Data transfers can occur between memory and I/O spaces (e.g., memory to I/O) or within the same space (e.g., memory-to-memory or I/O-to-I/O). Either bytes or words can be transferred to or from even or odd addresses on the Am186ES microcontroller. (The Am188ES microcontroller does not support word transfers.) Two bus cycles (a minimum of eight clocks) are necessary for each data transfer.

Each channel accepts a DMA request from one of three sources: the 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.

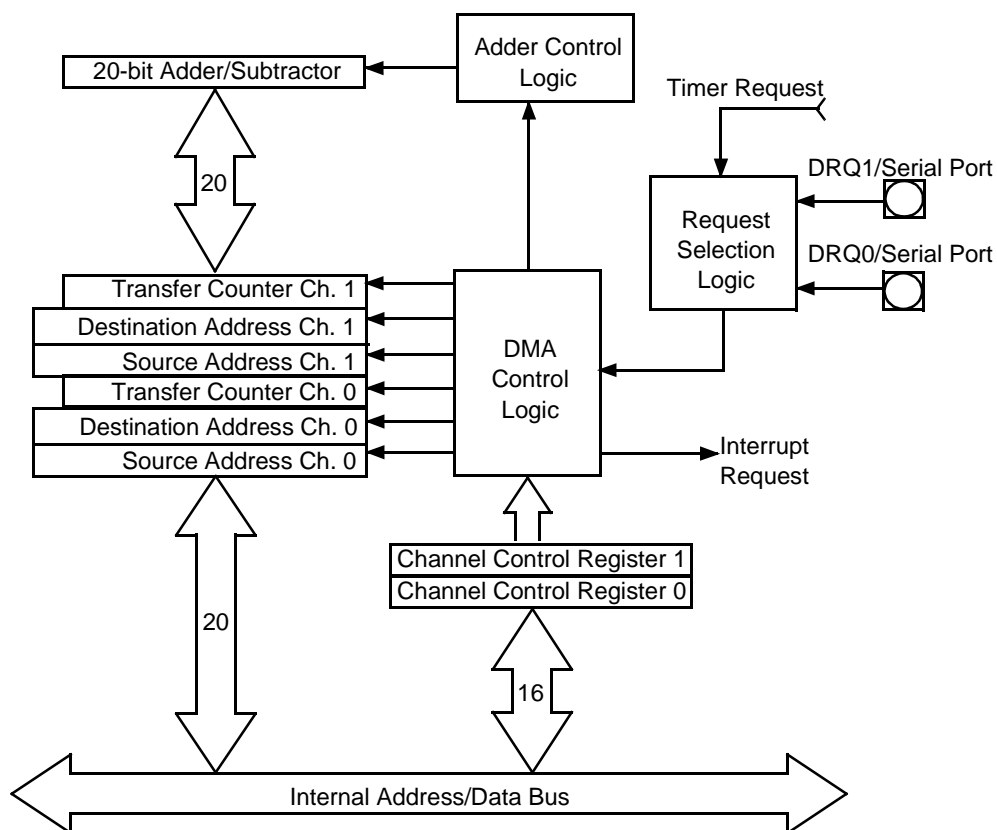
Table 9-1 DMA Controller Register Summary

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 Am186ES, up to 64 Kbytes or 64 Kwords can be transferred with automatic termination. The Am188ES does not support word transfers.

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.

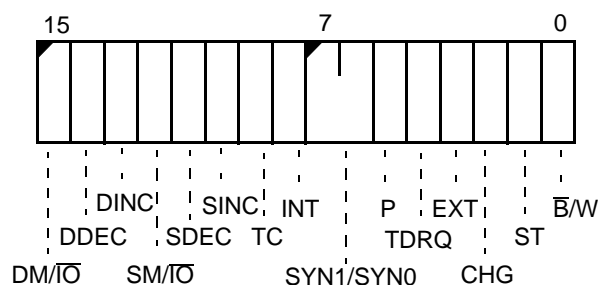
#### 9.3.1 DMA Control Registers (D0CON, Offset CAh, D1CON, Offset DAh)

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 (on the Am186 microcontroller only)
- Whether the 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/I/O)**—Selects memory or I/O space for the destination address. When DM/I/O is set to 1, the destination address is in memory space. When set to 0, the destination address is in I/O space.

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

**Bit 13: Destination Increment (DINC)**—When DINC is set to 1, the destination address is automatically incremented after each transfer. The address increments by 1 or 2 depending on the byte/word bit ( $\overline{B}/W$ , 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 Am186ES microcontroller, when  $\overline{B}/W$  is set to 1, word transfers are selected. When  $\overline{B}/W$  is set to 0, byte transfers are selected. Word transfers are not supported on the Am188ES microcontroller. 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 Am186ES and Am188ES 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 peripheral control 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.

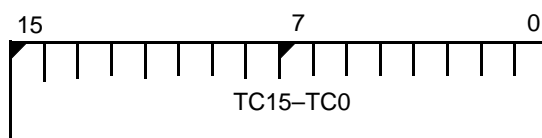
For DMA from the serial port, the receive data register address, either I/O mapped or memory mapped, should be specified as a byte source for the DMA by writing the address of the register into the DMA Source and DMA Source High registers. The source address (the address of the receive data register) 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.

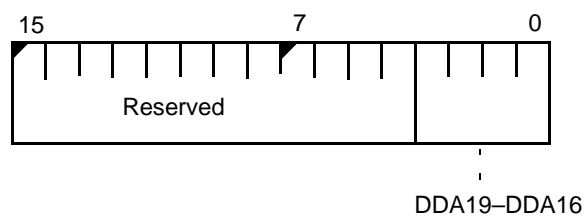
**Bits 15–0: DMA Transfer Count (TC15–TC0)**—Contains the transfer count for a DMA channel. Value is decremented by 1 after each transfer.

### 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 Am186ES microcontroller if all word transfers are performed to or from even addresses so that accesses occur in single 16-bit bus cycles.

Figure 9-4 DMA Destination Address High Register



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.

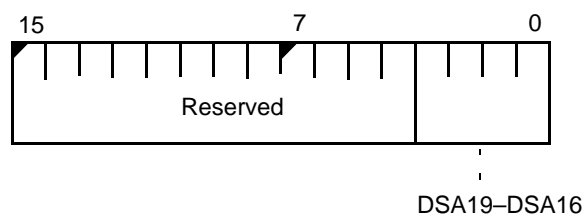
**Bits 15–0: DMA Destination Address Low (DDA15–DDA0)**—These bits are driven onto A15–A0 during the write phase of a DMA transfer.

### 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 Am186ES microcontroller if all word transfers are performed to or from even addresses so that accesses occur in single 16-bit bus cycles.

Figure 9-6 DMA Source Address High Register



The value of D0SRCH and D1SRCH at reset is undefined.

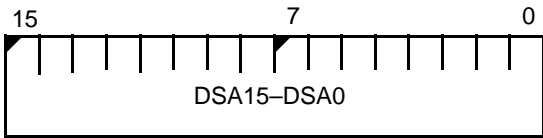
#### **Bits 15–4: Reserved**

**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

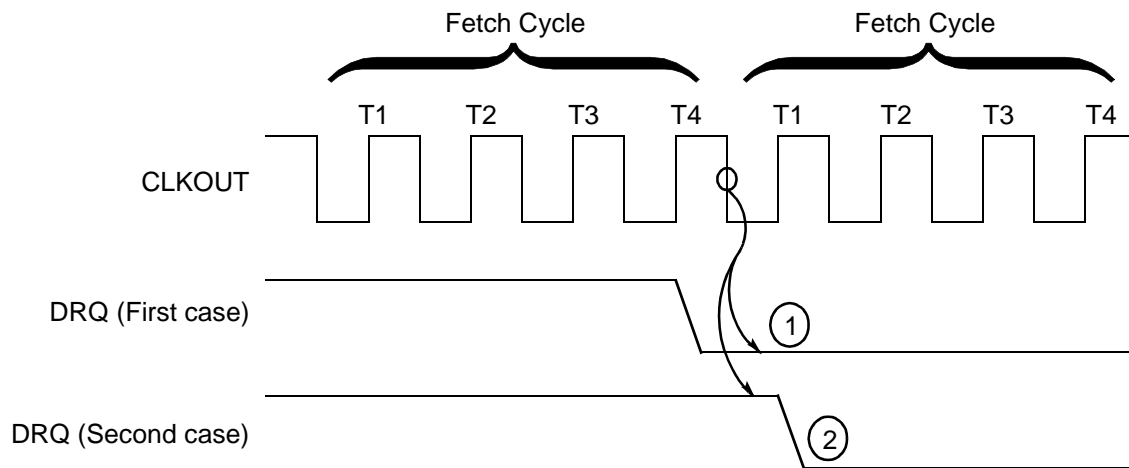
## 9.4.1 Synchronization Timing

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.

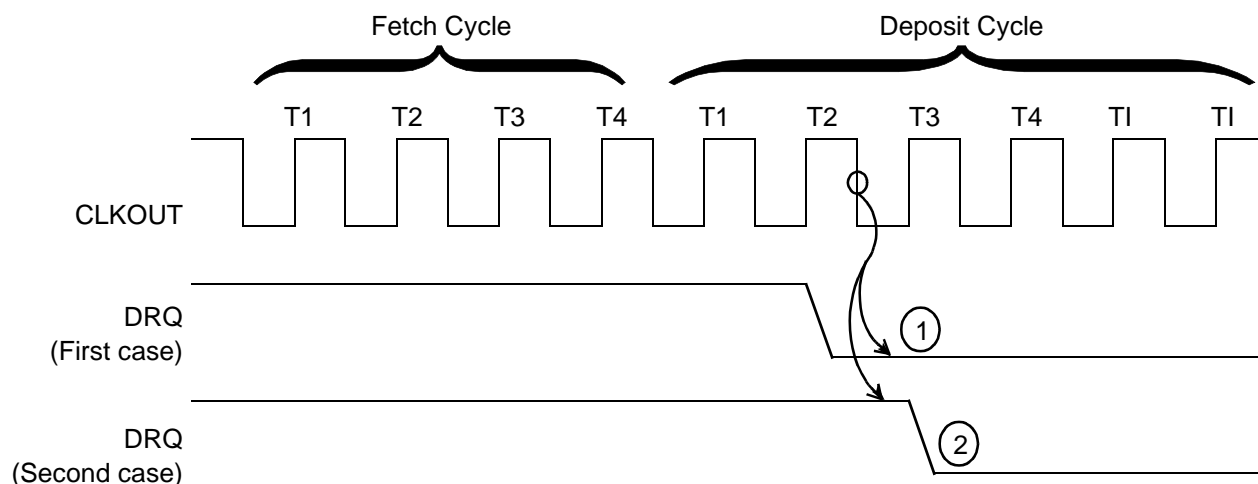
### 9.4.1.2 Destination Synchronization Timing

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.

## 10.1 OVERVIEW

The Am186ES and Am188ES 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 ( $\overline{\text{CTS}}$ )
  - Enable-receiver-request ( $\overline{\text{ENRX}}$ )
  - Ready-to-send ( $\overline{\text{RTS}}$ )
  - Ready-to-receive ( $\overline{\text{RTR}}$ )
- 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 Am186ES and Am188ES 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

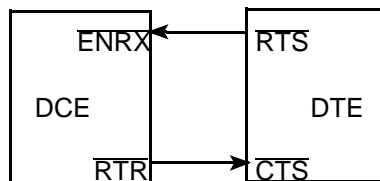
Pin Designation	Pin Function
RXD0, RXD1	Receives serial port data
TXD0, TXD1	Transmits serial port data
$\overline{\text{CTS}}0$ , $\overline{\text{CTS}}1$ / $\text{ENRX}0$ , $\text{ENRX}1$	Clear to send or enable receiver request
$\text{RTS}0$ , $\text{RTS}1$ / $\text{RTR}0$ , $\text{RTR}1$	Ready to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

Each port is provided with two data pins (RXD0/RXD1 and TXD0/TXD1) and two flow control signals ( $\overline{\text{RTS}}0$ ,  $\overline{\text{RTS}}1/\text{RTR}0$ ,  $\overline{\text{RTR}}1$ ). 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 DCE/DTE Protocol

The Am186ES and Am188ES 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 10.1.1.2.

Figure 10-1 DCE/DTE Protocol



$\overline{\text{RTS}}$  = Request to send output from transmitter

$\overline{\text{CTS}}$  = Clear to send input to transmitter

$\overline{\text{RTR}}$  = Ready to receive output from receiver

$\overline{\text{ENRX}}$  = Enable receiver request input to receiver

#### 10.1.1.2 DCE/DTE Protocol

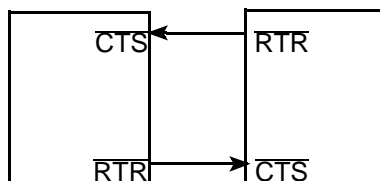
The Am186ES and Am188ES microcontrollers' serial ports can function as either DTE or DCE devices. To implement the DCE device, the ENRX bit should be set and the RTS bit should be cleared for the associated serial port. To implement the DTE device, the ENRX bit should be cleared and the RTS bit should 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  $\overline{\text{RTS}}$  signal. The  $\overline{\text{RTS}}$  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  $\overline{\text{RTR}}$  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



$\overline{\text{CTS}}$  = Clear to send input to transmitter

$\overline{\text{RTR}}$  = Ready to receive output from receiver

The Am186ES and Am188ES microcontrollers' serial ports 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  $\overline{\text{RTR}}$  signal. A device does not begin transmitting data until the  $\overline{\text{CTS}}$  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 Am186ES and Am188ES microcontrollers, the  $\overline{\text{RTR}}$  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 Am186ES and Am188ES 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.

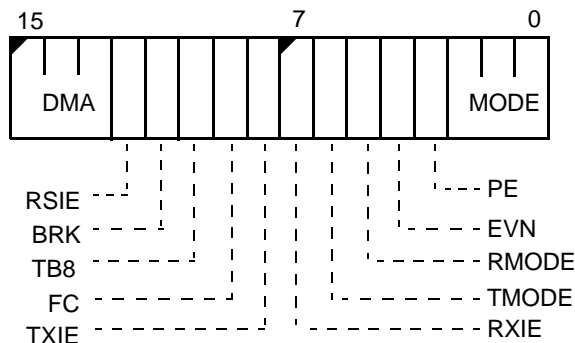
Table 10-2 Asynchronous Serial Port Register Summary

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 1 Transmit

## 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  $M = \text{start bit} + \text{data bits} (+ \text{parity bit}) + \text{stop bit}$ . The transmitter can be used to time the break by setting the BRK bit when the transmitter is empty (indicated by the TEMPTY bit of the serial port status register), writing the serial port transmit register with data, then waiting until the TEMPTY 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 TEMPTY 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 TEMPTY 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.

**Bit 10: Transmit Bit 8 (TB8)**—This bit is transmitted as the ninth data bit in modes 2 and 3 (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 TEMPTY 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 checking is enabled. When this bit is reset, parity checking is disabled.

**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	Reserved			

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.

This mode can be used in conjunction with mode 3 to allow for multidrop communications over a common serial link. In this case, the serial port is configured as mode 2 initially. Each time data is received with the ninth bit set, the data is compared by software against a unique ID for this receiver. If the received data does not match the port ID, the port is left in mode 2. If the received data matches the port ID, software should reconfigure the serial port to mode 3, allowing it to receive 9-bit data with the ninth bit reset.

In a serial multidrop configuration, multiple serial ports are attached to the same serial line. The master serial port is configured in mode 3 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 mode 3. The two mode 3 ports are able to exchange data freely.

It should be noted that only ports which are actively exchanging data (i.e. ports in mode 3) 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 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.

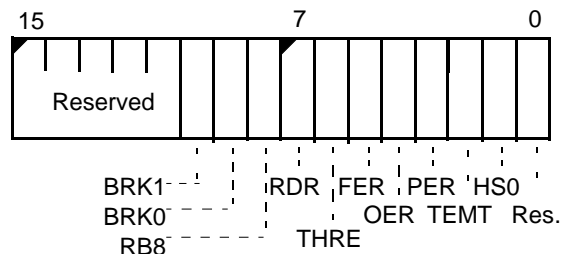
This mode can be used in conjunction with mode 2 (see above) to allow for multidrop communications over a common serial link. In this case, parity must be disabled. In this configuration, software interprets receive characters as data as long as the ninth data bit is reset (Low). When a character is 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.

## 10.2.2 Serial Port 0/1 Status Registers (SPOSTS/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  $M = (\text{start bit} + \# \text{ data bits} + \# \text{ parity bits} + \text{stop bit})$ .

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  $M = (\text{start bit} + \# \text{ data bits} + \# \text{ parity bits} + \text{stop bit})$ .

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 and 3. (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**

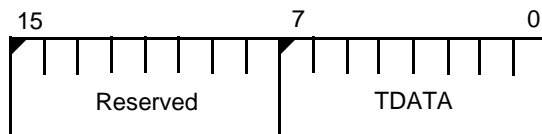
### 10.2.3 Serial Port 0/1 Transmit Registers (SP0TD/SP1TD, Offset 84h/14h)

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 state of the transmit and transmit shift registers is reflected in the TEND and THRE bits in the associated Serial Port Status register.

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

The serial port transmit register in the Am186EM and Am188EM microcontrollers is renamed in the Am186ES and Am188ES 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.

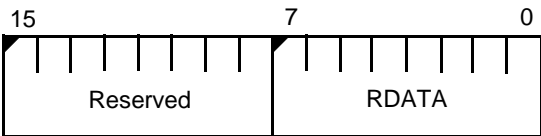
10.2.4 Serial Port 0/1 Receive Registers  
(SPORD/SP1RD, Offset 86h/16h)

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 (SPOBAUD/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:

$$\text{BAUDDIV} = (\text{Processor Frequency} \div (16 \cdot \text{baud rate}))$$

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 Kb at 20 MHz, 1562.5 Kb at 25MHz, 2062.5 Kb at 33 MHz, and 1250 Kb 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

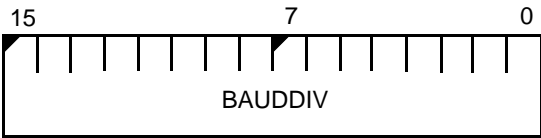
Baud Rate	Divisor Based on CPU Clock Rate			
	20 MHz	25 MHz	33 MHz	40 MHz
300	4166	5208	6875	8333
600	2083	2604	3437	4166
1050	1190	1488	1964	2380
1200	1041	1302	1718	2083
1800	694	868	1145	1388
2400	520	651	859	1041
4800	260	325	429	520
7200	173	217	286	347
9600	130	162	214	260
19200	65	81	107	130
28800	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

Baud Rate	Divisor Based on CPU Clock Rate			
	20 MHz	25 MHz	33 MHz	40 MHz
128000	9	12	16	19
153600	8	10	13	16

Special	15 MHz	21 MHz	24 MHz	30 MHz
187500	5	7	8	10

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

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.



## 11.1 OVERVIEW

Thirty-two pins on the Am186ES and Am188ES 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.

Figure 11-1 Programmable I/O Pin Operation

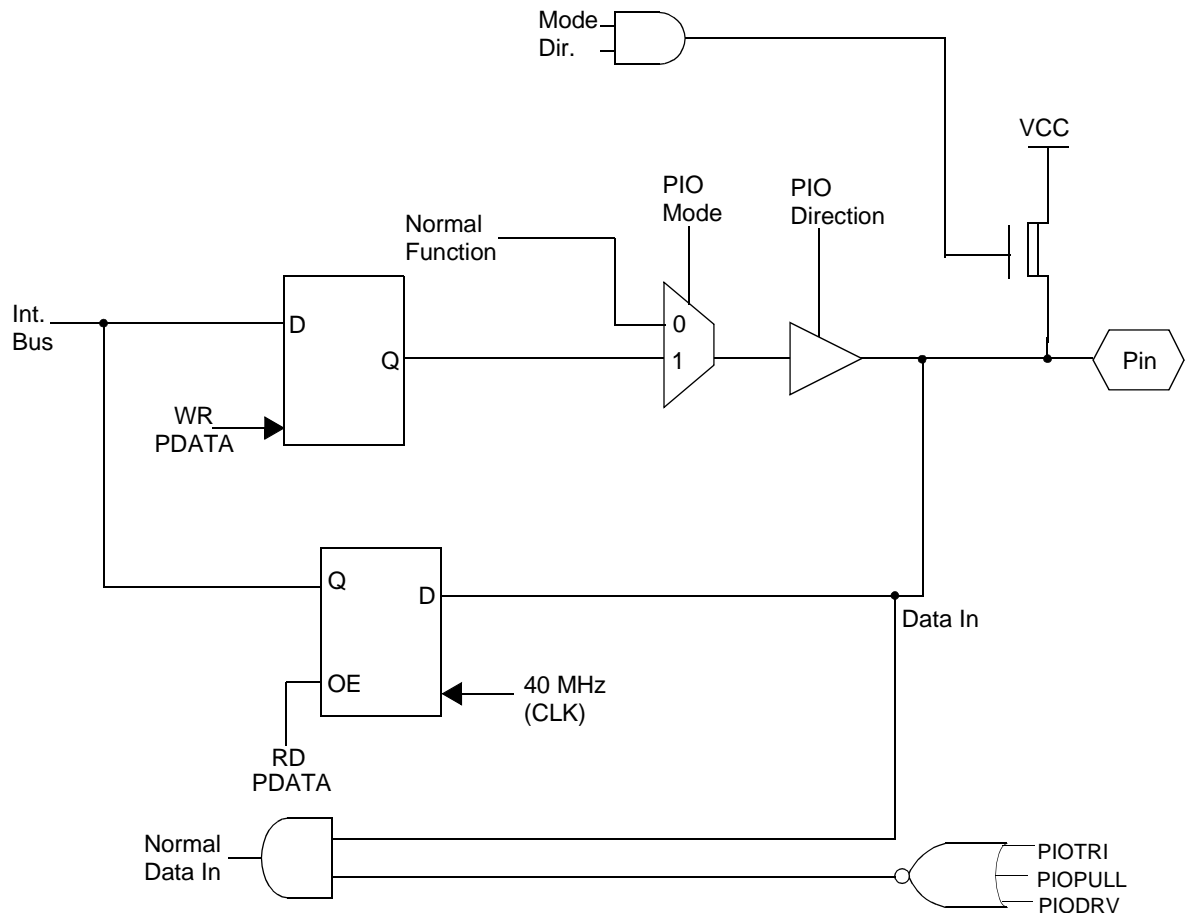


Table 11-1 PIO Pin Assignments

PIO No	Associated Pin	Power-On Reset Status
0	TMRIN1	Input with pullup
1	TMROUT1	Input with pulldown
2	$\overline{\text{PCS6/A2}}$	Input with pullup
3	$\overline{\text{PCS5/A1}}$	Input with pullup
4	DT/ $\overline{\text{R}}$	Normal operation <sup>(3)</sup>
5	$\overline{\text{DEN/DS}}$	Normal operation <sup>(3)</sup>
6	SRDY	Normal operation <sup>(4)</sup>
7 <sup>(1)</sup>	A17	Normal operation <sup>(3)</sup>
8 <sup>(1)</sup>	A18	Normal operation <sup>(3)</sup>
9 <sup>(1)</sup>	A19	Normal operation <sup>(3)</sup>
10	TMROUT0	Input with pulldown
11	TMRIN0	Input with pullup
12	DRQ0/INT5	Input with pullup
13	DRQ1/INT6	Input with pullup
14	$\overline{\text{MCS0}}$	Input with pullup
15	$\overline{\text{MCS1}}$	Input with pullup
16	$\overline{\text{PCS0}}$	Input with pullup
17	$\overline{\text{PCS1}}$	Input with pullup
18	$\overline{\text{PCS2/CTS1/ENRX1}}$	Input with pullup
19	$\overline{\text{PCS3/RTS1/RTR1}}$	Input with pullup
20	$\overline{\text{RTS0/RTR0}}$	Input with pullup
21	$\overline{\text{CTS0/ENRX0}}$	Input with pullup
22	TXD0	Input with pullup
23	RXD0	Input with pullup
24	$\overline{\text{MCS2}}$	Input with pullup
25	$\overline{\text{MCS3/RFSH}}$	Input with pullup
26 <sup>(1,2)</sup>	UZI	Input with pullup
27	TXD1	Input with pullup
28	RXD1	Input with pullup
29 <sup>(1,2)</sup>	$\overline{\text{S6/LOCK/CLKDIV2}}$	Input with pullup
30	INT4	Input with pullup
31	INT2/ $\overline{\text{INTA0/PWD}}$	Input with pullup

**Notes:**

1. These pins are used by emulators. (Emulators also use  $\overline{\text{S2-S0}}$ ,  $\overline{\text{RES}}$ , NMI, CLKOUTA,  $\overline{\text{BHE}}$ , ALE, AD15-AD0, and A16-A0.)
2. These pins revert to normal operation if  $\overline{\text{BHE/ADEN}}$  (186) or  $\overline{\text{RFSH2/ADEN}}$  (188) 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 Am186ES and Am188ES 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)

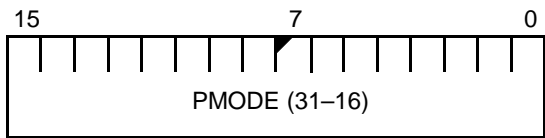
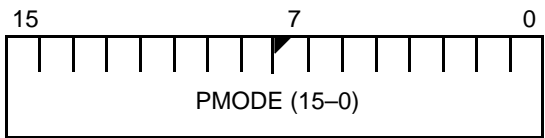


Figure 11-3 PIO Mode 0 Register (PIOMODE0, offset 70h)



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.

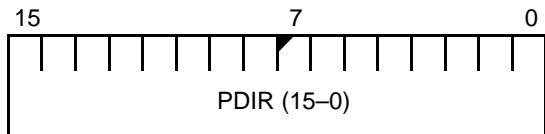
11.3 PIO DIRECTION REGISTERS

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)

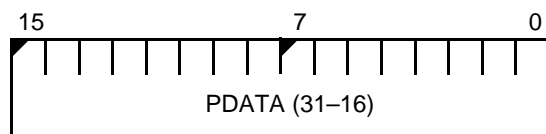
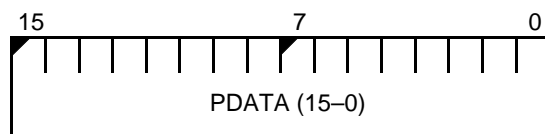


Figure 11-7 PIO Data 0 Register (PDATA0, offset 74h)



### 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.



# 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

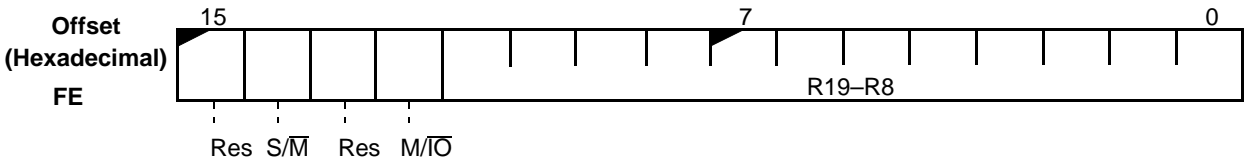
Hex Offset	Mnemonic	Register Description	Comment
FE	RELREG	Peripheral control block relocation register	
F6	RESCON	Reset configuration register	
F4	PRL	Processor release level register	
F2	AUXCON	Auxiliary configuration	
F0	SYSCON	System configuration register	
E6	WDT	Watchdog timer control register	
E4	EDRAM	Enable RCU register	
E2	CDRAM	Clock prescaler register	
E0	MDRAM	Memory partition 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	$\overline{PCS}$ and $\overline{MCS}$ 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
36	DMA1CON/ INT6CON	DMA 1 interrupt control register/INT6	Slave & master

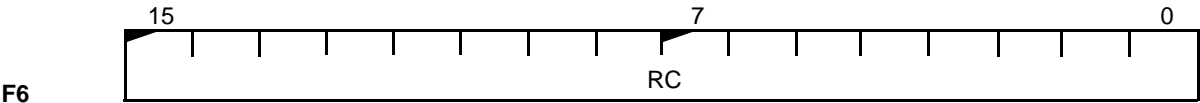


Hex Offset	Mnemonic	Register Description	Comment
34	DMA0CON/ INT5CON	DMA 0 interrupt control register/INT5	Slave & master
32	TCUCON	Timer interrupt control register	Master mode
	T0INTCON	Timer 0 interrupt control register	Slave mode
30	INTSTS	Interrupt status register	Slave & master
2E	REQST	Interrupt 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	

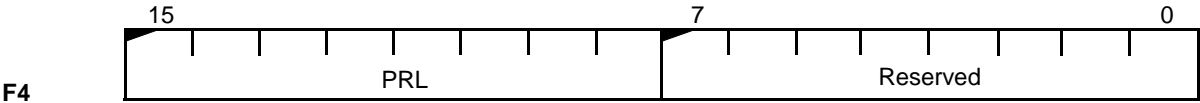
Figure A-1 Internal Register Summary



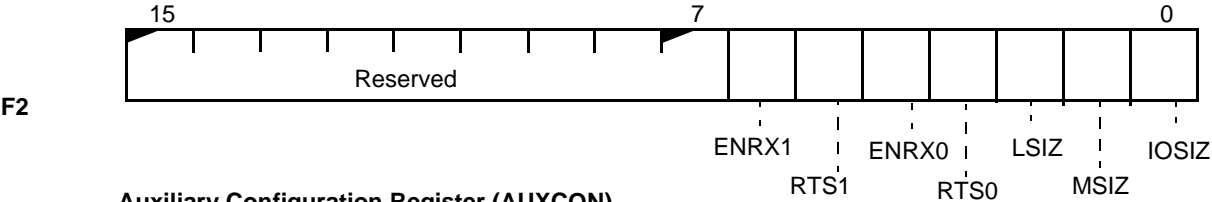
Peripheral Control Block Relocation Register (RELREG)  
Page 4-3



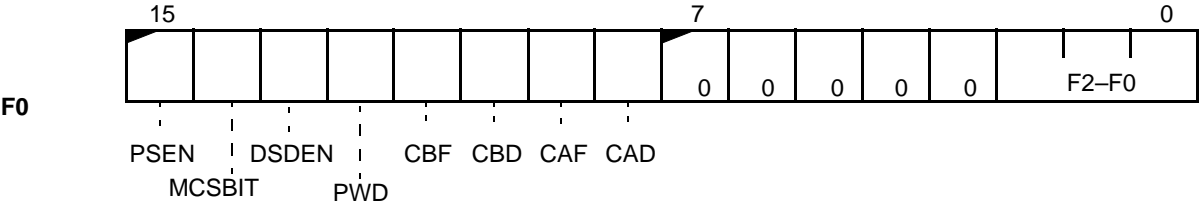
Reset Configuration Register (RESCON)  
Page 4-4



Processor Release Level Register (PRL)  
Page 4-5



Auxiliary Configuration Register (AUXCON)  
Page 4-6



System Configuration Register (SYSCON)  
Page 4-7

Figure A-1 Internal Register Summary (continued)

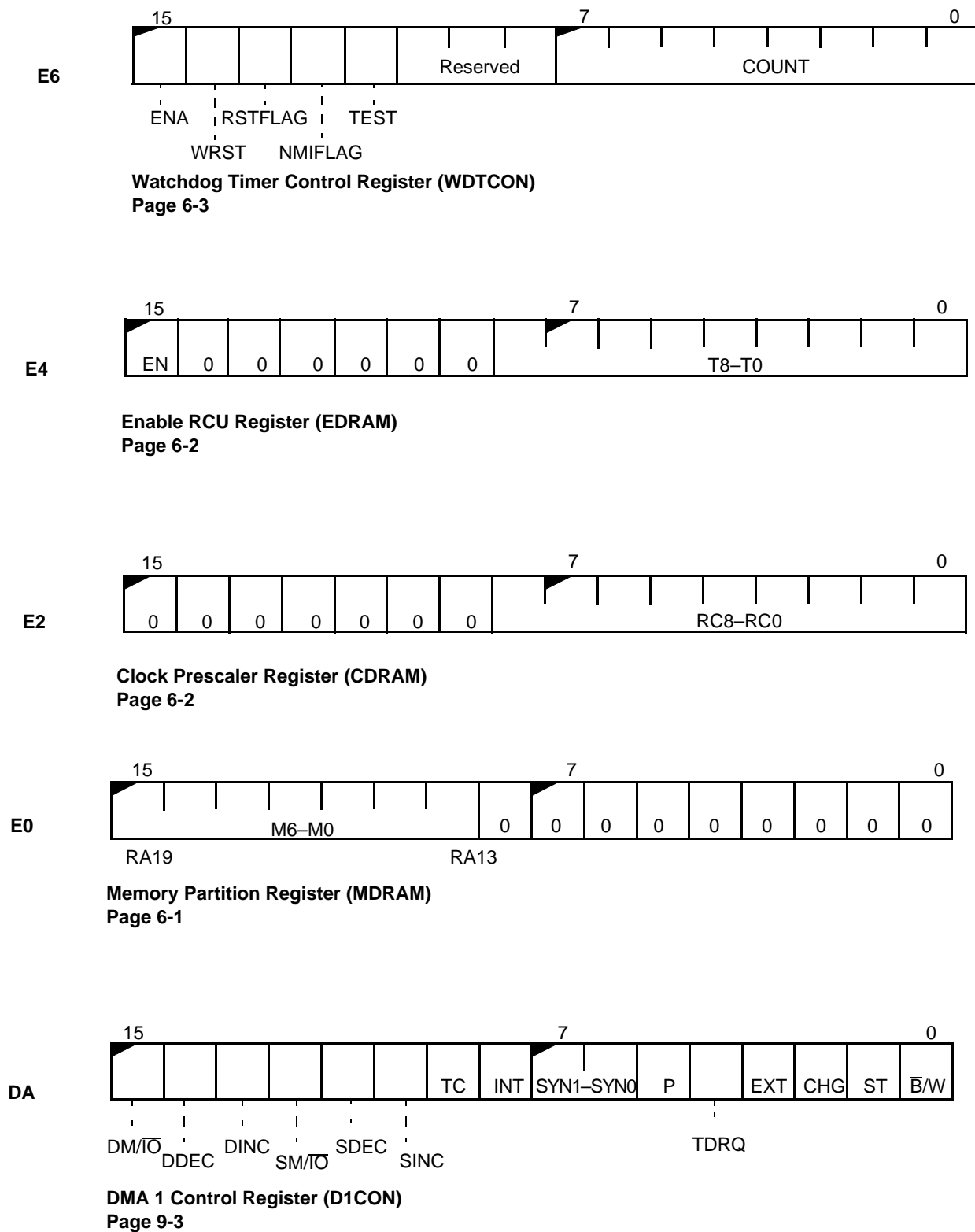
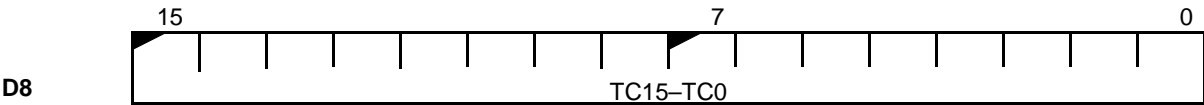
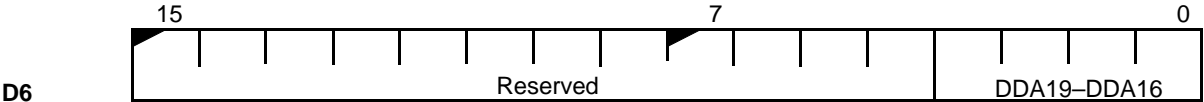


Figure A-1 Internal Register Summary (continued)



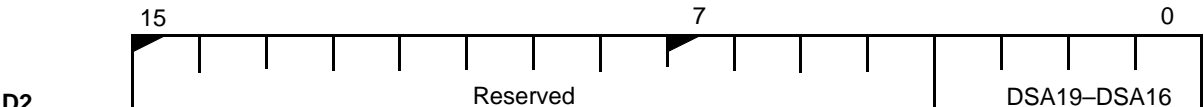
DMA 1 Transfer Count Register (D1TC)  
Page 9-6



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



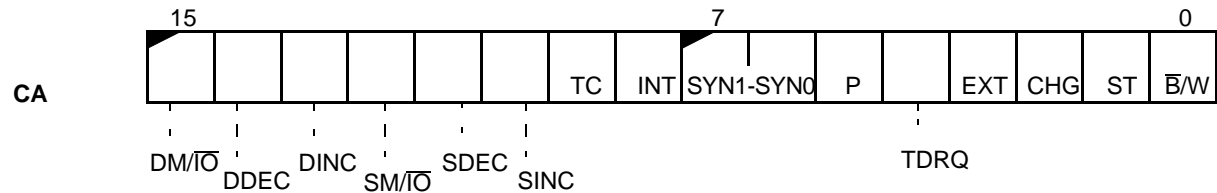
DMA 1 Destination Address Low Register (D1DSTL)  
Page 9-8



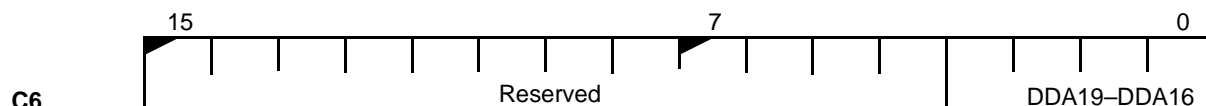
DMA 1 Source Address High Register (D1SRCH)  
Page 9-9



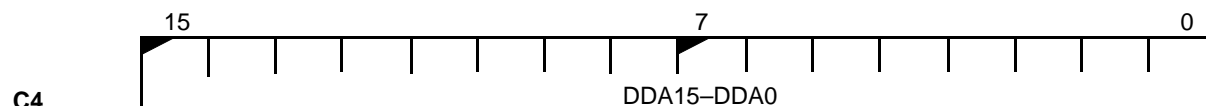
DMA 1 Source Address Low Register (D1SRCL)  
Page 9-10



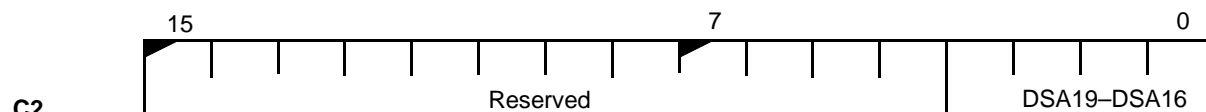
DMA 0 Control Register (D0CON)  
Page 9-3

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

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



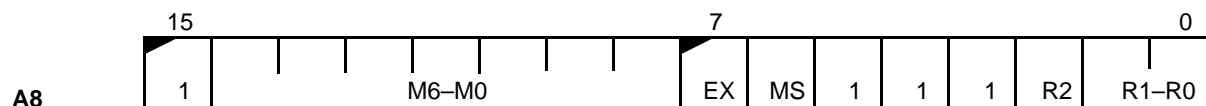
**DMA 0 Destination Address Low Register (D0DSTL)**  
Page 9-8



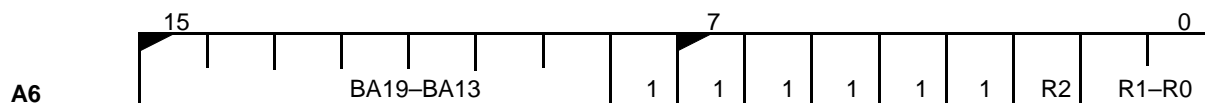
**DMA 0 Source Address High Register (D0SRCH)**  
Page 9-9



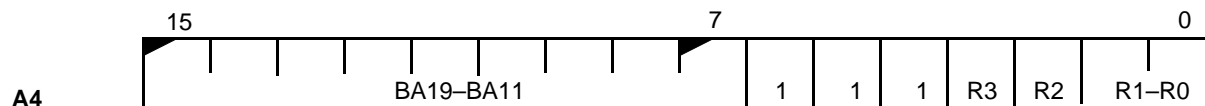
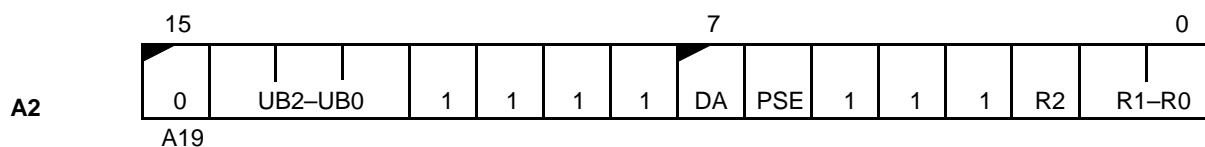
**DMA 0 Source Address Low Register (D0SRCL)**  
Page 9-10



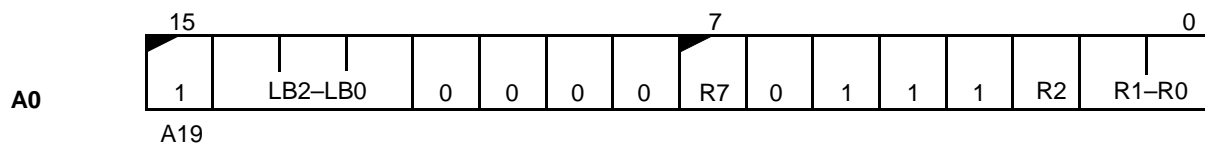
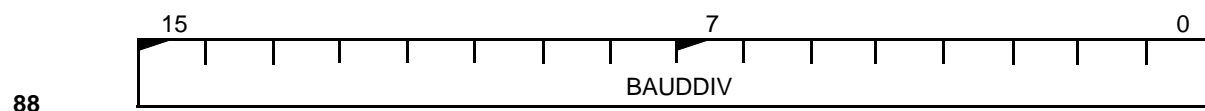
**PCS and MCS Auxiliary Register (MPCS)**  
**Page 5-10**



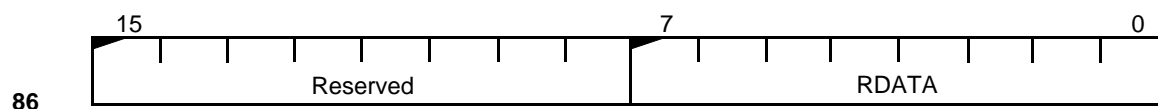
**Midrange Memory Chip Select Register (MMCS)**  
Page 5-8

Peripheral Chip Select Register (PACS)  
Page 5-12

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

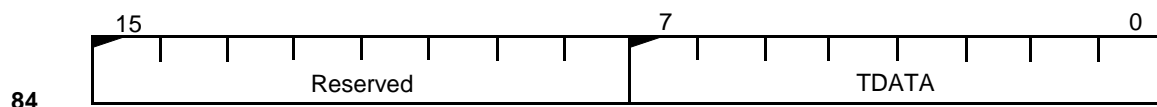
Upper Memory Chip Select Register (UMCS)  
Page 5-4

**Serial Port 0 Baud Rate Divisor Register (SP0BAUD)**  
**Page 10-14**

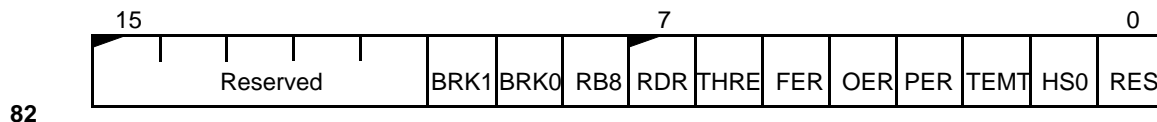


**Serial Port 0 Receive Register (SP0RD)**  
**Page 10-12**

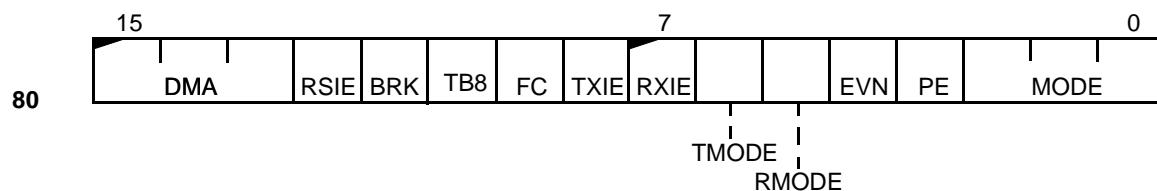
Figure A-1 Internal Register Summary (continued)



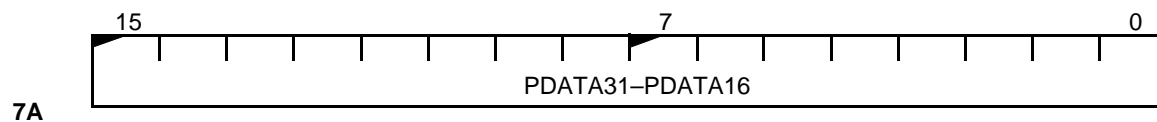
**Serial Port 0 Transmit Register (SP0TD)**  
Page 10-11



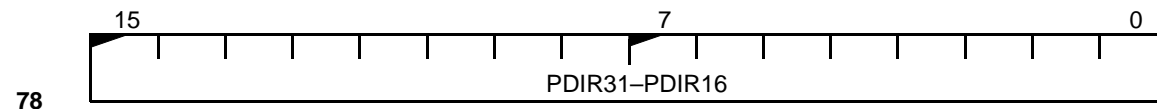
**Serial Port 0 Status Register (SP0STS)**  
Page 10-9



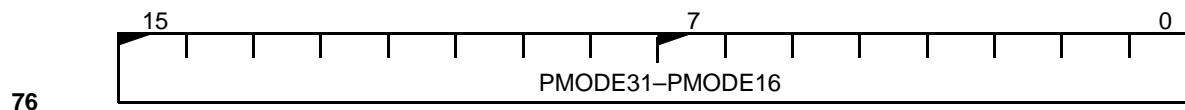
**Serial Port 0 Control Register (SP0CT)**  
Page 10-5



**PIO Data 1 Register (PDATA1)**  
Page 11-5

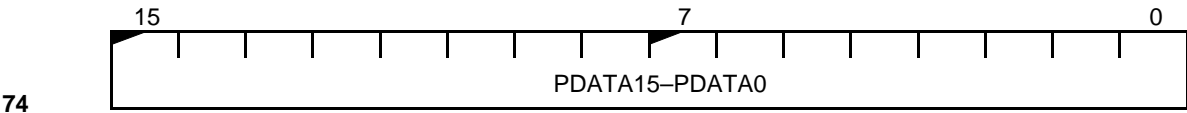


**PIO Direction 1 Register (PDIR1)**  
Page 11-4

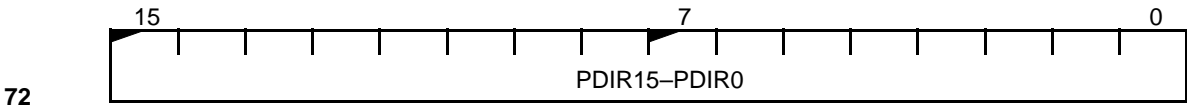


**PIO Mode 1 Register (PIOMODE1)**  
Page 11-3

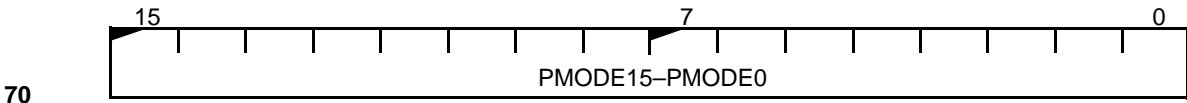
Figure A-1 Internal Register Summary (continued)



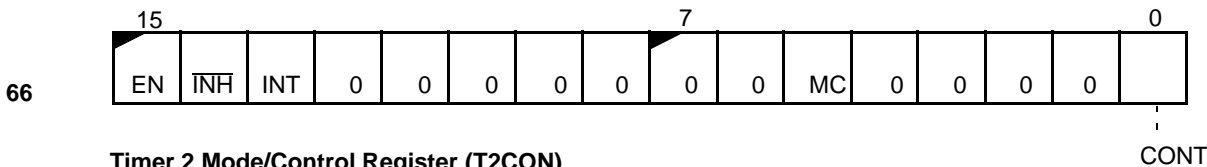
**PIO Data 0 Register (PDATA0)**  
Page 11-5



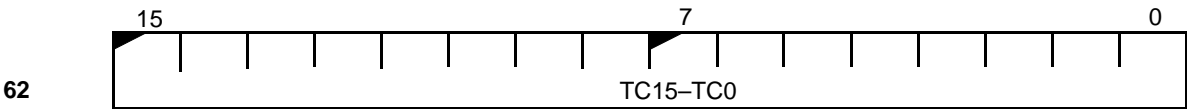
**PIO Direction 0 Register (PDIR0)**  
Page 11-4



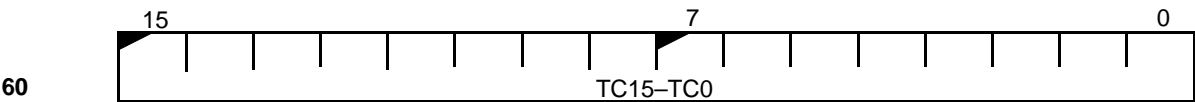
**PIO Mode 0 Register (PIOMODE0)**  
Page 11-3



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



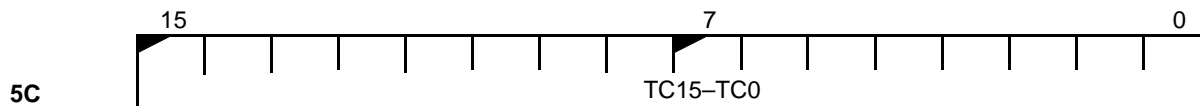
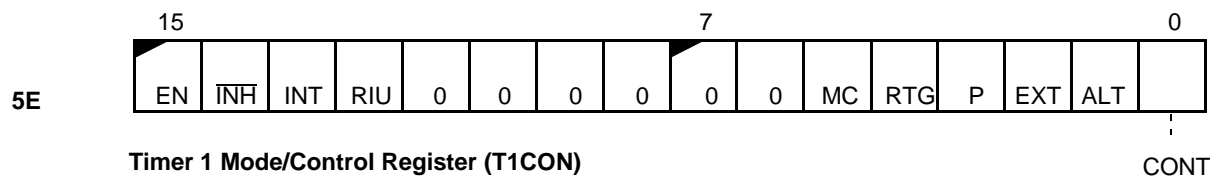
**Timer 2 Maxcount Compare A Register (T2CMPA)**  
Page 8-7



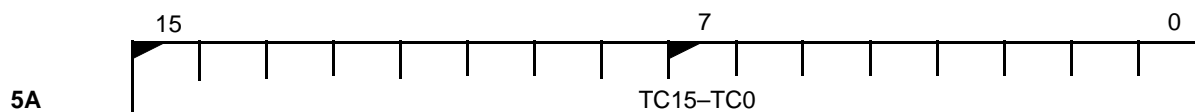
**Timer 2 Count Register (T2CNT)**  
Page 8-6



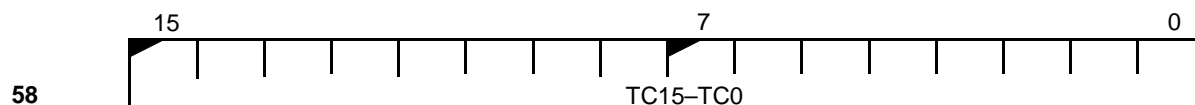
Figure A-1 Internal Register Summary (continued)



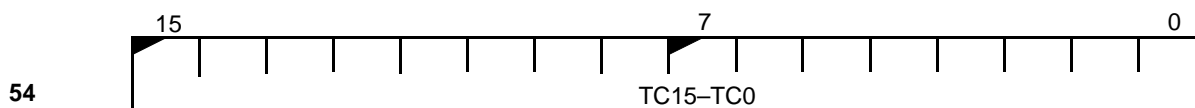
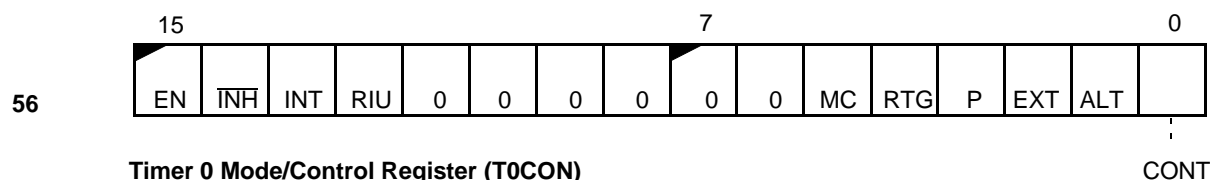
Timer 1 Maxcount Compare B Register (T1CMPB)  
Page 8-7



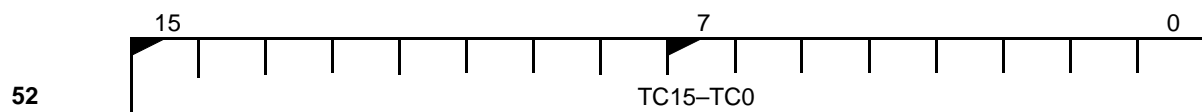
Timer 1 Maxcount Compare A Register (T1CMPA)  
Page 8-7



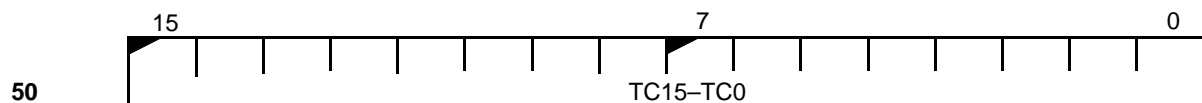
Timer 1 Count Register (T1CNT)  
Page 8-6



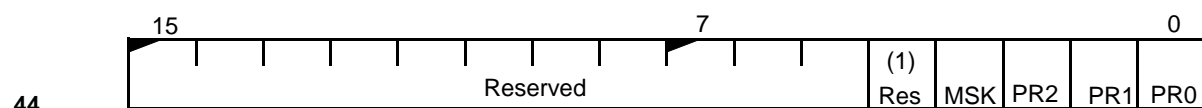
Timer 0 Maxcount Compare B Register (T0CMPB)  
Page 8-7



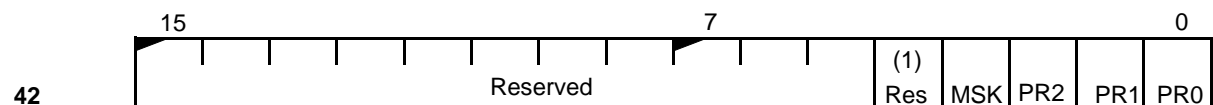
**Timer 0 Maxcount Compare A Register (T0CMPA)**  
Page 8-7



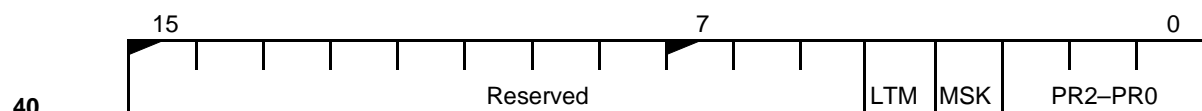
**Timer 0 Count Register (T0CNT)**  
**Page 8-6**



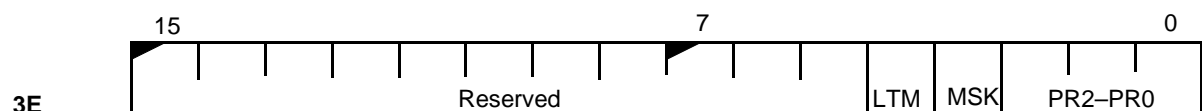
**Serial Port 0 Interrupt Control Register (SP0CON)**  
**Master Mode**  
**Page 7-18**



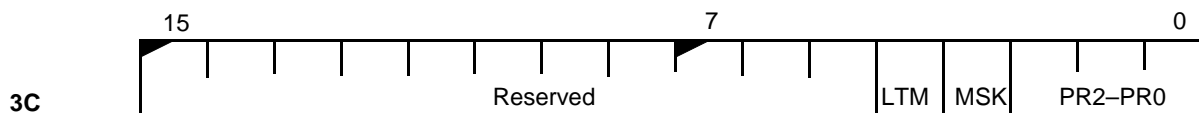
**Serial Port 1 Interrupt Control Register (SP1CON)**  
**Master Mode**  
**Page 7-18**



**INT4 Control Register (I4CON)**  
**Master Mode**  
**Page 7-16**



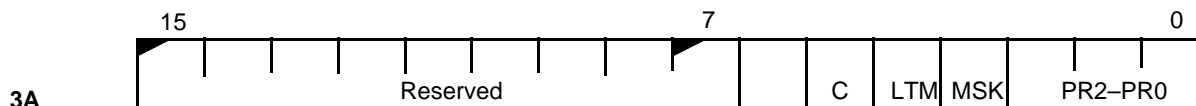
**INT3 Control Register (I3CON)**  
**Master Mode**  
**Page 7-15**



### INT2 Control Register (I2CON)

## Master Mode

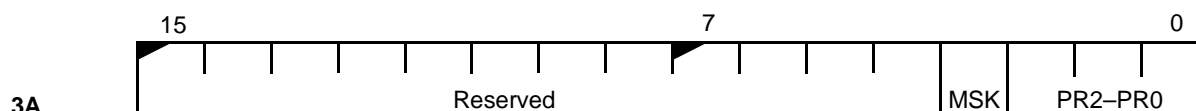
Page 7-15



### INT1 Control Register (I1CON)

## Master Mode

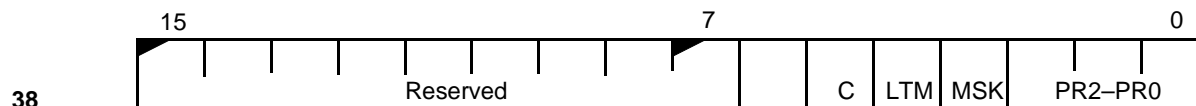
Page 7-14



### Timer 2 Interrupt Control Register (T2INTCON)

## Slave Mode

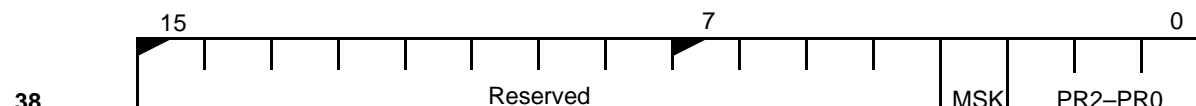
Page 7-29



### INT0 Control Register (I0CON)

## Master Mode

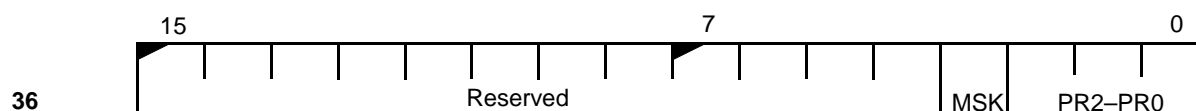
Page 7-14



### Timer 1 Interrupt Control Register (T1INTCON)

## Slave Mode

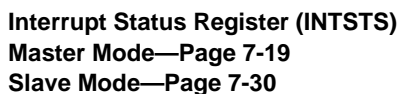
Page 7-29

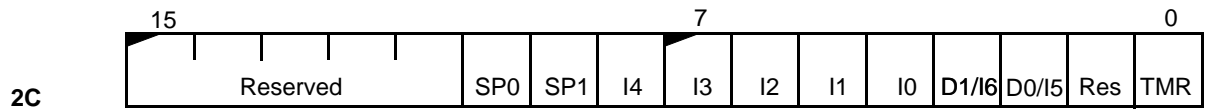


### DMA 1 Interrupt Control Register (DMA1CON)/INT6CON

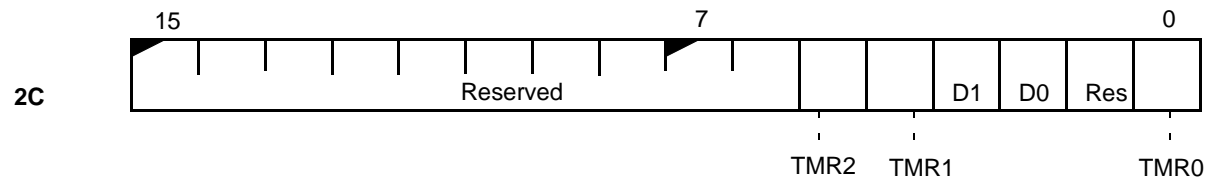
Master Mode—Page 7-17

Slave Mode—Page 7-29

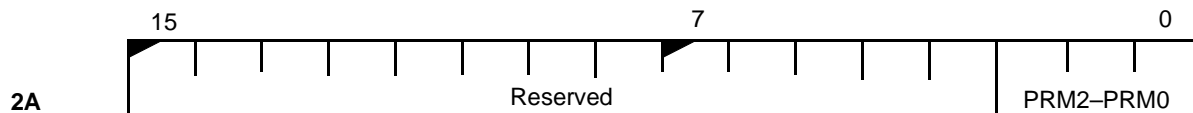




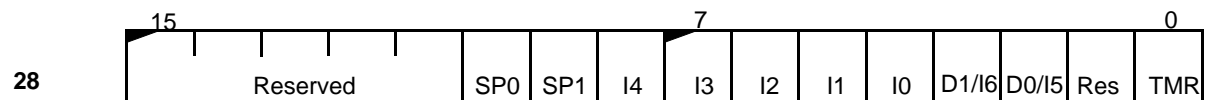
**Interrupt In-Service Register (INSERV)**  
**Master Mode**  
**Page 7-22**



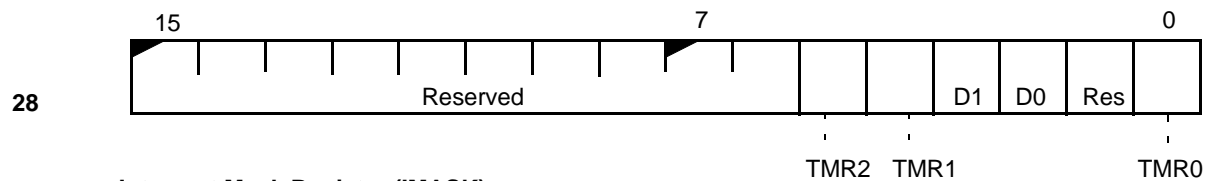
**In-Service Register (INSERV)  
Slave Mode  
Page 7-32**



**Priority Mask Register (PRIMSK)**  
**Master Mode—Page 7-23**  
**Slave Mode—Page 7-33**

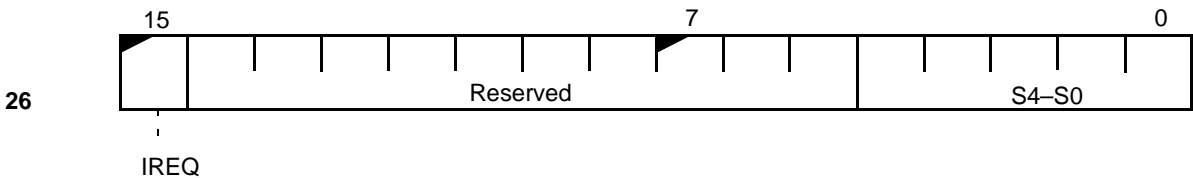


**Interrupt Mask Register (IMASK)**  
**Master Mode**  
**Page 7-24**

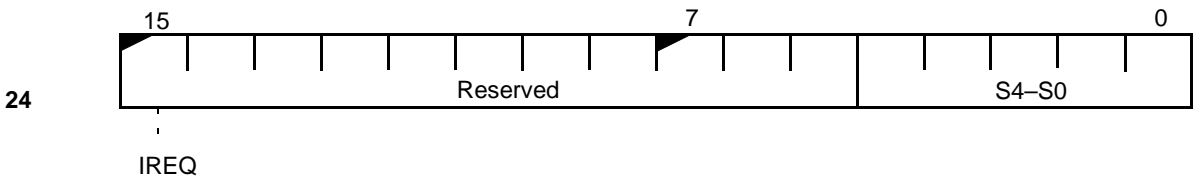


**Interrupt Mask Register (IMASK)**  
**Slave Mode**  
**Page 7-34**

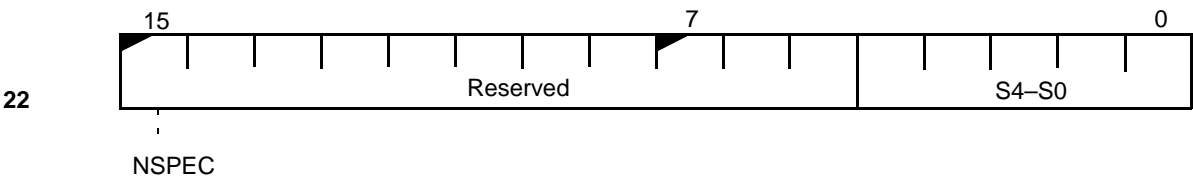
Figure A-1 Internal Register Summary (continued)



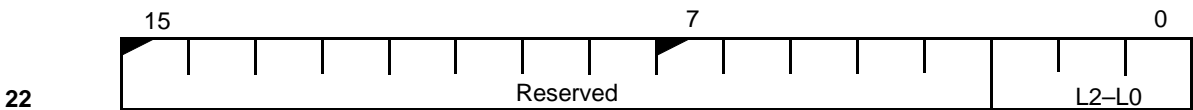
**Poll Status Register (POLLST)**  
**Master Mode**  
**Page 7-25**



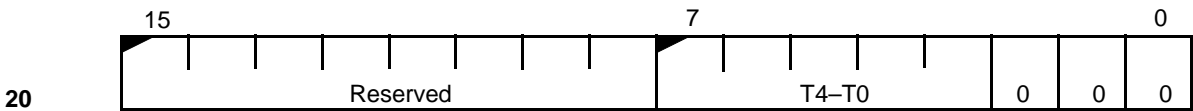
**Poll Register (POLL)**  
**Master Mode**  
**Page 7-26**



**End-of-Interrupt Register (EOI)**  
**Master Mode**  
**Page 7-27**



**Specific End-of-Interrupt Register (EOI)**  
**Slave Mode**  
**Page 7-35**



**Interrupt Vector Register (INTVEC)**  
**Slave Mode**  
**Page 7-36**

**Serial Port 1 Baud Rate Divisor Register (SP1BAUD)**  
**Page 10-14**

**Serial Port 1 Receive Register (SP1RD)**  
Page 10-12

**Serial Port 1 Transmit Register (SP1TD)**  
**Page 10-11**

**Serial Port 1 Status Register (SP1STS)**  
**Page 10-9**

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





## A

ALT bit (Alternate Compare Bit)  
 Timer 0 Mode/Control Register 8-4  
 Timer 1 Mode/Control Register 8-4  
 Am186ES microcontroller  
 design philosophy xiii  
 product support v

## B

B/W bit (Byte/Word Select) 9-5  
 BA19-BA11 field (Base Address)  
 Peripheral Chip Select Register 5-12  
 BA19-BA13 field (Base Address)  
 Midrange Memory Chip Select Register 5-8  
 BAUDDIV field (Baud Rate Divisor) 10-14  
 bits  
 ALT (Alternate Compare Bit) 8-4  
 B/W (Byte/Word Select) 9-5  
 BA19-BA11 (Base Address) 5-12  
 BA19-BA13 (Base Address) 5-8  
 BAUDDIV (Baud Rate Divisor) 10-14  
 BRK1 (Long Break Detected) 10-9  
 C (Cascade Mode) 7-14  
 CHG (Change Start Bit) 9-4  
 CONT (Continuous Mode Bit) 8-4, 8-5  
 D1-D0 (DMA Channel Interrupt In-Service) 7-22,  
 7-32, 7-34  
 D1-D0 (DMA Channel Interrupt Masks) 7-24  
 D1-D0 (DMA Channel Interrupt Request) 7-31  
 DDA15-DDA0 (DMA Destination Address Low) 9-8  
 DDA19-DDA16 (DMA Destination Address High) 9-  
 7  
 DDEC (Destination Decrement) 9-3  
 DHLT (DMA Halt) 7-19, 7-30  
 DINC (Destination Increment) 9-4  
 DM/IO (Destination Address Space Select) 9-3  
 DMA (DMA Control Field) 10-5  
 DSA15-DSA0 (DMA Source Address Low) 9-10  
 DSA19-DSA16 (DMA Source Address High) 9-9

E (Enable RCU) 6-2  
 EN (Enable Bit) 8-3, 8-5  
 EX (Pin Selector) 5-11  
 EXT (External Clock Bit) 8-3  
 F2-F0 (Clock Divisor Select) 4-8  
 FER (Framing Error Detected) 10-10  
 HS0 (Handshake Signal 0) 10-10  
 I4-I0 (Interrupt In-Service) 7-22  
 I4-I0 (Interrupt Mask) 7-24  
 I4-I0 (Interrupt Requests) 7-20  
 INH (Inhibit Bit) 8-3, 8-5  
 INT (Interrupt Bit) 8-3, 8-5  
 INT (Interrupt) 9-4  
 IREQ (Interrupt Request) 7-25, 7-26  
 L2-L0 (Interrupt Type) 7-35  
 LB2-LB0 (Lower Boundary) 5-4  
 LTM (Level-Triggered Mode) 7-14, 7-15, 7-16  
 M/IO (Memory/I/O Space) 4-3  
 M6-M0 (MCS Block Size) 5-10  
 M6-M0 (Refresh Base) 6-1  
 MC (Maximum Count Bit) 8-3, 8-5  
 MS (Memory/I/O Space Selector) 5-11  
 MSK (Interrupt Mask) 7-17  
 MSK (Mask) 7-14, 7-15, 7-16, 7-18, 7-29  
 NSPEC (NonSpecific EOI) 7-27  
 OER (Overrun Error Detected) 10-10  
 P (Prescaler Bit) 8-3  
 P (Relative Priority) 9-4  
 PDATA15-PDATA0 (PIO Data Blts) 11-5  
 PDATA31-PDATA16 (PIO Data Blts) 11-5  
 PDIR15-PDIR0 (PIO Direction Bits) 11-4  
 PDIR31-PDIR16 (PIO Direction Bits) 11-4  
 PER (Parity Error Detected) 10-10  
 PMODE15-PMODE0 (PIO Mode Bits) 11-3  
 PMODE31-PMODE16 (PIO Mode Bits) 11-3  
 PR2-PR0 (Priority Level) 7-29  
 PR2-PR0 (Priority) 7-14, 7-15, 7-16, 7-17, 7-18  
 PRM2-PRM0 (Priority Field Mask) 7-23, 7-33  
 PSE (PSRAM Mode Enable) 5-7

R19-R8 (Relocation Address Bits) 4-3  
 R1-R0 (Wait State Value) 5-5, 5-7, 5-9, 5-11  
 R2 (Ready Mode) 5-5, 5-7, 5-9, 5-11  
 R7 (Address Disable) 5-5, 5-7  
 RB8 (Received Bit 8) 10-9  
 RC (Reset Configuration) 4-4  
 RC8-RC0 (Refresh Counter Reload Value) 6-2  
 RDATA (Receive Data) 10-12  
 RDR (Receive Data Ready) 10-9  
 RIU (Register in Use) 8-3  
 RTG (Retrigger Bit) 8-3  
 S/M (Slave/Master) 4-3  
 S4-S0 (Poll Status) 7-25, 7-26  
 S4-S0 (Source Vector Type) 7-27  
 SDEC (Source Decrement) 9-4  
 SFNM (Special Fully Nested Mode) 7-14  
 SINC (Source Increment) 9-4  
 SM/IO (Source Address Space Select) 9-4  
 SPI (Serial Port Interrupt In-Service) 7-22  
 SPI (Serial Port Interrupt Mask) 7-24  
 SPI (Serial Port Interrupt Request) 7-20  
 ST (Start/Stop DMA Channel) 9-5  
 SYN1-SYN0 (Synchronization Type) 9-4  
 T4-T0 (Interrupt Type) 7-36  
 T8-T0 (Refresh Count) 6-2  
 TC (Terminal Count) 9-4  
 TC15-TC0 (Timer Compare Value) 8-7  
 TC15-TC0 (Timer Count Register) 9-6  
 TC15-TC0 (Timer Count Value) 8-6  
 TDATA (Transmit Data) 10-11  
 TDRQ (Timer Enable/Disable Request) 9-4  
 TEMT (Transmitter Empty) 10-10  
 THRE (Transmit Holding Register Empty) 10-9  
 TMR (Timer Interrupt In-Service) 7-22  
 TMR (Timer Interrupt Mask) 7-24  
 TMR (Timer Interrupt Request) 7-21  
 TMR0 (Timer 0 Interrupt In-Service) 7-32  
 TMR0 (Timer 0 Interrupt Mask) 7-34  
 TMR0 (Timer 0 Interrupt Request) 7-31  
 TMR2-TMR0 (Timer Interrupt Request) 7-19, 7-30  
 TMR2-TMR1 (Timer 2/Timer 1 Interrupt In-Service) 7-32  
 TMR2-TMR1 (Timer 2/Timer 1 Interrupt Mask) 7-34  
 TRM2-TMR1 (Timer2/Timer1 Interrupt Request) 7-31  
 UB2-UB0 (Upper Boundary) 5-6  
 WD (Virtual Watchdog Timer Interrupt In-Service)

7-22

WD (Virtual Watchdog Timer Interrupt Mask) 7-24  
 WD (Virtual Watchdog Timer Interrupt Request) 7-20

BRK1 bit (Long Break Detected) 10-9

## C

C bit (Cascade Mode) 7-14  
 Cascade mode 7-11  
 CHG bit (Change Start Bit) 9-4  
 Clock Prescaler Register  
     description 6-2  
 CONT bit (Continuous Mode Bit)  
     Timer 0 Mode/Control Register 8-4  
     Timer 1 Mode/Control Register 8-4  
     Timer 2 Mode/Control Register 8-5

## D

D1-D0 field (DMA Channel Interrupt In-Service) 7-22, 7-32, 7-34  
 D1-D0 field (DMA Channel Interrupt Masks) 7-24  
 D1-D0 field (DMA Channel Interrupt Request) 7-31  
 DDA15-DDA0 field (DMA Destination Address Low) 9-8  
 DDA19-DDA16 field (DMA Destination Address High) 9-7  
 DDEC bit (Destination Decrement) 9-3  
 development tools  
     third-party products xiv  
 DHLT bit (DMA Halt) 7-19, 7-30  
 DINC bit (Destination Increment) 9-4  
 DM/IO bit (Destination Address Space Select) 9-3  
 DMA (DMA Control Field) 10-5  
 DMA 0 Control Register  
     description 9-3  
 DMA 0 Destination Address High Register  
     description 9-7  
 DMA 0 Destination Address Low Register  
     description 9-8  
 DMA 0 Interrupt Control Register  
     description  
         Master mode 7-17  
         Slave mode 7-29  
 DMA 0 Source Address High Register  
     description 9-9  
 DMA 0 Source Address Low Register  
     description 9-10

DMA 0 Transfer Count Register  
description 9-6

DMA 1 Control Register  
description 9-3

DMA 1 Destination Address High Register  
description 9-7

DMA 1 Destination Address Low Register  
description 9-8

DMA 1 Interrupt Control Register  
description  
Master mode 7-17  
Slave mode 7-29

DMA 1 Source Address High Register  
description 9-9

DMA 1 Source Address Low Register  
description 9-10

DMA 1 Transfer Count Register  
description 9-6

documentation  
AMD E86 Family publications xiv  
ordering documentation and literature v

DSA15-DSA0 field (DMA Source Address Low) 9-10

DSA19-DSA16 field (DMA Source Address High) 9-9

## E

E bit (Enable RCU) 6-2

EN bit (Enable Bit)  
Timer 0 Mode/Control Register 8-3  
Timer 1 Mode/Control Register 8-3  
Timer 2 Mode/Control Register 8-5

Enable RCU Register  
description 6-2

End-of-interrupt processing 7-12

End-of-Interrupt Register  
description  
Master mode 7-27

EOI 7-12

EX bit (Pin Selector) 5-11

EXT bit (External Clock Bit)  
Timer 0 Mode/Control Register 8-3  
Timer 1 Mode/Control Register 8-3

External interrupt acknowledge bus cycles table 7-8

## F

F2-F0 field (Clock Divisor Select) 4-8

FER bit (Framing Error Detected) 10-10

Figure  
external interrupt acknowledge bus cycles 7-8  
Fully nested mode interrupt controller  
connections 7-10

Fully nested mode 7-10

Fully nested mode interrupt controller connections 7-10

## H

HS0 bit (Handshake Signal 0) 10-10

## I

I4-I0 field (Interrupt In-Service) 7-22

I4-I0 field (Interrupt Mask) 7-24

I4-I0 field (Interrupt Requests) 7-20

IF (interrupt enable flag) 7-2

INH bit (Inhibit Bit)  
Timer 0 Mode/Control Register 8-3  
Timer 1 Mode/Control Register 8-3  
Timer 2 Mode/Control Register 8-5

In-Service Register  
description  
Master mode 7-22  
Slave mode 7-32

Instruction exceptions 7-3

INT bit (Interrupt Bit)  
Timer 0 Mode/Control Register 8-3  
Timer 1 Mode/Control Register 8-3  
Timer 2 Mode/Control Register 8-5

INT0 Control Register  
description  
Master mode 7-14

INT1 Control Register  
description  
Master mode 7-14

INT2 Control Register  
description  
Master mode 7-15

INT3 Control Register  
description  
Master mode 7-15

INT4 Control Register  
description  
Master mode 7-16

Interrupt acknowledge 7-8

Interrupt conditions and sequence 7-5

Interrupt control unit 7-1  
Interrupt controller registers  
    master mode 7-13  
    slave mode 7-28  
Interrupt controller reset conditions 7-9  
Interrupt enable flag (IF) 7-2  
Interrupt mask bit 7-2  
Interrupt Mask Register  
    description  
        Master mode 7-24  
        Slave mode 7-34  
Interrupt priority 7-3, 7-6  
Interrupt Request Register  
    description  
        Master mode 7-20  
        Slave mode 7-31  
Interrupt return (IRET) 7-5  
Interrupt Status Register  
    description  
        Master mode 7-19  
        Slave mode 7-30  
Interrupt type 7-1  
Interrupt types 7-7  
Interrupt types table 7-4  
Interrupt Vector Register  
    description  
        Slave mode 7-36  
Interrupt vector table 7-2  
Interrupts  
    array BOUNDS exception 7-7  
    breakpoint 7-7  
    cascade mode 7-11  
    divide error exception 7-7  
    EOI 7-12  
    ESC opcode exception 7-8  
    fully nested mode 7-10  
    Instruction exceptions 7-3  
    INTO overflow detected 7-7  
    Maskable and nonmaskable 7-2  
    master mode operation 7-10  
    nonmaskable (NMI) 7-7  
    polled 7-12  
    slave mode 7-28  
    slave mode nesting 7-28  
    Special fully nested mode 7-12  
    trace 7-7  
    unused opcode 7-7

IREQ bit (Interrupt Request)  
    Poll Register 7-26  
    Poll Status Register 7-25  
IRET interrupt return 7-5

## L

L2-L0 field (Interrupt Type) 7-35  
LB2-LB0 field (Lower Boundary) 5-4  
Low Memory Chip Select Register  
    description 5-6  
LTM bit (Level-Triggered Mode)  
    INT0 Control Register 7-14  
    INT1 Control Register 7-14  
    INT2 Control Register 7-15  
    INT3 Control Register 7-15  
    INT4 Control Register 7-16

## M

M/IO bit (Memory/I/O Space) 4-3  
M6-M0 field (MCS Block Size) 5-10  
M6-M0 field (Refresh Base) 6-1  
Maskable interrupts 7-2  
Master mode interrupt registers 7-13  
Master mode operation 7-10  
MC bit (Maximum Count Bit)  
    Timer 0 Mode/Control Register 8-3  
    Timer 1 Mode/Control Register 8-3  
    Timer 2 Mode/Control Register 8-5  
Memory Partition Register  
    description 6-1  
Midrange Memory Chip Select Register  
    description 5-8  
MS bit (Memory/I/O Space Selector) 5-11  
MSK (interrupt mask bit) 7-2  
MSK bit (Interrupt Mask)  
    DMA Interrupt Control Registers 7-17  
    Timer Interrupt Control Registers 7-17  
MSK bit (Mask)  
    DMA Interrupt Control Registers 7-29  
    INT0 Control Register 7-14  
    INT1 Control Register 7-14  
    INT2 Control Register 7-15  
    INT3 Control Register 7-15  
    INT4 Control Register 7-16  
    Serial Port Interrupt Control Register 7-18

**N**

Nonmaskable interrupts 7-2, 7-7  
 NSPEC bit (NonSpecific EOI) 7-27

**O**

OER bit (Overrun Error Detected) 10-10

**P**

P bit (Prescaler Bit)  
     Timer 0 Mode/Control Register 8-3  
     Timer 1 Mode/Control Register 8-3  
 P bit (Relative Priority) 9-4  
 PCS and MCS Auxiliary Register  
     description 5-10  
 PDATA15-PDATA0 field (PIO Data Blts) 11-5  
 PDATA31-PDATA16 field (PIO Data Blts) 11-5  
 PDIR15-PDIR0 field (PIO Direction Bits) 11-4  
 PDIR31-PDIR16 field (PIO Direction Bits) 11-4  
 PER bit (Parity Error Detected) 10-10  
 Peripheral Chip Select Register  
     description 5-12  
 Peripheral Control Block Relocation Register 4-3  
 physical dimensions xiv  
 pin description xiv  
 PIO Data 0 Register  
     description 11-5  
 PIO Data 1 Register  
     description 11-5  
 PIO Direction 0 Register  
     description 11-4  
 PIO Direction 1 Register  
     description 11-4  
 PIO Mode 0 Register  
     description 11-3  
 PIO Mode 1 Register  
     description 11-3  
 PMODE15-PMODE0 field (PIO Mode Bits) 11-3  
 PMODE31-PMODE16 field (PIO Mode Bits) 11-3  
 Poll Register  
     description  
         Master mode 7-26  
 Poll Status Register

description

    Master mode 7-25

Polled interrupts 7-12

PR2-PR0 field (Priority Level)

    DMA Interrupt Control Register 7-29

    Timer Interrupt Control Register 7-29

PR2-PR0 field (Priority)

    DMA Interrupt Control Registers 7-17

    INT0 Control Register 7-14

    INT1 Control Register 7-14

    INT2 Control Register 7-15

    INT3 Control Register 7-15

    INT4 Control Register 7-16

    Serial Port Interrupt Control Register 7-18

    Timer Interrupt Control Registers 7-17

Priority Mask Register

    description

        Master mode 7-23

        Slave mode 7-33

PRM2-PRM0 field (Priority Field Mask) 7-23, 7-33

Processor Release Level Register

    description 4-5

product support

    bulletin board service v

    documentation and literature v

    technical support hotline v

PSE bit (PSRAM Mode Enable) 5-7

**R**

R19-R8 field (Relocation Address Bits) 4-3

R1-R0 field (Wait State Value)

    Low Memory Chip Select Register 5-7

    Midrange Memory Chip Select Register 5-9

    PCS and MCS Auxiliary Register 5-11

    Upper Memory Chip Select Register 5-5

R2 bit (Ready Mode)

    Low Memory Chip Select Register 5-7

    Midrange Memory Chip Select Register 5-9

    PCS and MCS Auxiliary Register 5-11

    Upper Memory Chip Select Register 5-5

R7 field (Address Disable)

    Upper Memory Chip Select Register 5-5, 5-7

RB8 bit (Received Bit 8) 10-9

RC field (Reset Configuration) 4-4

RC8-RC0 field (Refresh Counter Reload Value) 6-2

RDATA field (Receive Data) 10-12

## RDR bit (Receive Data Ready) 10-9

### registers

Clock Prescaler (CDRAM, Offset E2h) 6-2  
 DMA 0 Control (D0CON, Offset CAh) 9-3  
 DMA 0 Destination Address High (D0DSTH, Offset C6h) 9-7  
 DMA 0 Destination Address Low (D0DSTL, Offset C4h) 9-8  
 DMA 0 Interrupt Control (DMA0CON, Offset 34h) 7-17, 7-29  
 DMA 0 Source Address High (D0SRCH, Offset C2h) 9-9  
 DMA 0 Source Address Low (D0SRCL, Offset C0h) 9-10  
 DMA 0 Transfer Count (D0TC, Offset C8h) 9-6  
 DMA 1 Control (D1CON, Offset DAh) 9-3  
 DMA 1 Destination Address High (D1DSTH, Offset D6h) 9-7  
 DMA 1 Destination Address Low (D1DSTL, Offset D4h) 9-8  
 DMA 1 Interrupt Control (DMA1CON, Offset 36h) 7-17, 7-29  
 DMA 1 Source Address High (D1SRCH, Offset D2h) 9-9  
 DMA 1 Source Address Low (D1SRCL, Offset D0h) 9-10  
 DMA 1 Transfer Count (D1TC, Offset D8h) 9-6  
 Enable RCU (EDRAM, Offset E4h) 6-2  
 End-of-Interrupt (EOI, Offset 22h) 7-27  
 In-Service (INSERTV, Offset 2Ch) 7-22, 7-32  
 INT0 Control (INT0, Offset 38h)  
     Master mode 7-14  
 INT1 Control (INT1, Offset 3Ah)  
     Master mode 7-14  
 INT2 Control (INT2, Offset 3Ch)  
     Master mode 7-15  
 INT3 Control (INT3, Offset 3Eh)  
     Master mode 7-15  
 INT4 Control (INT4, Offset 40h)  
     Master mode 7-16  
 Interrupt Mask (IMASK, Offset 28h) 7-24, 7-34  
 Interrupt Request (REQST, Offset 2Eh) 7-20, 7-31  
 Interrupt Status (INSTS, Offset 30h) 7-19  
 Interrupt Status (INTSTS, Offset 30h) 7-30  
 Interrupt Vector (INTVEC, Offset 20h) 7-36  
 Low Memory Chip Select (LMCS, Offset A2h) 5-6  
 Memory Partition (MDRAM, Offset E0h) 6-1  
 Midrange Memory Chip Select (MMCS, Offset A6h) 5-8

PCS and MCS Auxiliary (MPCS, Offset A8h) 5-10  
 Peripheral Chip Select (PACS, Offset A4h) 5-12  
 Peripheral Control Block Relocation (RELREG, Offset FEh) 4-3  
 PIO Data 0 (PDATA0, Offset 74h) 11-5  
 PIO Data 1 (PDATA1, Offset 7Ah) 11-5  
 PIO Direction 0 (PDIR0, Offset 72h) 11-4  
 PIO Direction 1 (PDIR1, Offset 78h) 11-4  
 PIO Mode 0 (PIOMODE0, Offset 70h) 11-3  
 PIO Mode 1 (PIOMODE1, Offset 76h) 11-3  
 Poll (POLL, Offset 24h) 7-26  
 Poll Status (POLLST, Offset 26h) 7-25  
 Priority Mask (PRIMSK, Offset 2Ah) 7-23, 7-33  
 Processor Release Level (PRL, Offset F4h) 4-5  
 Reset Configuration (RESCON, Offset F6h) 4-4  
 Serial Port 0 Control (SP0CT, Offset 80h) 10-5  
 Serial Port 0 Status (SP0STS, Offset 82h) 10-9  
 Serial Port 1 Control (SP1CT, Offset 10h) 10-5  
 Serial Port 1 Status (SP1STS, Offset 12h) 10-9  
 Serial Port Baud Rate Divisor (SPBAUD, Offset 88h) 10-13  
 Serial Port Interrupt Control (SPICON, Offset 44h)  
     Master mode 7-18  
 Serial Port Receive Data (SPRD, Offset 86h) 10-12  
 Serial Port Transmit (SPTD, Offset 84h) 10-11  
 Specific End-of-Interrupt (EOI, Offset 22h) 7-35  
 System Configuration Register (SYSCON, Offset F0h) 4-7  
 Timer 0 Count (T0CNT, Offset 50h) 8-6  
 Timer 0 Interrupt Control (T0INTCON, Offset 32h) 7-29  
 Timer 0 Maxcount Compare A (T0CMPA, Offset 52h) 8-7  
 Timer 0 Maxcount Compare B (T0CMPB, Offset 54h) 8-7  
 Timer 0 Mode and Control (T0CON, Offset 56h) 8-3  
 Timer 1 Count (T1CNT, Offset 58h) 8-6  
 Timer 1 Interrupt Control (T1INTCON, Offset 38h) 7-29  
 Timer 1 Maxcount Compare A (T1CMPA, Offset 5Ah) 8-7  
 Timer 1 Maxcount Compare B (T1CMPB, Offset 5Ch) 8-7  
 Timer 1 Mode and Control (T1CON, Offset 5Eh) 8-3  
 Timer 2 Count (T2CNT, Offset 60h) 8-6  
 Timer 2 Interrupt Control (T2INTCON, Offset 3Ah) 7-29

Timer 2 Maxcount Compare A (T2CMPA, Offset 62h) 8-7  
 Timer 2 Mode and Control (T2CON, Offset 66h) 8-5  
 Timer Interrupt Control (TCUCON, Offset 32h) 7-17  
 Upper Memory Chip Select (UMCS, Offset A0h) 5-4

## Reset

interrupt controller conditions 7-9

Reset Configuration Register  
 description 4-4

RIU bit (Register in Use)

Timer 0 Mode/Control Register 8-3

Timer 1 Mode/Control Register 8-3

RTG bit (Retrigger Bit)

Timer 0 Mode/Control Register 8-3

Timer 1 Mode/Control Register 8-3

## S

S/M bit (Slave/Master) 4-3

S4-S0 field (Poll Status)

Poll Register 7-26

Poll Status Register 7-25

S4-S0 field (Source Vector Type) 7-27

SDEC bit (Source Decrement) 9-4

Serial Port 0 Control Register  
 description 10-5

Serial Port 0 Status Register  
 description 10-9

Serial Port 1 Control Register  
 description 10-5

Serial Port 1 Status Register  
 description 10-9

Serial Port Baud Rate Divisor Register  
 description 10-13

Serial Port Interrupt Control Register  
 description

Master mode 7-18

Serial Port Receive Data Register  
 description 10-12

Serial Port Transmit Data Register  
 description 10-11

SFNM bit (Special Fully Nested Mode) 7-14

SINC bit (Source Increment) 9-4

Slave mode interrupts 7-28

Slave mode nesting 7-28

SM/IO bit (Source Address Space Select) 9-4

Software interrupt 7-3

Special fully nested mode 7-12  
 Specific End-of-Interrupt Register  
 description

Slave mode 7-35

SPI bit (Serial Port Interrupt In-Service) 7-22

SPI bit (Serial Port Interrupt Mask) 7-24

SPI bit (Serial Port Interrupt Request) 7-20

ST bit (Start/Stop DMA Channel) 9-5

SYN1-SYN0 field (Synchronization Type) 9-4

System Configuration Register  
 description 4-7

## T

T4-T0 field (Interrupt Type) 7-36

T8-T0 field (Refresh Count) 6-2

Table

interrupt controller registers in master mode 7-13

interrupt controller registers in slave mode 7-28

Interrupt types 7-4

TC bit (Terminal Count) 9-4

TC15-TC0 field (Timer Compare Value) 8-7

TC15-TC0 field (Timer Count Register) 9-6

TC15-TC0 field (Timer Count Value) 8-6

TDATA field (Transmit Data) 10-11

TDRQ bit (Timer Enable/Disable Request) 9-4

TEMT bit (Transmitter Empty) 10-10

thermal characteristics xiv

THRE bit (Transmit Holding Register Empty) 10-9

Timer 0 Count Register  
 description 8-6

Timer 0 Interrupt Control Register  
 description

Slave mode 7-29

Timer 0 Maxcount Compare A Register  
 description 8-7

Timer 0 Maxcount Compare B Register  
 description 8-7

Timer 0 Mode and Control Register  
 description 8-3

Timer 1 Count Register  
 description 8-6

Timer 1 Interrupt Control Register  
 description

Slave mode 7-29

Timer 1 Maxcount Compare A Register  
 description 8-7

Timer 1 Maxcount Compare B Register

description 8-7

Timer 1 Mode and Control Register

description 8-3

Timer 2 Count Register

description 8-6

Timer 2 Interrupt Control Register

description

Slave mode 7-29

Timer 2 Maxcount Compare B Register

description 8-7

Timer 2 Mode and Control Register

description 8-5

Timer Interrupt Control Register

description

Master mode 7-17

timing characteristics xiv

TMR bit (Timer Interrupt In-Service) 7-22

TMR bit (Timer Interrupt Mask) 7-24

TMR bit (Timer Interrupt Request) 7-21

TMR0 bit (Timer 0 Interrupt In-Service) 7-32

TMR0 bit (Timer 0 Interrupt Mask) 7-34

TMR0 bit (Timer 0 Interrupt Request) 7-31

TMR2-TMR0 field (Timer Interrupt Request) 7-19, 7-30

TMR2-TMR1 field (Timer 2/Timer 1 Interrupt  
In-Service) 7-32

TMR2-TMR1 field (Timer 2/Timer 1 Interrupt  
Mask) 7-34

trace interrupt 7-7

TRM2-TMR1 field (Timer2/Timer1 Interrupt  
Request) 7-31

## **U**

UB2-UB0 field (Upper Boundary) 5-6

Upper Memory Chip Select Register

description 5-4

## **W**

WD bit (Virtual Watchdog Timer Interrupt  
In-Service) 7-22

WD bit (Virtual Watchdog Timer Interrupt Mask) 7-24

WD bit (Virtual Watchdog Timer Interrupt Request) 7-20