Am186™CC/CH/CU Microcontrollers

Register Set Manual

Order #21916B

AMD

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.

Am186, CodeKit, Comm86, E86, and SmartDMA 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 and may be trademarks of their respective companies.

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

The AMD customer service network includes U.S. offices, international offices, and a customer training center. Expert technical assistance is available from the AMD worldwide staff of field application engineers and factory support staff to answer E86™ and Comm86™ family hardware and software development questions.

Frequently accessed numbers are listed below. Additional contact information is listed on the back of this manual. AMD's WWW site lists the latest phone numbers.

Technical Support

Answers to technical questions are available online, through e-mail, and by telephone.

Go to AMD's home page at **www.amd.com** and follow the **Service** link for the latest AMD technical support phone numbers, software, and Frequently Asked Questions.

For technical support questions on all E86 and Comm86 products, send e-mail to **epd.support@amd.com** (in the US and Canada) or **euro.tech@amd.com** (in Europe and the UK).

You can also call the AMD Corporate Applications Hotline at:

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

WWW Support

For specific information on E86 and Comm86 products, access the AMD home page at **www.amd.com** and follow the **Embedded Processors** link. These pages provide information on upcoming product releases, overviews of existing products, information on product support and tools, and a list of technical documentation. Support tools include online benchmarking tools and CodeKit™ software—tested source code example applications. Many of the technical documents are available online in PDF form.

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

Documentation and Literature Support

Data books, user's manuals, data sheets, application notes, and product CDs are free with a simple phone call. Internationally, contact your local AMD sales office for product literature.

To order literature, call:

(800) 222-9323	Toll-free for U.S. and Canada
(512) 602-5651	Direct dial worldwide
(512) 602-7639	Fax

Third-Party Support

AMD FusionE86^{SM} partners provide an array of products designed to meet critical time-to-market needs. Products and solutions available include emulators, hardware and software debuggers, board-level products, and software development tools, among others. The WWW site and the *E86 Family Products Development Tools CD, order# 21058*, describe these solutions. In addition, mature development tools and applications for the x86 platform are widely available in the general marketplace.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

INTRODUCTION

Am186™CC MICROCONTROLLER

The Am186™CC communications controller is the first member in the AMD Comm86™ product family. The Am186CC microcontroller is a cost-effective, high-performance microcontroller solution for communications applications. This highly integrated microcontroller enables customers to save system costs and increase performance over 8-bit microcontrollers and other 16-bit microcontrollers.

The Am186CC microcontroller offers the advantages of the x86 development environment's widely available native development tools, applications, and system software. Additionally, the microcontroller uses the industry-standard 186 instruction set that is part of the AMD E86 family, which continually offers instruction-set-compatible upgrades. Built into the Am186CC microcontroller is a wide range of communications features required in many communications applications, including High-level Data Link Control (HDLC) and the Universal Serial Bus (USB). See Chapter 1, "Peripheral Control Block Register Overview", for a list of included features.

AMD offers complete solutions with the Am186CC microcontroller. A customer development platform board is available. Reference designs that are available or being developed include an Integrated Services Digital Network (ISDN) Terminal Adapter featuring USB, and a low-end router with ISDN, Ethernet, USB, and plain-old telephone service (POTS). AMD and its FusionE86℠ Partners offer boards, schematics, drivers, protocol stacks, and routing software for these reference designs, enabling fast time to market.

Am186CH HDLC MICROCONTROLLER

The Am186CH HDLC microcontroller is a cost-reduced derivative of the Am186CC microcontroller that is targeted towards cost-sensitive applications such as linecards and digital phones. The Am186CH HDLC microcontroller is pin-compatible with the Am186CC microcontroller and offers many of the same features, yet the Am186CH HDLC microcontroller provides a cost-effective solution for communications devices that require fewer HDLC channels and do not need GCI or USB.

Am186CU USB MICROCONTROLLER

The Am186CU USB microcontroller is a cost-reduced derivative of the Am186CC microcontroller that is targeted towards cost-sensitive applications such as USB peripherals and digital-subscriber-line (DSL) modems. The Am186CU USB microcontroller is pin-compatible with the Am186CC microcontroller and offers many of the same features, yet the Am186CU USB microcontroller provides a cost-effective solution for USB devices that do not need GCI or HDLC.

PURPOSE OF THIS MANUAL

This manual includes in reference format the complete set of peripheral control registers required to program the Am186CC/CH/CU microcontrollers. For information about using the internal processor registers, see the *Am186 and Am188 Family Instruction Set Manual*, order #21267.

Intended Audience

This reference manual is intended primarily for programmers who are developing code for the Am186CC/CH/CU microcontrollers. Computer software and hardware architects and system engineers who are designing or are considering designing systems based on these controllers may also be interested in the information contained in this document. For more information on programming these controllers, see the *Am186CC/CH/CU Microcontrollers User's Manual*, order #21914.

Overview of This Manual

This manual is organized into the following chapters.

- Chapter 1 contains an overview of the microcontroller's **Peripheral Control Block (PCB) registers**.
- Chapter 2 includes descriptions for all of the **High-level Data Link Control (HDLC) registers** (Am186CC and Am186CH microcontrollers only).
- Chapter 3 includes descriptions for the **Direct Memory Access (DMA) registers**.
- Chapter 4 includes descriptions for the **Universal Serial Bus (USB) registers** (Am186CC and Am186CU microcontrollers only).
- Chapter 5 includes descriptions for the **Asynchronous Serial Port (UART) registers**.
- Chapter 6 includes descriptions for the **General Circuit Interface (GCI) registers** (Am186CC microcontroller only).
- Chapter 7 includes descriptions for the **Time Slot Assigner (TSA) registers** (Am186CC and Am186CH microcontrollers only).
- Chapter 8 includes descriptions for the **Synchronous Serial Interface (SSI) registers**.
- Chapter 9 includes descriptions for the **Interrupt Controller registers**.
- Chapter 10 includes descriptions for the **Timer registers**.
- Chapter 11 includes descriptions for the **Chip Select registers**.
- Chapter 12 includes descriptions for the **Dynamic Random-Access Memory (DRAM) registers**.
- Chapter 13 includes descriptions for the **Programmable I/O (PIO) registers**.
- Chapter 14 describes the **Reset Configuration register**.
- Chapter 15 describes the **Watchdog Timer register**.
- Chapter 16 includes descriptions for the **Miscellaneous Configuration registers**.

Within each chapter, the registers are listed in ascending hexadecimal order unless descriptions for identical register sets (for example, the HDLC registers) can be combined.

RELATED DOCUMENTS

AMD Documentation

The following AMD documents provide additional information about the Am186CC microcontroller.

- The *Am186™CC Communications Controller Data Sheet*, order #21915, the *Am186™CH HDLC Microcontroller Data Sheet*, order #22024, and the *Am186™CU USB Microcontroller Data Sheet*, order #22025, include complete pin lists, pin state tables, timing and thermal characteristics, and package dimensions for the Am186CC/CH/CU microcontrollers.
- The *Am186CC/CH/CU Microcontrollers User's Manual*, order #21914, provides a functional description of the microcontrollers for both hardware and software designers.
- The *Am186 and Am188 Family Instruction Set Manual*, order #21267, includes the Am186 microprocessor instruction set used by the Am186CC/CH/CU microcontrollers.
- The *E86™ Family Products and Development Tools CD*, order #21058, contains product documentation, utilities, third-party development tools, and evaluation software for AMD microcontrollers and microprocessors.

DOCUMENTATION CONVENTIONS

Table 0-1 lists the documentation conventions used throughout this manual.

Table 0-1 Documentation Conventions

Notation	Meaning
General	
bit	A single bit in a register.
bit field	Two or more consecutive and related bits.
set the EN bit	Write a 1 to the EN bit.
clear the EN bit	Write a 0 to the EN bit.
external reset	A reset caused by asserting the $\overline{\text{RES}}$ input signal.
internal reset	A reset initiated by the watchdog timer (see page 15-2).
system reset	Assertion of the RESOUT signal to reset external peripherals.
offset 000h	A register offset, relative to the base of the current PCB space defined in the Relocation (RELOC) register (see page 16-5).
Register Descriptions	
Software Read/Write	This row indicates whether software running on the microcontroller can change the value of a field.
R in software read/write row	Software can read this field; for reserved fields, always write the chip reset default value; otherwise, writing this field has no effect.
W in software read/write row	Software can only write this field; when read, the field has no meaning.
R/W in software read/write row	Software can read or write this field.
R/W0 in software read/write row	Software can read or clear bits in this field.
R/W1 in software read/write row	Software can read or set bits in this field.

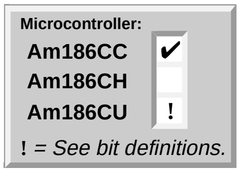
Table 0-1 Documentation Conventions (Continued)

Notation	Meaning
Hardware Set/Clear	This row indicates whether microcontroller hardware, such as a timer or interrupt, can change the value of a field.
— in hardware set/clear row	Hardware cannot change this field’s value after reset.
S in hardware set/clear row	Hardware can set bits in this field.
C in hardware set/clear row	Hardware can clear bits in this field.
S/C in hardware set/clear row	Hardware can set or clear bits in this field.
Chip Reset Default	Value upon an external or internal reset (except in the Watchdog Timer Control (WDTCON) register; see page 15-4).
0 in default register value	Cleared on chip reset.
1 in default register value	Set on chip reset.
x in default register value	Non-deterministic or floating; no value is guaranteed.
? in default register value	Determined by sources external to the microcontroller.
Pin Naming	
{ }	Pin function during hardware reset (pinstrap).
[ ]	Alternative pin function.
$\overline{\text{MCS3}}$–$\overline{\text{MCS0}}$	All four signals (or registers, or fields).
$\overline{\text{MCSx}}$	Any of the four signals (or registers, or fields).
Numbers	
b	Binary number.
d	Decimal number. Decimal is the default radix.
h	Hexadecimal number.
x in a number	Any of several values is legal; for example, 0x01b can be either 0001b or 0101b.
[X–Y]	The bit field that consists of bits X through Y.

Microcontroller-Specific Information

Throughout this manual, information is provided about differences between the Am186CC/CH/CU microcontrollers. A quick summary of differences is provided for each register description in a small diagram as shown in the following example.

Figure 0-1 Example Difference Diagram



Microcontroller:	
Am186CC	✓
Am186CH	
Am186CU	!

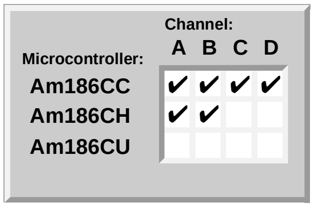
! *= See bit definitions.*

The box next to each microcontroller name indicates the extent to which that microcontroller supports register functions described in the accompanying text. The symbols in the diagram have the following meanings:

- **Check mark (✓):** the register's functions are fully supported as described.
- **Empty box:** the register described is not supported in this microcontroller.
- **Exclamation point (!):** if this symbol is present, one or more of the register's bit definitions include microcontroller-specific notes describing any differences that apply. Except for the noted differences, the register's functions are supported as described.

In some chapters, a single register description can apply to several different registers. For example, each register description in Chapter 2, "HDLC Registers (Am186CC and Am186CH Microcontrollers Only)", can apply to as many as four different HDLC channels, depending on the microcontroller. In this situation the difference diagram is expanded, as shown in Figure 0-2, to include information about which channels can be used with each microcontroller.

Figure 0-2 Example Difference Diagram for Multichannel Registers



	Channel:			
Microcontroller:	A	B	C	D
Am186CC	✓	✓	✓	✓
Am186CH	✓	✓		
Am186CU				

Note: *Registers that are not supported on a particular microcontroller must be left in their default states unless otherwise noted in the bit field descriptions. Changing the state of an unsupported register can have unpredictable effects.*

CHAPTER

1 PERIPHERAL CONTROL BLOCK REGISTER OVERVIEW

The Am186CC/CH/CU microcontrollers' internal peripherals are controlled by 16-bit read/write registers. The peripheral registers are contained within an internal 1-Kbyte control block called the peripheral control block (PCB). Registers are physically located in the peripheral devices they control, but they are addressed as a single 1-Kbyte block.

The configuration registers on the Am186CC microcontroller control the following peripherals:

- Four High-level Data Link Control (HDLC) channels
- Four general-purpose DMA and eight SmartDMA™ channels
- Six Universal Serial Bus (USB) endpoints (control, interrupt, and A–D)
- Asynchronous and high-speed asynchronous serial ports
- General Circuit Interface (GCI)
- Four Time Slot Assigners (TSAs)
- Synchronous Serial Interface (SSI)
- 15 interrupt controller channels (36 interrupt sources)
- Three timers
- Four programmable memory and peripheral chip selects, with DRAM control
- 48 programmable input/output pins
- Reset configuration inputs for design-specific jumper and switch settings
- Watchdog timer

Am186CH Microcontroller: *The Am186CH HDLC microcontroller peripherals are similar to the Am186CC microcontroller, except the Am186CH HDLC microcontroller supports only two HDLC and TSA channels and it has no GCI or USB support. In addition, because it supports fewer internal peripherals, the Am186CH HDLC microcontroller uses only four SmartDMA channels and 32 interrupt sources.*

Am186CU Microcontroller: *The Am186CU USB microcontroller peripherals are similar to the Am186CC microcontroller, except the Am186CU USB microcontroller does not support HDLC, TSA, or GCI. In addition, because it supports fewer internal peripherals, the Am186CU USB microcontroller uses only four SmartDMA channels and 31 interrupt sources.*

A System Configuration register is provided to select microcontroller modes of operation, and a read-only Processor Revision Level register provides information about the specific processor level being used.

The peripheral control block can be mapped into either memory or I/O space. The base address of the PCB must be on an even 1024-byte boundary (i.e., the lower ten bits of the base address are all 0b). Internal logic recognizes control block addresses and responds to bus cycles. During bus cycles to internal registers, the bus controller signals the operation externally (i.e., the $\overline{\text{RD}}$, $\overline{\text{WR}}$, status, address, and data lines are driven as in a normal bus cycle), but the data bus, SRDY, and ARDY are ignored.

On an external or watchdog timer reset, the Relocation (RELOC) register value is set to 20FCh, which maps the PCB to start at FC00h in I/O space. This places the RELOC register at FFFEh. Appendix A provides a summary of PCB registers in offset order, including default address locations. See Chapter 16, “Miscellaneous Configuration Registers” for a complete description of the Relocation register.

Table 1-1 shows peripheral control block ranges that are used for each functional block. Unused addresses within this block are reserved. The entire peripheral control block can be relocated in memory or I/O space by using the Relocation register.

Table 1-1 Peripheral Control Block Address Map Summary

Function	I/O Offset Range
High-level Data Link Control (HDLC)	000h–0F0h[1]
General-Purpose DMA	100h–13Ch
SmartDMA	140h–198h
Universal Serial Bus (USB)	1E0h–25Eh[2]
High-Speed Asynchronous Serial Port	260h–27Ch
Asynchronous Serial Port	280h–28Eh
General Circuit Interface (GCI)	2A0h–2BEh[3]
Time Slot Assigner (TSA)	2C0h–2DCh[1]
Synchronous Serial Interface (SSI)	2F0h–2F8h
Interrupt Controller	300h–338h
Programmable Timers	340h–354h
Chip Select	3A0h–3A8h
DRAM Refresh	3AAh–3ACh
Programmable I/O (PIO)	3C0h–3DCh
Reset Configuration	3DEh
Watchdog Timer	3E0h
System Configuration	3F0h
Processor Release Level	3F4h
Relocation	3FEh

Notes:

1. Reserved in the Am186CU USB microcontroller.

2. Reserved in the Am186CH HDLC microcontroller.

3. Reserved in the Am186CH and Am186CU microcontrollers.

CHAPTER

2 HDLC REGISTERS (Am186CC and Am186CH Microcontrollers Only)

2.1 OVERVIEW

This chapter describes the High-level Data Link Control (HDLC) registers on the Am186CC and Am186CH microcontrollers. *The Am186CU USB microcontroller does not support HDLC.*

The Am186CC microcontroller provides four HDLC channels. The Am186CH HDLC microcontroller provides two HDLC channels. The HDLCs are typically used to transmit and receive frames based on the HDLC or Synchronous Data Link Control (SDLC) formats. These formats use flags to determine the start and stop of a frame, and they use “bit stuffing” to maintain data transparency.

In the transmit direction (data leaving the device), the HDLCs add the required error detection bytes (cyclic redundancy check, or CRC) at the end of the frame, bit stuff the data as needed, and surround it with flags.

In the receive direction (data coming into the device), the HDLCs search for flags to determine the start and stop of the frame, remove any bit stuffing, and check the CRC bytes. The HDLCs also check the addresses of incoming frames and reject frames with the wrong address.

First-in-first-out buffers (FIFOs) are used in both directions to isolate data requests from the execution unit. The FIFOs can be filled or emptied using the SmartDMA controller (see Chapter 3, “DMA Registers”) or using programmed I/O.

Each of the four HDLCs is used with a Time Slot Assigner (TSA). See Chapter 7, “TSA Registers (Am186CC and Am186CH Microcontrollers Only)”.

For more information about using HDLCs and TSAs, see the *Am186CC/CH/CU Microcontrollers User’s Manual*, order #21914.

The HDLC register sets are identical for each of the channels. This chapter describes registers with identical functions only once. The unique register names and offsets that apply are listed at the top of each register page. Table 2-1 lists the HDLC registers in offset order, with the corresponding description’s page number.

Table 2-1 HDLC Register Map

Register Name	Mnemonic	Offset	Page Number
HDLC Channel A Registers (Am186CC and Am186CH HDLC Microcontrollers Only)			
HDLC Channel A Control	HACON	00h	page 2-5
HDLC Channel A Transmit Control 0	HATCON0	02h	page 2-7
HDLC Channel A Transmit Control 1	HATCON1	04h	page 2-9
HDLC Channel A Receive Control 0	HARCON0	06h	page 2-11
HDLC Channel A Receive Max Length 1	HARCON1	08h	page 2-13
HDLC Channel A Status	HASTATE	0Ah	page 2-14
HDLC Channel A Interrupt Status 0	HAISTAT0	0Ch	page 2-16
HDLC Channel A Interrupt Mask 0	HAIMSK0	0Eh	page 2-18

Table 2-1 HDLC Register Map (Continued)

Register Name	Mnemonic	Offset	Page Number
HDLC Channel A Interrupt Status 1	HAISTAT1	10h	page 2-20
HDLC Channel A Interrupt Mask 1	HAIMSK1	12h	page 2-22
HDLC Channel A Transmit FIFO Data	HATD	14h	page 2-24
HDLC Channel A Receive FIFO Data	HARD	16h	page 2-25
HDLC Channel A Receive FIFO Data Peek	HARDP	18h	page 2-27
HDLC Channel A Receive Frame Status 1	HARFS1	16h	page 2-29
HDLC Channel A Receive Frame Status 2	HARFS2	16h	page 2-31
HDLC Channel A Receive Frame Status 3	HARFS3	16h	page 2-33
HDLC Channel A Short Frame Counter	HASFCNT	1Ah	page 2-35
HDLC Channel A Short Frame Counter Peek	HASFCNTP	1Ch	page 2-36
HDLC Channel A Mismatch Address Counter	HAMACNT	1Eh	page 2-37
HDLC Channel A Mismatch Address Counter Peek	HAMACNTP	20h	page 2-38
HDLC Channel A Address 0	HAA0	22h	page 2-39
HDLC Channel A Address 1	HAA1	26h	page 2-39
HDLC Channel A Address 2	HAA2	2Ah	page 2-39
HDLC Channel A Address 3	HAA3	2Eh	page 2-39
HDLC Channel A Address Mask 0	HAA0MSK	24h	page 2-41
HDLC Channel A Address Mask 1	HAA1MSK	28h	page 2-41
HDLC Channel A Address Mask 2	HAA2MSK	2Ch	page 2-41
HDLC Channel A Address Mask 3	HAA3MSK	30h	page 2-41
HDLC Channel B Registers (Am186CC and Am186CH HDLC Microcontrollers Only)			
HDLC Channel B Control	HBCON	40h	page 2-5
HDLC Channel B Transmit Control 0	HBTCON0	42h	page 2-7
HDLC Channel B Transmit Control 1	HBTCON1	44h	page 2-9
HDLC Channel B Receive Control 0	HBRCON0	46h	page 2-11
HDLC Channel B Receive Max Length 1	HBRCON1	48h	page 2-13
HDLC Channel B Status	HBSTATE	4Ah	page 2-14
HDLC Channel B Interrupt Status 0	HBISTAT0	4Ch	page 2-16
HDLC Channel B Interrupt Mask 0	HBIMSK0	4Eh	page 2-18
HDLC Channel B Interrupt Status 1	HBISTAT1	50h	page 2-20
HDLC Channel B Interrupt Mask 1	HBIMSK1	52h	page 2-22
HDLC Channel B Transmit FIFO Data	HBTD	54h	page 2-24
HDLC Channel B Receive FIFO Data	HBRD	56h	page 2-25
HDLC Channel B Receive FIFO Data Peek	HBRDP	58h	page 2-27
HDLC Channel B Receive Frame Status 1	HBRFS1	56h	page 2-29
HDLC Channel B Receive Frame Status 2	HBRFS2	56h	page 2-31
HDLC Channel B Receive Frame Status 3	HBRFS3	56h	page 2-33
HDLC Channel B Short Frame Counter	HBSFCNT	5Ah	page 2-35
HDLC Channel B Short Frame Counter Peek	HBSFCNTP	5Ch	page 2-36
HDLC Channel B Mismatch Address Counter	HBMACNT	5Eh	page 2-37

Table 2-1 HDLC Register Map (Continued)

Register Name	Mnemonic	Offset	Page Number
HDLC Channel B Mismatch Address Counter Peek	HBMACNTP	60h	page 2-38
HDLC Channel B Address 0	HBA0	62h	page 2-39
HDLC Channel B Address 1	HBA1	66h	page 2-39
HDLC Channel B Address 2	HBA2	6Ah	page 2-39
HDLC Channel B Address 3	HBA3	6Eh	page 2-39
HDLC Channel B Address Mask 0	HBA0MSK	64h	page 2-41
HDLC Channel B Address Mask 1	HBA1MSK	68h	page 2-41
HDLC Channel B Address Mask 2	HBA2MSK	6Ch	page 2-41
HDLC Channel B Address Mask 3	HBA3MSK	70h	page 2-41
HDLC Channel C Registers (Am186CC Microcontroller Only)			
HDLC Channel C Control	HCCON	80h	page 2-5
HDLC Channel C Transmit Control 0	HCTCON0	82h	page 2-7
HDLC Channel C Transmit Control 1	HCTCON1	84h	page 2-9
HDLC Channel C Receive Control 0	HCRCON0	86h	page 2-11
HDLC Channel C Receive Max Length 1	HCRCON1	88h	page 2-13
HDLC Channel C Status	HCSTATE	8Ah	page 2-14
HDLC Channel C Interrupt Status 0	HCISTAT0	8Ch	page 2-16
HDLC Channel C Interrupt Mask 0	HCIMSK0	8Eh	page 2-18
HDLC Channel C Interrupt Status 1	HCISTAT1	90h	page 2-20
HDLC Channel C Interrupt Mask 1	HCIMSK1	92h	page 2-22
HDLC Channel C Transmit FIFO Data	HCTD	94h	page 2-24
HDLC Channel C Receive FIFO Data	HCRD	96h	page 2-25
HDLC Channel C Receive FIFO Data Peek	HCRDP	98h	page 2-27
HDLC Channel C Receive Frame Status 1	HCRFS1	96h	page 2-29
HDLC Channel C Receive Frame Status 2	HCRFS2	96h	page 2-31
HDLC Channel C Receive Frame Status 3	HCRFS3	96h	page 2-33
HDLC Channel C Short Frame Counter	HCSFCNT	9Ah	page 2-35
HDLC Channel C Short Frame Counter Peek	HCSFCNTP	9Ch	page 2-36
HDLC Channel C Mismatch Address Counter	HCMACNT	9Eh	page 2-37
HDLC Channel C Mismatch Address Counter Peek	HCMACNTP	A0h	page 2-38
HDLC Channel C Address 0	HCA0	A2h	page 2-39
HDLC Channel C Address 1	HCA1	A6h	page 2-39
HDLC Channel C Address 2	HCA2	AAh	page 2-39
HDLC Channel C Address 3	HCA3	AEh	page 2-39
HDLC Channel C Address Mask 0	HCA0MSK	A4h	page 2-41
HDLC Channel C Address Mask 1	HCA1MSK	A8h	page 2-41
HDLC Channel C Address Mask 2	HCA2MSK	ACh	page 2-41
HDLC Channel C Address Mask 3	HCA3MSK	B0h	page 2-41

Table 2-1 HDLC Register Map (Continued)

Register Name	Mnemonic	Offset	Page Number
HDLC Channel D Registers (Am186CC Microcontroller Only)			
HDLC Channel D Control	HDCON	C0h	page 2-5
HDLC Channel D Transmit Control 0	HDTCON0	C2h	page 2-7
HDLC Channel D Transmit Control 1	HDTCON1	C4h	page 2-9
HDLC Channel D Receive Control 0	HDRCON0	C6h	page 2-11
HDLC Channel D Receive Max Length 1	HDRCON1	C8h	page 2-13
HDLC Channel D Status	HDSTATE	CAh	page 2-14
HDLC Channel D Interrupt Status 0	HDISTAT0	CCh	page 2-16
HDLC Channel D Interrupt Mask 0	HDIMSK0	CEh	page 2-18
HDLC Channel D Interrupt Status 1	HDISTAT1	D0h	page 2-20
HDLC Channel D Interrupt Mask 1	HDIMSK1	D2h	page 2-22
HDLC Channel D Transmit FIFO Data	HDTD	D4h	page 2-24
HDLC Channel D Receive FIFO Data	HDRD	D6h	page 2-25
HDLC Channel D Receive FIFO Data Peek	HDRDP	D8h	page 2-27
HDLC Channel D Receive Frame Status 1	HDRFS1	D6h	page 2-29
HDLC Channel D Receive Frame Status 2	HDRFS2	D6h	page 2-31
HDLC Channel D Receive Frame Status 3	HDRFS3	D6h	page 2-33
HDLC Channel D Short Frame Counter	HDSFCNT	DAh	page 2-35
HDLC Channel D Short Frame Counter Peek	HDSFCNTP	DCh	page 2-36
HDLC Channel D Mismatch Address Counter	HDMACNT	DEh	page 2-37
HDLC Channel D Mismatch Address Counter Peek	HDMACNTP	E0h	page 2-38
HDLC Channel D Address 0	HDA0	E2h	page 2-39
HDLC Channel D Address 1	HDA1	E6h	page 2-39
HDLC Channel D Address 2	HDA2	EAh	page 2-39
HDLC Channel D Address 3	HDA3	EEh	page 2-39
HDLC Channel D Address Mask 0	HDA0MSK	E4h	page 2-41
HDLC Channel D Address Mask 1	HDA1MSK	E8h	page 2-41
HDLC Channel D Address Mask 2	HDA2MSK	ECh	page 2-41
HDLC Channel D Address Mask 3	HDA3MSK	F0h	page 2-41

HDLC Channel A Control (HACON) Offset 00h
HDLC Channel B Control (HBCON) Offset 40h
HDLC Channel C Control (HCCON) Offset 80h
HDLC Channel D Control (HDCON) Offset C0h

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	*Res*								HRESET	*Res*	NRZI	TRANSM	LOOPR	LOOPL	CRCTYPE	
Software Read/Write	R								R/W	R	R/W	R/W	R/W	R/W	R/W	
Hardware Set/Clear	—								—	—	—	—	—	—	—	
Chip Reset Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Microcontroller:	Channel: A	B	C	D
Am186CC	✓	✓	✓	✓
Am186CH	✓	✓		
Am186CU				

Register Description

This register sets operating modes for both the receiver and transmitter.

Bit Definitions

Bit	Name	Function
15–8	Res	**Reserved** For compatibility with future devices, always write this bit field with its chip reset default value.
7	HRESET	**HDLC Reset** 0 = Allows FIFOs and state machines to operate normally. 1 = Clears the transmit and receive FIFOs and asynchronously resets the transmitter and receiver state machines. Setting this bit does not clear the R/W0 interrupt bits in the HxISTAT0 and HxISTAT1 registers, but it does clear the HxSFCNT and HxMACNT registers. If the HDLC channel is used on a PCM highway in transparent mode, setting and then clearing this bit causes the HDLC channel to synchronize its byte alignment with the PCM highway. The first byte received or transmitted might be corrupted while the HDLC controller is performing the alignment. To mask this effect on the transmit side, clear the FLAGIDL bit in the HxTCON1 register (see page 2-9), and make the first transmitted byte all 1s (FFh).
6	Res	**Reserved** For compatibility with future devices, always write this bit field with its chip reset default value.
5	NRZI	**Non-Return to Zero Invert** 0 = Enables NRZ (normal) encoding. 1 = Enables NRZI encoding.
4	TRANSM	**Transparent Mode** 0 = Normal operation is selected. 1 = Transparent mode is selected. Transparent mode disables cyclic redundancy check (CRC) generation, flag generation/detection, abort detection, bit stuffing/unstuffing, and CTS lost abort generation. Before using Transparent mode, clear all of the channel’s Address Mask (HAMSK) bit fields (see page 2-41) and its Flag Idle (FLAGIDL) bit (see page 2-9). Do not change the TRANSM bit unless both the Transmitter Stop bit (see TSTOP on page 2-16) and Receiver Stop bit (see RSTOP on page 2-12) are 1.

Bit	Name	Function
3	LOOPR	**Loop Remote** 0 = Transmitter controls the transmitter output and uses the transmit clock (TCLK) signal. 1 = Receiver input is fed directly to the transmitter output. Both the transmit clock (TCLK) and the receive clock (RCLK) signals are required. Do not change the LOOPR bit unless the TSTOP bit is 1 (see page 2-16).
2	LOOPL	**Loop Local** 0 = Receiver input comes from the Time Slot Assigner (TSA) and the receive clock (RCLK) signal is used. 1 = Transmitter output is fed directly to the receiver input. Only the transmit clock (TCLK) signal is used. The LOOPL bit should not be changed unless the RSTOP bit is 1 (see page 2-12).
1–0	CRCTYPE	**CRC Type** The CRCTYPE bit selects the algorithm of CRC generation/checking. 00 = CRC-CCITT (16 bit) ($x^{16} + x^{12} + x^5 + 1$) 01 = CRC-16 (16 bit) ($x^{16} + x^{15} + x^2 + 1$) 10 = CRC-32 (32 bit) ($x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$) 11 = Reserved

Programming Notes

HDLC Channel A Transmit Control 0 (HATCON0) Offset 02h
HDLC Channel B Transmit Control 0 (HBTCON0) Offset 42h
HDLC Channel C Transmit Control 0 (HCTCON0) Offset 82h
HDLC Channel D Transmit Control 0 (HDTCON0) Offset C2h

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	*Res*				TTHRSH		*Res*			TFIFOEN	FORABR	HTEN	IMSTART	CRCDIS	LBREAD	LBNOW
Software Read/Write	R				R/W		R			R/W	R/W	R/W	R/W	R/W	R/W	W
Hardware Set/Clear	—				—		—			—	—	—	—	—	—	—
Chip Reset Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	Channel:			
Microcontroller:	**A**	**B**	**C**	**D**
Am186CC	✔	✔	✔	✔
Am186CH	✔	✔		
Am186CU				

Register Description

This register sets operating modes for the transmitter.

Bit Definitions

Bit	Name	Function
15–12	Res	**Reserved** For compatibility with future devices, always write this bit field with its chip reset default value.
11–10	TTHRSH	**Transmit Threshold** The TTHRSH bit is used to program the amount of free space desired in the transmit FIFO before generating an interrupt. This same size is used to determine when to request a DMA access. When TTHRSH is set to 16 bytes, the IMSTART bit must be set to 1 to ensure proper operation of the transmitter. 00 = 1 byte 01 = 9 bytes 10 = 16 bytes 11 = Reserved
9–7	Res	**Reserved** For compatibility with future devices, always write this bit field with its chip reset default value.
6	TFIFOEN	**Transmit FIFO Enable** 0 = Flushes the transmit FIFO of data; no requests are performed. The TFIFOEN bit should only be cleared when the TSTOP bit is 1 (see page 2-17). 1 = Enables the transmit FIFO; the FIFO can request data.
5	FORABR	**Force Abort** 0 = The transmitter operates normally. 1 = The transmitter sends abort sequences instead of data; the FIFO is flushed. The FORABR bit must be cleared before the transmitter will resume operation. An abort sequence is seven to fourteen 1s. If the FORABR bit is set and then immediately cleared, one abort sequence is transmitted. If the FORABR bit is set during the closing flag, the flag completes before the abort is sent. If the MLTDRP bit is 1 or the GCIDEN bit is 1 (see page 2-10), only one abort is sent and then the transmitter stops even if the FORABR bit is still 1.

Bit	Name	Function
4	HTEN	**Transmit Enable** 0 = Disables transmission at the end of the current frame. 1 = Enables transmission, which begins when the channel's clear-to-send ($\overline{\text{CTS}}$) signal is asserted and sufficient data is in the transmit FIFO.
3	IMSTART	**Immediate Start** 0 = The transmitter waits to start transmission until the FIFO is half full or the end of the frame is placed in the FIFO. 1 = Immediate start. The transmitter begins transmitting as soon as data is in the transmit FIFO. When the TTHRSH bit is programmed to 16 bytes (TTHRSH = 10b), the IMSTART bit must be set to 1 to ensure proper operation of the transmitter.
2	CRCDIS	**CRC Disable** 0 = The CRC is appended to the end of the frame. The receiver on the other end can interpret the last few bytes of the frame as the CRC. 1 = CRC generation is disabled. The CRC is not appended to the end of the frame. The CRCDIS bit should only be changed while the CRC is not being transmitted.
1	LBREAD	**Last Byte Read** 0 = Reading the transmit FIFO has no effect on frame transmission. 1 = Software can indicate the last byte for a frame by reading the transmit FIFO immediately after writing the last byte to the transmit FIFO.
0	LBNOW	**Last Byte Now** 0 = When read, the LBNOW bit is always zero. 1 = When written with a 1 by software, the LBNOW bit indicates to the transmitter that the last byte for a frame has been placed in the transmit FIFO.

Programming Notes

HDLC Channel A Transmit Control 1 (HATCON1) Offset 04h
HDLC Channel B Transmit Control 1 (HBTCON1) Offset 44h
HDLC Channel C Transmit Control 1 (HCTCON1) Offset 84h
HDLC Channel D Transmit Control 1 (HDTCON1) Offset C4h

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	*Res*				FLAGIDL	MLTDRP	AUTOCTS	TMSBF	TXCINV	GCIDEN	ODRV		TDELAY			
Software Read/Write	R				R/W	R/W	R/W	R/W	R/W	R/W	R/W		R/W			
Hardware Set/Clear	—				—	—	—	—	—	—	—		—			
Chip Reset Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	Channel:			
Microcontroller:	**A**	**B**	**C**	**D**
Am186CC	✓	✓	✓	✓
Am186CH	✓	✓		
Am186CU				

Register Description

This register sets operating modes for the transmitter.

Bit Definitions

Bit **Name** **Function**

15–12 Res **Reserved**
For compatibility with future devices, always write this bit field with its chip reset default value.

11 FLAGIDL **Flag Idle**
0 = Mark idle. Mark sequences are transmitted in the idle state. A mark sequence is fifteen or more 1s.

1 = Flag idle. Flag sequences are transmitted in the idle state. A flag sequence is one 0, then six 1s, then one 0.

The FLAGIDL bit should only be changed when the TSTOP bit is 1 (see page 2-17).

10 MLTDRP **Multidrop**
0 = The multidrop bus mode is not enabled.

1 = The multidrop bus mode is enabled. When the MLTDRP bit is 1, the AUTOCTS, FLAGIDL, and GCIDEN bits must be 0 and the output drive must be open drain (ODRV = 10b). TDELAY should be set for 8, 9, 10, or 11.

9 AUTOCTS **Automatic CTS**
0 = CTS is ignored.

1 = CTS is used to automatically enable and disable transmission.

8 TMSBF **Transmit MSB First**
0 = Transmit data least-significant bit (LSB) first.

1 = Transmit data most-significant bit (MSB) first.

HDLC normally is LSB first. Data could be corrupted if the TMSBF bit is changed when the TSTOP bit is 0 (see page 2-17).

7 TXCINV **Transmit Clock Invert**
0 = The transmit clock is not inverted.

1 = The transmit clock is inverted.

The TXCINV bit should only be changed when the TSTOP bit is 1 (see page 2-17). Inversion of the transmit clock (TCLK) signal is only recommended for Data Communications Equipment (DCE) operation and does not operate properly with time-division-multiplexed configurations.

Bit	Name	Function
6	GCIDEN	**GCI D Channel Enable** 0 = The HDLC channel's transmission is independent of control of the GCI D-channel. 1 = Transmission will not begin until the GCI interface has control of the D-channel. Only one HDLC channel at a time should have the GCIDEN bit set. When the GCIDEN bit is 1, the time slot assigner must be programmed for this HDLC to use the D-channel time slot. When the GCIDEN bit is 1, the FLAGIDL bit should be 0. The GCIDEN bit should only be changed when the TSTOP bit is 1 (see page 2-17).
5–4	ODRV	**Output Drive** The ODRV bit controls the state of the transmit data output. The ODRV bit has no effect when in Pulse Code Modulation (PCM) Highway or GCI mode. 00 = Three state 01 = Totem pole (normal drive) 10 = Open drain 11 = Reserved
3–0	TDELAY	**Transmit Delay** The TDELAY bit field controls the number of 1s to wait before transmitting in multidrop mode. The TDELAY bit field must be 0000 if the MLTDRP bit is 0. The TDELAY bit should only be changed when the TSTOP bit is 1 (see page 2-17). 0000 = Do not wait. 0001 = Wait for one 1. 0010 = Wait for two 1s. and so on... 1111 = Wait for fifteen 1s.

Programming Notes

HDLC Channel A Receive Control 0 (HARCON0) Offset 06h
HDLC Channel B Receive Control 0 (HBRCON0) Offset 46h
HDLC Channel C Receive Control 0 (HCRCON0) Offset 86h
HDLC Channel D Receive Control 0 (HDRCON0) Offset C6h

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	*Res*				RTHRSH		RCPST	RMSBF	RXCINV	RREJECT	RSTOP	HREN	MINRL			
Software Read/Write	R				R/W		R/W	R/W	R/W	R/W	R/W	R/W	R/W			
Hardware Set/Clear	—				—		—	—	—	—	—	—	—			
Chip Reset Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Microcontroller:	Channel: A	B	C	D
Am186CC	✓	✓	✓	✓
Am186CH	✓	✓		
Am186CU				

Register Description

This register sets operating modes for the receiver.

Bit Definitions

Bit **Name** **Function**

15–12 Res **Reserved**
For compatibility with future devices, always write this bit field with its chip reset default value.

11–10 RTHRSH **Receive Threshold**
The RTHRSH bit programs the amount of data needed in the receive FIFO before giving an interrupt. This same size is used to determine when to request a DMA access.

00 = 1 byte

01 = 8 bytes

10 = 16 bytes

11 = 32 bytes

9 RCPST **Receive Copy Status**
0 = The FBCNT[15–13] bits in Receive Frame Status 2 are not changed (see page 2-31).

1 = The FABORT bit from Receive Frame Status 3 (see page 2-34) replaces the FBCNT[15] bit in Receive Frame Status 2, the FLONG bit replaces the FBCNT[14] bit, and the FSHORT bit replaces the FBCNT[13] bit.

This feature is intended for use with the SmartDMA controller.

8 RMSBF **Receive MSB First**
0 = Receive data LSB first.

1 = Receive data MSB first.

HDLC normally is LSB first. Data could be corrupted if the RMSBF bit is changed when the RSTOP bit is 0.

Bit	Name	Function
7	RXCINV	**Receive-Clock Invert** 0 = The receive clock is not inverted. 1 = The receive clock is inverted. The RXCINV bit should not be changed unless the RSTOP bit is 1 or the receiver has been disabled (by clearing the HREN bit and waiting for the end of frame). Inversion of the receive clock is only recommended for DCE operation and will not operate properly with time-division-multiplexed configurations.
6	RREJECT	**Receiver Reject** 0 = The receive performs normally. 1 = Reception is disabled but the channel's ready-to-receive ($\overline{\text{RTR}}$) signal is unaffected.
5	RSTOP	**Receiver Stop** 0 = The receive performs normally. 1 = Reception is disabled and the channel's $\overline{\text{RTR}}$ signal is deasserted.
4	HREN	**Receiver Enable** 0 = Reception is disabled at the end of the current frame; then the channel's $\overline{\text{RTR}}$ signal is deasserted. 1 = The receiver is enabled and will start receiving after finding a flag. After setting the HREN bit to enable the receiver, the device software must reset the HDLC FIFOs by setting the HRESET bit in the HxCON register (see page 2-5). This clears any invalid data in the receiver FIFO that might be mistaken as the start of the data stream. Invalid data is a concern when using Transparent mode (TRANSM = 1 in the HxCON register) because in Transparent mode the receiver cannot rely on flag sequences to indicate the start of valid data.
3–0	MINRL	**Minimum Receive Length** The MINRL bit field configures the length in bytes of the smallest frame that will be received without causing a short frame error (reported by the FSHORT status bit, see page 2-34). If the MINRL bit field is set to a value of 2 or less, no frames affect the FSHORT bit. The receiver cannot receive a frame shorter than two bytes and automatically rejects it regardless of the value of the MINRL bit field; frames rejected for this reason are reported by the VSHORT status bit (see page 2-20). 0000 = Only very short frames are reported (see the VSHORT bit on page 2-20). 0001 = Only very short frames are reported. 0010 = Only very short frames are reported. 0011 = Frames that are 16–23 bits long set the FSHORT bit (see page 2-34). 0100 = Frames that are 16–31 bits long set the FSHORT bit. 0101 = Frames that are 16–39 bits long set the FSHORT bit. . . . . . 1111 = Frames that are 16–119 bits long set the FSHORT bit.

Programming Notes

HDLC Channel A Receive Max Length 1 (HARCON1) Offset 08h
HDLC Channel B Receive Max Length 1 (HBRCON1) Offset 48h
HDLC Channel C Receive Max Length 1 (HCRCON1) Offset 88h
HDLC Channel D Receive Max Length 1 (HDRCON1) Offset C8h

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	MAXRL															
Software Read/Write	R/W															
Hardware Set/Clear	—															
Chip Reset Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Microcontroller:	Channel: A	B	C	D
Am186CC	✔	✔	✔	✔
Am186CH	✔	✔		
Am186CU				

Register Description

This register sets the maximum length for a received frame.

Bit Definitions

Bit	Name	Function
15–0	MAXRL	**Maximum Receive Length** The MAXRL bit field configures the maximum length of a frame that will be received. Because the HDLC receiver automatically rejects frames less than two bytes in length, the MAXRL bit field should never be set to 2 or less.

Programming Notes

HDLC Channel A Status (HASTATE) Offset 0Ah
HDLC Channel B Status (HBSTATE) Offset 4Ah
HDLC Channel C Status (HCSTATE) Offset 8Ah
HDLC Channel D Status (HDSTATE) Offset CAh

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	*Res*										CTSS	RTRS	ABORTS	MARKIS	FLAGS	FRAMES
Software Read/Write	R										R	R	R	R	R	R
Hardware Set/Clear	—										S/C	S/C	S/C	S/C	S/C	S/C
Chip Reset Default	0	0	0	0	0	0	0	0	0	0	x	1	0	0	0	0

Microcontroller:	Channel: A	B	C	D
Am186CC	✓	✓	✓	✓
Am186CH	✓	✓		
Am186CU				

Register Description

This register contains read-only status for the receiver and transmitter.

Bit Definitions

Bit	Name	Function
15–6	Res	**Reserved** For compatibility with future devices, always write this bit field with its chip reset default value.
5	CTSS	**$\overline{\text{CTS}}$ State** 0 = The current state of the internal clear-to-send ($\overline{\text{CTS}}$) signal is 0 or asserted. When the GCIDEN bit is 1 (see page 2-10), CTSS = 0 indicates that access to the GCI D-channel is currently allowed. 1 = The current state of the internal clear-to-send ($\overline{\text{CTS}}$) signal is 1 or deasserted. When the GCIDEN bit is 1, CTSS = 1 indicates that access to the GCI D-channel is not currently allowed.
4	RTRS	**$\overline{\text{RTR}}$ State** 0 = The current state of the ready-to-receive ($\overline{\text{RTR}}$) signal output is Low. 1 = The current state of the ready-to-receive ($\overline{\text{RTR}}$) signal output is High.
3	ABORTS	**Abort State** 0 = The receiver is not in the abort state. 1 = The receiver is currently in the abort state. An abort sequence (seven to fourteen 1s) was detected while in frame state, and a flag or mark idle sequence has not been detected since the abort.
2	MARKIS	**Mark Idle State** 0 = The receiver is not in the mark idle state. 1 = The receiver is currently in the mark idle state. A mark idle sequence (fifteen or more 1s) was detected, and a flag has not been detected since the mark idle began.

Bit	Name	Function
1	FLAGS	**Flag State** 0 = The receiver is not in the flag state. 1 = The receiver is currently in the flag state. A flag sequence (one 0, six 1s, then one 0) was the last fully-decoded sequence.
0	FRAMES	**Frame State** 0 = The receiver is not in the frame state. 1 = The receiver is currently in the frame state. A flag followed by data was detected, and the subsequent data has not contained an abort, mark idle, or flag.

Programming Notes

Only one of the ABORTS, MARKIS, FLAGS, or FRAMES bits can be 1 at any given time. After reset, none of the ABORTS, MARKIS, FLAGS, or FRAMES bits are 1 until the internal pipe is filled. After the first flag is detected, this register always reports that the receiver is in the aborts, marks, flags, or frame state unless the receiver is reset. No line status bits are valid in transparent mode except the CTSS and RTRS bits.

HDLC Channel A Interrupt Status 0 (HAISTAT0) Offset 0Ch
HDLC Channel B Interrupt Status 0 (HBISTAT0) Offset 4Ch
HDLC Channel C Interrupt Status 0 (HCISTAT0) Offset 8Ch
HDLC Channel D Interrupt Status 0 (HDISTAT0) Offset CCh

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	*Res*			REOF	RTHRES	RDATA1	TTHRES	TDATA1	*Res*			FABRST	CTSLST	TUFLO	TGOODF	TSTOP
Software Read/Write	R			R	R	R	R	R	R			R/W0	R/W0	R/W0	R/W0	R/W0
Hardware Set/Clear	—			S/C	S/C	S/C	S/C	S/C	—			S	S	S	S	S
Chip Reset Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Microcontroller:	Channel: A	B	C	D
Am186CC	✓	✓	✓	✓
Am186CH	✓	✓		
Am186CU				

Register Description

This is the interrupt status register for the transmitter as well as the receive FIFO. All bits can generate an interrupt if not masked off in the HxIMSK0 register (see page 2-18). The R/W0 bits can only be cleared by software writing a 0 or by a microcontroller reset (external or watchdog timer). Setting the HRESET bit (see page 2-5) does not clear these bits. Additional receiver interrupt bits can be found in the HxISTAT1 register (see page 2-20).

Bit Definitions

Bit **Name** **Function**

15–13 Res **Reserved**
For compatibility with future devices, always write this bit field with its chip reset default value.

12 REOF **Received End-of-Frame**
0 = No status bytes presently reside in the receive FIFO.

1 = Indicates that one or more status bytes presently reside in the receive FIFO.

11 RTHRES **Receive FIFO Threshold Reached**
0 = The number of data bytes has dropped below the threshold value programmed in the RTHRSH bit field (see page 2-11), or the receive FIFO has been read and there are status bytes present in the FIFO. In the latter case (status bytes present), the total number of bytes in the FIFO might still exceed RTHRSH.

1 = The programmed threshold has been reached or exceeded in the receive FIFO and there were no status bytes present in the FIFO at that time. This guarantees that bytes can be read up to the programmed threshold value that are exclusively data bytes, not status bytes.

If a status byte enters the FIFO after the RTHRES bit is set, the first read of the FIFO clears the RTHRES bit even though the threshold value of data bytes is present in the FIFO. After the RTHRES bit is set, software can read the programmed threshold value of data bytes consecutively without checking the RTHRES bit state.

10 RDATA1 **One Receive Data Byte Available**
0 = No data byte is available in the receive FIFO.

1 = Indicates that one byte of data (not status) is presently available in the receive FIFO. The indication that status is not present applies to the currently available byte, not the entire FIFO contents (e.g., if the next available byte is status and the following byte is data, the RDATA1 bit is not set).

9 TTHRES **Transmit FIFO Threshold Reached**
0 = Either the event has not occurred or software has filled the FIFO to within the transmit free space threshold set in the TTHRSH bit field (see page 2-7).

1 = The amount of space available in the transmit FIFO exceeded the transmit threshold value.

Bit	Name	Function
8	TDATA1	**One Transmit Data Byte Space Available** 0 = No space is available in the transmit FIFO. 1 = Indicates that space is available for at least one more data byte in the transmit FIFO.
7–5	Res	**Reserved** For compatibility with future devices, always write this bit field with its chip reset default value.
4	FABRST	**Forced Abort Sent** 0 = Either the event has not occurred or software has cleared the bit. 1 = The transmitter has completed sending out a forced abort caused by the setting of the FORABR bit (see page 2-7). If multiple aborts are sent out because the FORABR bit is 1 for an extended time, the FABRST bit goes to 1 after the FORABR bit is 0 and the last abort is complete. If the MLTDRP bit is 1 (see page 2-9) or the GCIDEN bit is 1 (see page 2-10), and the FORABR bit is 1, only one abort is sent. In this case, the FABRST bit is 1 after the single abort has been transmitted (and while the FORABR bit is still 1).
3	CTSLST	**Clear-to-Send ($\overline{\text{CTS}}$) Signal Lost** 0 = Either the event has not occurred or software has cleared the bit. 1 = The channel's $\overline{\text{CTS}}$ signal was lost during transmission of a frame (not including flags). One abort sequence is transmitted, followed by idle sequences. When the MLTDRP bit is 1 (see page 2-9), CTSLST = 1 indicates that the current transmission was stopped (idle sequences are sent) because of a collision. The CTSLST bit must be cleared before the transmitter will resume normal operation. If a collision stops transmission during the closing flag of a frame, the transmit status might already have been reported in the HxISTAT0 register. In this case, the transmit status reported in the HxISTAT0 register will not reflect the collision that occurred during the closing flag. Transmission will still stop during the closing flag and the CTSLST bit is set. If $\overline{\text{CTS}}$ is lost during the closing flag (or opening flag) and the MLTDRP bit is 0 (see page 2-9), the CTSLST is not set.
2	TUFLO	**Underflow** 0 = Either the event has not occurred or software has cleared the bit. 1 = The transmit FIFO has underflowed. One abort sequence is transmitted, followed by idle sequences. This bit must be cleared before the transmitter will resume normal operation.
1	TGOODF	**Good Frame Transmitted** 0 = Either the event has not occurred or software has cleared the bit. 1 = The last byte of a frame has been transmitted (which does not include the closing flag) without any errors.
0	TSTOP	**Transmit Stopped** 0 = Either the event has not occurred or software has cleared the bit. 1 = The transmitter stopped because the HTEN bit was cleared and the transmitter has completed transmitting the current frame. If the transmitter is not transmitting when the HTEN bit is cleared, the TSTOP bit is set immediately.

Programming Notes

All HxISTAT0 register bits can generate an interrupt if not masked off in the HxIMSK0 register. However, before an HDLC channel can generate any interrupts, software must configure the interrupt Channel 4–7 Control registers correctly. The appropriate CH4CON, CH5CON, CH6CON, or CH7CON registers must be configured to select each required interrupt channel's internal source (HDLC A, B, C and D, respectively) and to enable the interrupt channel. See Chapter 9, "Interrupt Controller Registers".

HDLC Channel A Interrupt Mask 0 (HAIMSK0) Offset 0Eh
HDLC Channel B Interrupt Mask 0 (HBIMSK0) Offset 4Eh
HDLC Channel C Interrupt Mask 0 (HCIMSK0) Offset 8Eh
HDLC Channel D Interrupt Mask 0 (HDIMSK0) Offset CEh

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	*Res*			REOF	RTHRES	RDATA1	TTHRES	TDATA1	*Res*			FABRST	CTSLST	TUFLO	TGOODF	TSTOP
Software Read/Write	R			R/W	R/W	R/W	R/W	R/w	R			R/W	R/W	R/W	R/W	R/w
Hardware Set/Clear	—			—	—	—	—	—	—			—	—	—	—	—
Chip Reset Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Microcontroller:	Channel: A	B	C	D
Am186CC	✔	✔	✔	✔
Am186CH	✔	✔		
Am186CU				

Register Description

This is the mask register for the HxISTAT0 register (see page 2-16).

Bit Definitions

Bit	Name	Function
15–13	Res	**Reserved** For compatibility with future devices, always write this bit field with its chip reset default value.
12	REOF	**Received End-of-Frame Interrupt Mask** 0 = The REOF bit interrupt in the HxISTAT0 register is masked off. 1 = The REOF bit interrupt in the HxISTAT0 register is enabled.
11	RTHRES	**Receive FIFO Threshold Reached Interrupt Mask** 0 = The RTHRES bit interrupt in the HxISTAT0 register is masked off. 1 = The RTHRES bit interrupt in the HxISTAT0 register is enabled.
10	RDATA1	**One Receive Data Byte Available Interrupt Mask** 0 = The RDATA1 bit interrupt in the HxISTAT0 register is masked off. 1 = The RDATA1 bit interrupt in the HxISTAT0 register is enabled.
9	TTHRES	**Transmit FIFO Threshold Reached Interrupt Mask** 0 = The TTHRES bit interrupt in the HxISTAT0 register is masked off. 1 = The TTHRES bit interrupt in the HxISTAT0 register is enabled.
8	TDATA1	**One Transmit Data Byte Space Available Interrupt Mask** 0 = The TDATA1 bit interrupt in the HxISTAT0 register is masked off. 1 = The TDATA1 bit interrupt in the HxISTAT0 register is enabled.
7–5	Res	**Reserved** For compatibility with future devices, always write this bit field with its chip reset default value.
4	FABRST	**Forced Abort Sent Interrupt Mask** 0 = The FABRST bit interrupt in the HxISTAT0 register is masked off. 1 = The FABRST bit interrupt in the HxISTAT0 register is enabled.
3	CTSLST	**CTS Lost Interrupt Mask** 0 = The CTSLST bit interrupt in the HxISTAT0 register is masked off. 1 = The CTSLST bit interrupt in the HxISTAT0 register is enabled.

Bit	Name	Function
2	TUFLO	**Underflow Interrupt Mask** 0 = The TUFLO bit interrupt in the HxISTAT0 register is masked off. 1 = The TUFLO bit interrupt in the HxISTAT0 register is enabled.
1	TGOODF	**Good Frame Transmitted Interrupt Mask** 0 = The TGOODF bit interrupt in the HxISTAT0 register is masked off. 1 = The TGOODF bit interrupt in the HxISTAT0 register is enabled.
0	TSTOP	**Transmit Stopped Interrupt Mask** 0 = The TSTOP bit interrupt in the HxISTAT0 register is masked off. 1 = The TSTOP bit interrupt in the HxISTAT0 register is enabled.

Programming Notes

All HxISTAT0 register bits can generate an interrupt if not masked off in the HxIMSK0 register. However, before an HDLC channel can generate any interrupts, software must configure the interrupt Channel 4–7 Control registers correctly. The appropriate CH4CON, CH5CON, CH6CON, or CH7CON registers must be configured to select each required interrupt channel’s internal source (HDLC A, B, C and D, respectively) and to enable the interrupt channel. See Chapter 9, “Interrupt Controller Registers”.

HDLC Channel A Interrupt Status 1 (HAISTAT1) Offset 10h
HDLC Channel B Interrupt Status 1 (HBISTAT1) Offset 50h
HDLC Channel C Interrupt Status 1 (HCISTAT1) Offset 90h
HDLC Channel D Interrupt Status 1 (HDISTAT1) Offset D0h

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	*Res*						MAMC	SFMC	SHORT	VSHORT	RTRDES	ROFLO	ABORTE	MARKIE	FLAGE	FRAMEE
Software Read/Write	R						R/W0	R/W0	R/W0	R/W0	R/W0	R/W0	R/W0	R/W0	R/W0	R/W0
Hardware Set/Clear	—						S	S	S	S	S	S	S	S	S	S
Chip Reset Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Microcontroller:	Channel: A	B	C	D
Am186CC	✔	✔	✔	✔
Am186CH	✔	✔		
Am186CU				

Register Description

This is the interrupt register for the receiver. All bits can generate an interrupt if not masked off in the HxIMSK1 register (see page 2-22). The R/W0 bits can only be cleared by software writing a 0 or by a reset generated externally or by the watchdog timer. Setting the HRESET bit (see page 2-5) does not clear these bits.

Three additional receiver interrupt bits can be found in the HxISTAT0 register (see page 2-16).

Bit Definitions

Bit **Name** **Function**

15–10 Res **Reserved**
For compatibility with future devices, always write this bit field with its chip reset default value.

9 MAMC **Mismatch Address Max Count**
0 = Either the event has not occurred or software has cleared the bit.

1 = The mismatch address counter (HMACNT bit field, see page 2-37) exceeded its maximum count and rolled over to 0.

8 SFMC **Short Frame Max Count**
0 = Either the event has not occurred or software has cleared the bit.

1 = The short frame counter (HSFCNT bit field, see page 2-35) exceeded its maximum count and rolled over to 0.

7 SHORT **Short Frame Received**
0 = Either the event has not occurred or software has cleared the bit.

1 = A frame was received that had a length less than specified in minimum receive length (MINRL bit field, see page 2-12), and also had an address match and was large enough that part of it was transferred to the receive FIFO. The FSHORT bit is set in the third status byte (see page 2-34).

6 VSHORT **Very Short Frame Received**
0 = Either the event has not occurred or software has cleared the bit.

1 = A frame was received that had a length less than specified in minimum receive length (MINRL bit field, see page 2-12), but was small enough to be discarded before being transferred to the receive FIFO. Frames less than two bytes are counted as a very short frame whether or not there is an address match.

5 RTRDES **Ready-to-Receive ($\overline{\text{RTR}}$) Signal Deasserted**
0 = Either the event has not occurred or software has cleared the bit.

1 = The ready-to-receive signal was deasserted (regardless of the line status when the event occurred).

Bit	Name	Function
4	ROFLO	**Overflow** 0 = Either the event has not occurred or software has cleared the bit. 1 = The receive FIFO has overflowed. This bit may be cleared even if the overflow that caused the event has not been corrected. After an overflow occurs, the receiver resumes normal operation as soon as the receive FIFO is emptied. Reception is not dependent on this bit's value.
3	ABORTE	**Abort Entered** 0 = Either the event has not occurred or software has cleared the bit. 1 = The receiver entered the abort state. An abort sequence (seven to fourteen 1s) was detected while in frame state.
2	MARKIE	**Mark Idle Entered** 0 = Either the event has not occurred or software has cleared the bit. 1 = The receiver entered the mark idle state. A mark idle sequence (fifteen or more 1s) was detected.
1	FLAGE	**Flag Entered** 0 = Either the event has not occurred or software has cleared the bit. 1 = The receiver entered the flag state. A flag sequence (one 0, six 1s, then one 0) was detected.
0	FRAMEE	**Frame Entered** 0 = Either the event has not occurred or software has cleared the bit. 1 = The receiver entered the frame state. A flag followed by data was detected.

Programming Notes

All HxISTAT1 register bits can generate an interrupt if not masked off in the HxIMSK1 register. However, before an HDLC channel can generate any interrupts, software must configure the interrupt Channel 4–7 Control registers correctly. The appropriate CH4CON, CH5CON, CH6CON, or CH7CON registers must be configured to select each required interrupt channel's internal source (HDLC A, B, C and D, respectively) and to enable the interrupt channel. See Chapter 9, “Interrupt Controller Registers”.

HDLC Channel A Interrupt Mask 1 (HAIMSK1) Offset 12h
HDLC Channel B Interrupt Mask 1 (HBIMSK1) Offset 52h
HDLC Channel C Interrupt Mask 1 (HCIMSK1) Offset 92h
HDLC Channel D Interrupt Mask 1 (HDIMSK1) Offset D2h

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	*Res*						MAMC	SFMC	SHORT	VSHORT	RTRDES	ROFLO	ABORTE	MARKIE	FLAGE	FRAMEE
Software Read/Write	R						R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Hardware Set/Clear	—						—	—	—	—	—	—	—	—	—	—
Chip Reset Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Microcontroller:	Channel: A	B	C	D
Am186CC	✓	✓	✓	✓
Am186CH	✓	✓		
Am186CU				

Register Description

This is the mask register for the HxISTAT1 register (see page 2-20).

Bit Definitions

Bit	Name	Function
15–10	Res	**Reserved** For compatibility with future devices, always write this bit field with its chip reset default value.
9	MAMC	**Mismatch Address Max Count Interrupt Mask** 0 = The MAMC bit interrupt in the HxISTAT1 register is masked off. 1 = The MAMC bit interrupt in the HxISTAT1 register is enabled.
8	SFMC	**Short Frame Max Count Interrupt Mask** 0 = The SFMC bit interrupt in the HxISTAT1 register is masked off. 1 = The SFMC bit interrupt in the HxISTAT1 register is enabled.
7	SHORT	**Short Frame Received Interrupt Mask** 0 = The SHORT bit interrupt in the HxISTAT1 register is masked off. 1 = The SHORT bit interrupt in the HxISTAT1 register is enabled.
6	VSHORT	**Very Short Frame Received Interrupt Mask** 0 = The VSHORT bit interrupt in the HxISTAT1 register is masked off. 1 = The VSHORT bit interrupt in the HxISTAT1 register is enabled.
5	RTRDES	**RTR Deasserted Interrupt Mask** 0 = The RTRDES bit interrupt in the HxISTAT1 register is masked off. 1 = The RTRDES bit interrupt in the HxISTAT1 register is enabled.
4	ROFLO	**Overflow Interrupt Mask** 0 = The ROFLO bit interrupt in the HxISTAT1 register is masked off. 1 = The ROFLO bit interrupt in the HxISTAT1 register is enabled.
3	ABORTE	**Abort Entered Interrupt Mask** 0 = The ABORTE bit interrupt in the HxISTAT1 register is masked off. 1 = The ABORTE bit interrupt in the HxISTAT1 register is enabled.

Bit	Name	Function
2	MARKIE	**Mark Idle Entered Interrupt Mask** 0 = The MARKIE bit interrupt in the HxISTAT1 register is masked off. 1 = The MARKIE bit interrupt in the HxISTAT1 register is enabled.
1	FLAGE	**Flag Entered Interrupt Mask** 0 = The FLAGE bit interrupt in the HxISTAT1 register is masked off. 1 = The FLAGE bit interrupt in the HxISTAT1 register is enabled.
0	FRAMEE	**Frame Entered Interrupt Mask** 0 = The FRAMEE bit interrupt in the HxISTAT1 register is masked off. 1 = The FRAMEE bit interrupt in the HxISTAT1 register is enabled.

Programming Notes

All HxISTAT1 register bits can generate an interrupt if not masked off in the HxIMSK1 register. However, before an HDLC channel can generate any interrupts, software must configure the interrupt Channel 4–7 Control registers correctly. The appropriate CH4CON, CH5CON, CH6CON, or CH7CON registers must be configured to select each required interrupt channel’s internal source (HDLC A, B, C and D, respectively) and to enable the interrupt channel. See Chapter 9, “Interrupt Controller Registers”.

HDLC Channel A Transmit FIFO Data (HATD) Offset 14h
HDLC Channel B Transmit FIFO Data (HBTD) Offset 54h
HDLC Channel C Transmit FIFO Data (HCTD) Offset 94h
HDLC Channel D Transmit FIFO Data (HDTD) Offset D4h

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	*Res*								TDATA							
Software Read/Write	R								W							
Hardware Set/Clear	—								—							
Chip Reset Default	0	0	0	0	0	0	0	0	x	x	x	x	x	x	x	x

	Channel:			
Microcontroller:	**A**	**B**	**C**	**D**
Am186CC	✓	✓	✓	✓
Am186CH	✓	✓		
Am186CU				

Register Description

This is the location for writing data to the HDLC Transmit FIFO.

Bit Definitions

Bit	Name	Function
15–8	Res	**Reserved** For compatibility with future devices, always write this bit field with its chip reset default value.
7–0	TDATA	**Transmit FIFO Data** When transmitting a frame, the data is written to the HDLC Transmit FIFO. After writing the last byte of a frame into the Transmit FIFO, the LBNOW bit (in the HxTCON0 register, see page 2-8) must be set. Alternatively the LBREAD bit must be set (in the HxTCON0 register) and then this register can be read immediately after the last byte is written to the FIFO to infer the last byte of the frame.

Programming Notes

HDLC Channel A Receive FIFO Data (HARD) Offset 16h
HDLC Channel B Receive FIFO Data (HBRD) Offset 56h
HDLC Channel C Receive FIFO Data (HCRD) Offset 96h
HDLC Channel D Receive FIFO Data (HDRD) Offset D6h

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	STAT1A	STAT0A	STATNUM		RTHRES	RDATA1	TTHRES	TDATA1	RDATA							
Software Read/Write	R	R	R		R	R	R	R	R							
Hardware Set/Clear	S/C	S/C	S/C		S/C	S/C	S/C	S/C	S/C							
Chip Reset Default	0	0	0	0	0	0	0	0	x	x	x	x	x	x	x	x

Microcontroller:	Channel: A	B	C	D
Am186CC	✓	✓	✓	✓
Am186CH	✓	✓		
Am186CU				

Register Description

This is the location for sequentially reading data and status bytes from the HDLC Receive FIFO.

Bit Definitions

Bit	Name	Function
15	STAT1A	**Any HxISTAT1 Register Interrupt** This bit represents the logical OR of all the unmasked interrupt bits in the HxISTAT1 register (see page 2-20).
14	STAT0A	**Any HxISTAT0 Register Interrupt** This bit represents the logical OR of all the unmasked interrupt bits in the HxISTAT0 register (see page 2-16). It does not include the REOF, RTHRES, RDATA1, TTHRES, or TDATA1 bits because these can be inferred by other bits available in this register.
13–12	STATNUM	**Status Byte Number** These two bits indicate which byte of status is presently available in the receive FIFO (i.e., 1–3). If the byte is data, these bits are both 0. 00 = The RDATA bit field is data (use this register description). 01 = The RDATA bit field contains Receive Frame Status 1 (see page 2-29). 10 = The RDATA bit field contains Receive Frame Status 2 (see page 2-31). 11 = The RDATA bit field contains Receive Frame Status 3 (see page 2-33).
11	RTHRES	**Receive FIFO Threshold Reached** This bit is a copy of the RTHRES bit located in the HxISTAT0 register (see page 2-16).
10	RDATA1	**One Receive Data Byte Available** This bit is a copy of the RDATA1 bit located in the HxISTAT0 register (see page 2-16).
9	TTHRES	**Transmit FIFO Threshold Reached** This bit is a copy of the TTHRES bit located in the HxISTAT0 register (see page 2-16).
8	TDATA1	**One Transmit Byte Space Available** This bit is a copy of the TDATA1 bit located in the HxISTAT0 register (see page 2-16).
7–0	RDATA	**Receive FIFO Data** This is the present data byte available for read from the receive FIFO.

Programming Notes

When receiving a frame, the HDLC Receive FIFO contains the data. Reading this register increments to the next byte of data. After all the data for the frame has been read, this register contains the three bytes of status for the frame (see the HxRFS1, HxRFS2, and HxRFS3 register descriptions, beginning on page 2-29).

HDLC Channel A Receive FIFO Data Peek (HARDP) Offset 18h
HDLC Channel B Receive FIFO Data Peek (HBRDP) Offset 58h
HDLC Channel C Receive FIFO Data Peek (HCRDP) Offset 98h
HDLC Channel D Receive FIFO Data Peek (HDRDP) Offset D8h

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	STAT1A	STAT0A	STATNUM		RTHRES	RDATA1	TTHRES	TDATA1	RDATA							
Software Read/Write	R	R	R		R	R	R	R	R							
Hardware Set/Clear	S/C	S/C	S/C		S/C	S/C	S/C	S/C	S/C							
Chip Reset Default	0	0	0	0	0	0	0	0	x	x	x	x	x	x	x	x

Microcontroller:	Channel: A	B	C	D
Am186CC	✓	✓	✓	✓
Am186CH	✓	✓		
Am186CU				

Register Description

This is a copy of the HDLC Receive FIFO Data register (HxRD, see page 2-25) that does not change when read.

Bit Definitions

Bit **Name** **Function**

15 STAT1A **Any HxISTAT1 Register Interrupt (Same as HxRD)**
This bit represents the logical OR of all the unmasked interrupt bits in the HxISTAT1 register (see page 2-20).

14 STAT0A **Any HxISTAT0 Register Interrupt (Same as HxRD)**
This bit represents the logical OR of all the unmasked interrupt bits in the HxISTAT0 register (see page 2-16). It does not include the REOF, RTHRES, RDATA1, TTHRES, or TDATA1 bits because these can be inferred by other bits available in this register.

13–12 STATNUM **Status Byte Number (Same as HxRD)**
These two bits indicate which byte of status is presently available in the receive FIFO (i.e., 1–3). If the byte is data, these bits are both 0.

00 = The RDATA bit field is data (see page 2-25).

01 = The RDATA bit field contains Receive Frame Status 1 (see page 2-29).

10 = The RDATA bit field contains Receive Frame Status 2 (see page 2-31).

11 = The RDATA bit field contains Receive Frame Status 3 (see page 2-33).

11 RTHRES **Receive FIFO Threshold Reached (Same as HxRD)**
This bit is a copy of the RTHRES bit located in the HxISTAT0 register (see page 2-16).

10 RDATA1 **One Receive Data Byte Available (Same as HxRD)**
This bit is a copy of the RDATA1 bit located in the HxISTAT0 register (see page 2-16).

9 TTHRES **Transmit FIFO Threshold Reached (Same as HxRD)**
This bit is a copy of the TTHRES bit located in the HxISTAT0 register (see page 2-16).

8 TDATA1 **One Transmit Byte Space Available (Same as HxRD)**
This bit is a copy of the TDATA1 bit located in the HxISTAT0 register (see page 2-16).

7–0 RDATA **Receive FIFO Data (Same as HxRD)**
This is the present data byte available for read from the receive FIFO.

Programming Notes

HDLC Channel A Receive Frame Status 1 (HARFS1) Same as HARD
HDLC Channel B Receive Frame Status 1 (HBRFS1) Same as HBRD
HDLC Channel C Receive Frame Status 1 (HCRFS1) Same as HCRD
HDLC Channel D Receive Frame Status 1 (HDRFS1) Same as HDRD

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	STAT1A	STAT0A	STATNUM		RTHRES	RDATA1	TTHRES	TDATA1	FBCNT[7–0]							
Software Read/Write	R	R	R		R	R	R	R	R							
Hardware Set/Clear	S/C	S/C	S/C		S/C	S/C	S/C	S/C	S/C							
Chip Reset Default	0	0	0	0	0	0	0	0	x	x	x	x	x	x	x	x

Microcontroller:	Channel: A	B	C	D
Am186CC	✓	✓	✓	✓
Am186CH	✓	✓		
Am186CU				

Register Description

After the last frame data byte is read from the HDLC Receive FIFO Data register (see page 2-25), the next three bytes read contain frame status information. Receive Frame Status 1 is the first byte of status information.

Bit Definitions

Bit	Name	Function
15	STAT1A	**Any HxISTAT1 Register Interrupt (Same as HxRD)** This bit represents the logical OR of all the unmasked interrupt bits in the HxISTAT1 register (see page 2-20).
14	STAT0A	**Any HxISTAT0 Register Interrupt (Same as HxRD)** This bit represents the logical OR of all the unmasked interrupt bits in the HxISTAT0 register (see page 2-16). It does not include the REOF, RTHRES, RDATA1, TTHRES, or TDATA1 bits because these can be inferred by other bits available in this register.
13–12	STATNUM	**Status Byte Number (Same as HxRD)** These two bits indicate which byte of status is presently available in the receive FIFO (i.e., 1–3). If the byte is data, these bits are both 0. 00 = The RDATA bit field is data (see page 2-25). 01 = The RDATA bit field contains Receive Frame Status 1 (use this register description). 10 = The RDATA bit field contains Receive Frame Status 2 (see page 2-31). 11 = The RDATA bit field contains Receive Frame Status 3 (see page 2-33).
11	RTHRES	**Receive FIFO Threshold Reached (Same as HxRD)** This bit is a copy of the RTHRES bit located in the HxISTAT0 register (see page 2-16).
10	RDATA1	**One Receive Data Byte Available (Same as HxRD)** This bit is a copy of the RDATA1 bit located in the HxISTAT0 register (see page 2-16).
9	TTHRES	**Transmit FIFO Threshold Reached (Same as HxRD)** This bit is a copy of the TTHRES bit located in the HxISTAT0 register (see page 2-16).
8	TDATA1	**One Transmit Byte Space Available (Same as HxRD)** This bit is a copy of the TDATA1 bit located in the HxISTAT0 register (see page 2-16).
7–0	FBCNT[7–0]	**Frame Byte Count [7–0]** This bit field contains the least significant byte of the length of the received message or frame.

Programming Notes

When receiving a frame, the HDLC Receive FIFO contains the data. Reading this register increments to the next byte of data. After all the data for the frame has been read (see the HxRD register description on page 2-25), this register contains the three bytes of status for the frame.

HDLC Channel A Receive Frame Status 2 (HARFS2) — Same as HARD
HDLC Channel B Receive Frame Status 2 (HBRFS2) — Same as HBRD
HDLC Channel C Receive Frame Status 2 (HCRFS2) — Same as HCRD
HDLC Channel D Receive Frame Status 2 (HDRFS2) — Same as HDRD

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	STAT1A	STAT0A	STATNUM		RTHRES	RDATA1	TTHRES	TDATA1	FBCNT[15–8]							
Software Read/Write	R	R	R		R	R	R	R	R							
Hardware Set/Clear	S/C	S/C	S/C		S/C	S/C	S/C	S/C	S/C							
Chip Reset Default	0	0	0	0	0	0	0	0	x	x	x	x	x	x	x	x

Microcontroller:	Channel: A	B	C	D
Am186CC	✓	✓	✓	✓
Am186CH	✓	✓		
Am186CU				

Register Description

After the last frame data byte is read from the HDLC Receive FIFO Data register (see page 2-25), the next three bytes read contain frame status information. Receive Frame Status 2 is the second byte of status information.

Bit Definitions

Bit	Name	Function
15	STAT1A	**Any HxISTAT1 Register Interrupt (Same as HxRD)** This bit represents the logical OR of all the unmasked interrupt bits in the HxISTAT1 register (see page 2-20).
14	STAT0A	**Any HxISTAT0 Register Interrupt (Same as HxRD)** This bit represents the logical OR of all the unmasked interrupt bits in the HxISTAT0 register (see page 2-16). It does not include the REOF, RTHRES, RDATA1, TTHRES, or TDATA1 bits because these can be inferred by other bits available in this register.
13–12	STATNUM	**Status Byte Number (Same as HxRD)** These two bits indicate which byte of status is presently available in the receive FIFO (i.e., 1–3). If the byte is data, these bits are both 0. 00 = The RDATA bit field is data (see page 2-25). 01 = The RDATA bit field contains Receive Frame Status 1 (see page 2-29). 10 = The RDATA bit field contains Receive Frame Status 2 (use this register description). 11 = The RDATA bit field contains Receive Frame Status 3 (see page 2-33).
11	RTHRES	**Receive FIFO Threshold Reached (Same as HxRD)** This bit is a copy of the RTHRES bit located in the HxISTAT0 register (see page 2-16).
10	RDATA1	**One Receive Data Byte Available (Same as HxRD)** This bit is a copy of the RDATA1 bit located in the HxISTAT0 register (see page 2-16).
9	TTHRES	**Transmit FIFO Threshold Reached (Same as HxRD)** This bit is a copy of the TTHRES bit located in the HxISTAT0 register (see page 2-16).
8	TDATA1	**One Transmit Byte Space Available (Same as HxRD)** This bit is a copy of the TDATA1 bit located in the HxISTAT0 register (see page 2-16).
7–0	FBCNT[15–8]	**Frame Byte Count [15–8]** If RCPST is 0 (see page 2-11) this bit field contains the most significant byte of the length of the received message or frame. If RCPST is 1 (see page 2-11), bits 7–5 (FBCNT[15–13]) are replaced by the FABORT, FLONG, and FSHORT bits, respectively (see page 2-34).

Programming Notes

When receiving a frame, the HDLC Receive FIFO contains the data. Reading this register increments to the next byte of data. After all the data for the frame has been read (see the HxRD register description on page 2-25), this register contains the three bytes of status for the frame.

HDLC Channel A Receive Frame Status 3 (HARFS3) Same as HARD
HDLC Channel B Receive Frame Status 3 (HBRFS3) Same as HBRD
HDLC Channel C Receive Frame Status 3 (HCRFS3) Same as HCRD
HDLC Channel D Receive Frame Status 3 (HDRFS3) Same as HDRD

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	STAT1A	STAT0A	STATNUM		RTHRES	RDATA1	TTHRES	TDATA1	FRAM	FOFLO	CRCE	MTCH		FABORT	FLONG	FSHORT
Software Read/Write	R	R	R		R	R	R	R	R	R	R	R		R	R	R
Hardware Set/Clear	S/C	S/C	S/C		S/C	S/C	S/C	S/C	S/C	S/C	S/C	S/C		S/C	S/C	S/C
Chip Reset Default	0	0	0	0	0	0	0	0	x	x	x	x	x	x	x	x

Microcontroller:	Channel: A	B	C	D
Am186CC	✓	✓	✓	✓
Am186CH	✓	✓		
Am186CU				

Register Description

After the last frame data byte is read from the HDLC Receive FIFO Data register (see page 2-25), the next three bytes contain frame status information. Receive Frame Status 3 is the third byte of status information.

Bit Definitions

Bit	Name	Function
15	STAT1A	**Any HxISTAT1 Register Interrupt (Same as HxRD)** This bit represents the logical OR of all the unmasked interrupt bits in the HxISTAT1 register (see page 2-20).
14	STAT0A	**Any HxISTAT0 Register Interrupt (Same as HxRD)** This bit represents the logical OR of all the unmasked interrupt bits in the HxISTAT0 register (see page 2-16). It does not include the REOF, RTHRES, RDATA1, TTHRES, or TDATA1 bits because these can be inferred by other bits available in this register.
13–12	STATNUM	**Status Byte Number (Same as HxRD)** These two bits indicate which byte of status is presently available in the receive FIFO (i.e., 1–3). If the byte is data, these bits are both 0. 00 = The RDATA bit field is data (see page 2-25). 01 = The RDATA bit field contains Receive Frame Status 1 (see page 2-29). 10 = The RDATA bit field contains Receive Frame Status 2 (see page 2-31). 11 = The RDATA bit field contains Receive Frame Status 3 (use this register description).
11	RTHRES	**Receive FIFO Threshold Reached (Same as HxRD)** This bit is a copy of the RTHRES bit located in the HxISTAT0 register (see page 2-16).
10	RDATA1	**One Receive Data Byte Available (Same as HxRD)** This bit is a copy of the RDATA1 bit located in the HxISTAT0 register (see page 2-16).
9	TTHRES	**Transmit FIFO Threshold Reached (Same as HxRD)** This bit is a copy of the TTHRES bit located in the HxISTAT0 register (see page 2-16).
8	TDATA1	**One Transmit Byte Space Available (Same as HxRD)** This bit is a copy of the TDATA1 bit located in the HxISTAT0 register (see page 2-16).

Bit	Name	Function
7	FRAM	**Framing Error** 0 = The frame was an even multiple of eight. 1 = The number of bits in the frame was not a multiple of eight and the last received byte might be corrupted. This bit is not set for an FABORT, FOFLO, or FLONG bit status.
6	FOFLO	**Frame Overflow** 0 = The receive FIFO did not overflow. 1 = The receive FIFO overflowed during reception. When set, this is the only active bit for this byte.
5	CRCE	**Cyclic Redundancy Check (CRC) Error** 0 = The CRC was correct. 1 = A CRC error was detected. This bit is not set for an FABORT, FOFLO, or FLONG bit status.
4–3	MTCH	**Match Address Register** The MTCH bit field indicates which address register had a match that allowed this frame to be accepted. If multiple address registers match, the lowest matching register is indicated. 00 = HxA0 01 = HxA1 10 = HxA2 11 = HxA3 This bit field is not valid for an FLONG or FOFLO bit status.
2	FABORT	**Frame Abort** 0 = A normal closing flag was seen and the RREJECT and RSTOP bits were not asserted (see page 2-12). 1 = An abort was seen before the closing flag, or the RREJECT bit was asserted during reception, or the RSTOP bit was asserted during reception.
1	FLONG	**Frame Too Long** 0 = The frame was less than or equal to the maximum frame length (MAXRL bit field, see page 2-13). 1 = The frame exceeded the maximum frame length. When set, this is the only active bit for this byte. The FLONG bit has precedence over the FSHORT bit.
0	FSHORT	**Frame Too Short** 0 = The frame was equal to or longer than the minimum frame length (MINRL bit field, see page 2-12). 1 = The frame was not as long as the minimum frame length, but had an address match and was large enough that part of it was transferred to the receive FIFO. This bit is not set if the frame is too short as a result of an aborted or truncated frame.

Programming Notes

When receiving a frame, the HDLC Receive FIFO contains the data. Reading this register increments to the next byte of data. After all the data for the frame has been read (see the HxRD register description on page 2-25), this register contains the three bytes of status for the frame.

In addition to containing status for the frame, the third status byte also indicates which address was matched (0–3) for this frame.

HDLC Channel A Short Frame Counter (HASFCNT) Offset 1Ah
HDLC Channel B Short Frame Counter (HBSFCNT) Offset 5Ah
HDLC Channel C Short Frame Counter (HCSFCNT) Offset 9Ah
HDLC Channel D Short Frame Counter (HDSFCNT) Offset DAh

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	HSFCNT															
Software Read/Write	R															
Hardware Set/Clear	S/C															
Chip Reset Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Microcontroller:	Channel: A	B	C	D
Am186CC	✓	✓	✓	✓
Am186CH	✓	✓		
Am186CU				

Register Description

This register contains the count of short frames discarded. Reading this register resets the count to zero. This register is also cleared after any external or watchdog reset, or after an HRESET bit event (see page 2-5).

Bit Definitions

Bit	Name	Function
15–0	HSFCNT	**Short Frame Counter** This is the count of the number of frames that were smaller than the minimum receive length (MINRL bit field, see page 2-12). This includes very short frames (frames shorter than two bytes), which are discarded without going to memory. This count does not include short frames that have mismatched addresses. Mismatched addresses are counted by the HxMACNT register (see page 2-37).

Programming Notes

HDLC Channel A Short Frame Counter Peek (HASFCNTP) Offset 1Ch
HDLC Channel B Short Frame Counter Peek (HBSFCNTP) Offset 5Ch
HDLC Channel C Short Frame Counter Peek (HCSFCNTP) Offset 9Ch
HDLC Channel D Short Frame Counter Peek (HDSFCNTP) Offset DCh

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	HSFCNTP															
Software Read/Write	R															
Hardware Set/Clear	S/C															
Chip Reset Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Microcontroller:	Channel: A	B	C	D
Am186CC	✓	✓	✓	✓
Am186CH	✓	✓		
Am186CU				

Register Description

This is a copy of the HDLC Short Frame Counter register (see page 2-35) that does not change when read.

Bit Definitions

Bit	Name	Function
15–0	HSFCNTP	**Short Frame Counter (Same as HxSFCNT)** This is the count of the number of frames that were smaller than the minimum receive length (MINRL bit field, see page 2-12). This includes very short frames (frames shorter than two bytes), which are discarded without going to memory. This count does not include short frames that have mismatched addresses. Mismatched addresses are counted by the HxMACNT register (see page 2-37).

Programming Notes

HDLC Channel A Mismatch Address Counter (HAMACNT) Offset 1Eh
HDLC Channel B Mismatch Address Counter (HBMACNT) Offset 5Eh
HDLC Channel C Mismatch Address Counter (HCMACNT) Offset 9Eh
HDLC Channel D Mismatch Address Counter (HDMACNT) Offset DEh

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	HMACNT															
Software Read/Write	R															
Hardware Set/Clear	S/C															
Chip Reset Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Microcontroller:	Channel: A	B	C	D
Am186CC	✓	✓	✓	✓
Am186CH	✓	✓		
Am186CU				

Register Description

This register contains the count of mismatched address frames discarded. Reading this register resets the count to zero. This register is also cleared after any external or watchdog reset, or after an HRESET bit event (see page 2-5).

Bit Definitions

Bit	Name	Function
15–0	HMACNT	**Mismatch Address Counter** This is the count of the number of frames that were discarded because they did not match any of the address registers. A frame that is shorter than the minimum receive length and does not have an address match is counted as an address mismatch. This count does not include very short frames (frames shorter than two bytes), which are discarded without going into memory. Very short frames are counted by the HxSFCNT register (see page 2-35).

Programming Notes

HDLC Channel A Mismatch Address Counter Peek (HAMACNTP) Offset 20h
HDLC Channel B Mismatch Address Counter Peek (HBMACNTP) Offset 60h
HDLC Channel C Mismatch Address Counter Peek (HCMACNTP) Offset A0h
HDLC Channel D Mismatch Address Counter Peek (HDMACNTP) Offset E0h

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	HMACNTP															
Software Read/Write	R															
Hardware Set/Clear	S/C															
Chip Reset Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Microcontroller:	Channel: A	B	C	D
Am186CC	✓	✓	✓	✓
Am186CH	✓	✓		
Am186CU				

Register Description

This is a copy of the HDLC Mismatch Address Counter register (see page 2-37) that does not change when read.

Bit Definitions

Bit	Name	Function
15–0	HMACNTP	**Mismatch Address Counter (Same as HxMACNT)** This is the count of the number of frames that were discarded because they did not match any of the address registers. A frame that is shorter than the minimum receive length and does not have an address match is counted as an address mismatch. This count does not include very short frames (frames shorter than two bytes), which are discarded without going into memory. Very short frames are counted by the HxSFCNT register (see page 2-35).

Programming Notes

HDLC Channel A Address 0 (HAA0) Offset 22h
HDLC Channel B Address 0 (HBA0) Offset 62h
HDLC Channel C Address 0 (HCA0) Offset A2h
HDLC Channel D Address 0 (HDA0) Offset E2h

HDLC Channel A Address 1 (HAA1) Offset 26h
HDLC Channel B Address 1 (HBA1) Offset 66h
HDLC Channel C Address 1 (HCA1) Offset A6h
HDLC Channel D Address 1 (HDA1) Offset E6h

HDLC Channel A Address 2 (HAA2) Offset 2Ah
HDLC Channel B Address 2 (HBA2) Offset 6Ah
HDLC Channel C Address 2 (HCA2) Offset AAh
HDLC Channel D Address 2 (HDA2) Offset EAh

HDLC Channel A Address 3 (HAA3) Offset 2Eh
HDLC Channel B Address 3 (HBA3) Offset 6Eh
HDLC Channel C Address 3 (HCA3) Offset AEh
HDLC Channel D Address 3 (HDA3) Offset EEh

<table>
<tr><th></th><th>15</th><th>14</th><th>13</th><th>12</th><th>11</th><th>10</th><th>9</th><th>8</th><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td>Bit Name</td><td colspan="16">HA</td></tr>
<tr><td>Software Read/Write</td><td colspan="16">R/W</td></tr>
<tr><td>Hardware Set/Clear</td><td colspan="16">—</td></tr>
<tr><td>Chip Reset Default</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
</table>

Microcontroller:	Channel: A	B	C	D
Am186CC	✓	✓	✓	✓
Am186CH	✓	✓		
Am186CU				

Register Description

Four address registers are provided for each HDLC channel. Each register contains a value to compare to the address in the received frames. If the corresponding bit in the corresponding address mask register is cleared, the bit is masked and that bit is always matched. If there are multiple address registers (with their corresponding address mask register) that match, only the lowest matching address register is reported as a match.

Bit Definitions

Bit	Name	Function
15–0	HA	**Address Compare Value** This value is compared to the address in received frames. Bits 7–0 are the first byte received and bits 15–8 are the second byte received. If the RMSBF bit is 0 (see page 2-11), bit 0 is the first bit received after the opening flag. If the RMSBF bit is 1, bit 8 is the first bit received after the opening flag.

Programming Notes

HDLC Channel A Address Mask 0 (HAA0MSK) Offset 24h
HDLC Channel B Address Mask 0 (HBA0MSK) Offset 64h
HDLC Channel C Address Mask 0 (HCA0MSK) Offset A4h
HDLC Channel D Address Mask 0 (HDA0MSK) Offset E4h

HDLC Channel A Address Mask 1 (HAA1MSK) Offset 28h
HDLC Channel B Address Mask 1 (HBA1MSK) Offset 68h
HDLC Channel C Address Mask 1 (HCA1MSK) Offset A8h
HDLC Channel D Address Mask 1 (HDA1MSK) Offset E8h

HDLC Channel A Address Mask 2 (HAA2MSK) Offset 2Ch
HDLC Channel B Address Mask 2 (HBA2MSK) Offset 6Ch
HDLC Channel C Address Mask 2 (HCA2MSK) Offset ACh
HDLC Channel D Address Mask 2 (HDA2MSK) Offset ECh

HDLC Channel A Address Mask 3 (HAA3MSK) Offset 30h
HDLC Channel B Address Mask 3 (HBA3MSK) Offset 70h
HDLC Channel C Address Mask 3 (HCA3MSK) Offset B0h
HDLC Channel D Address Mask 3 (HDA3MSK) Offset F0h

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	HAMSK															
Software Read/Write	R/W															
Hardware Set/Clear	—															
Chip Reset Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Microcontroller:	Channel: A	B	C	D
Am186CC	✓	✓	✓	✓
Am186CH	✓	✓		
Am186CU				

Register Description

These are the mask registers for address comparison. Each address mask register corresponds to one of the four address registers that are provided for each HDLC channel (see page 2-39).

Bit Definitions

Bit	Name	Function
15–0	HAMSK	**Address Mask** These are the bits to be ignored in the address when doing address comparisons. If an address mask bit is 0, the corresponding bit in the corresponding address register is ignored. Therefore, the default condition is to accept all frames.

Programming Notes

CHAPTER

3 DMA REGISTERS

3.1 OVERVIEW

This chapter describes the general-purpose Direct Memory Access (DMA) and SmartDMA controller registers on the Am186CC/CH/CU microcontrollers.

The Am186CC microcontroller provides a total of 12 DMA channels. The Am186CH and Am186CU microcontrollers provide eight DMA channels.

On any of these controllers, four of the DMA channels are for general-purpose memory-to-memory, memory-to-I/O, I/O-to-memory, or I/O-to-I/O transfers, as well as access to the Asynchronous Serial Ports or, where supported, the Universal Serial Bus (USB).

The remaining DMA channels are SmartDMA channels that provide a sophisticated buffer-chaining mechanism. SmartDMA channels are always used in transmit and receive pairs.

The Am186CC microcontroller provides four SmartDMA channel pairs (eight channels). Two SmartDMA channel pairs are dedicated for use with HDLC channels A and B. The remaining two SmartDMA channel pairs on the Am186CC microcontroller can be used with HDLC channels C and D or with Universal Serial Bus (USB) endpoints A, B, C, and D.

The Am186CH and Am186CU microcontrollers each provide two SmartDMA channel pairs (four channels). On the Am186CH HDLC microcontroller, the two SmartDMA channel pairs can be used with HDLC channels A and B. On the Am186CU USB microcontroller, the two SmartDMA channel pairs can be used with USB endpoints A, B, C, and D.

For more information about using DMA, see the *Am186CC/CH/CU Microcontrollers User's Manual*, order #21914.

This chapter describes all the general-purpose DMA registers first, then all the SmartDMA controller registers. Within each of these types, the register sets are identical for each of the four channels (or SmartDMA channel pairs). This chapter describes registers with identical functions only once. The unique register names and offsets that apply are listed at the top of each register page. Table 3-1 lists the DMA registers in offset order, with the corresponding description's page number.

Table 3-1 DMA Register Map

Register Name	Mnemonic	Offset	Page Number
General-Purpose DMA Channel 0 Registers			
General-Purpose DMA 0 Control 0	GD0CON0	100h	page 3-4
General-Purpose DMA 0 Control 1	GD0CON1	102h	page 3-7
General-Purpose DMA 0 Source Address Low	GD0SRCL	104h	page 3-9
General-Purpose DMA 0 Source Address High	GD0SRCH	106h	page 3-10
General-Purpose DMA 0 Destination Address Low	GD0DSTL	108h	page 3-11
General-Purpose DMA 0 Destination Address High	GD0DSTH	10Ah	page 3-12
General-Purpose DMA 0 Transfer Count	GD0TC	10Ch	page 3-13

Table 3-1 DMA Register Map (Continued)

Register Name	Mnemonic	Offset	Page Number
General-Purpose DMA Channel 1 Registers			
General-Purpose DMA 1 Control 0	GD1CON0	110h	page 3-4
General-Purpose DMA 1 Control 1	GD1CON1	112h	page 3-7
General-Purpose DMA 1 Source Address Low	GD1SRCL	114h	page 3-9
General-Purpose DMA 1 Source Address High	GD1SRCH	116h	page 3-10
General-Purpose DMA 1 Destination Address Low	GD1DSTL	118h	page 3-11
General-Purpose DMA 1 Destination Address High	GD1DSTH	11Ah	page 3-12
General-Purpose DMA 1 Transfer Count	GD1TC	11Ch	page 3-13
General-Purpose DMA Channel 2 Registers			
General-Purpose DMA 2 Control 0	GD2CON0	120h	page 3-4
General-Purpose DMA 2 Control 1	GD2CON1	122h	page 3-7
General-Purpose DMA 2 Source Address Low	GD2SRCL	124h	page 3-9
General-Purpose DMA 2 Source Address High	GD2SRCH	126h	page 3-10
General-Purpose DMA 2 Destination Address Low	GD2DSTL	128h	page 3-11
General-Purpose DMA 2 Destination Address High	GD2DSTH	12Ah	page 3-12
General-Purpose DMA 2 Transfer Count	GD2TC	12Ch	page 3-13
General-Purpose DMA Channel 3 Registers			
General-Purpose DMA 3 Control 0	GD3CON0	130h	page 3-4
General-Purpose DMA 3 Control 1	GD3CON1	132h	page 3-7
General-Purpose DMA 3 Source Address Low	GD3SRCL	134h	page 3-9
General-Purpose DMA 3 Source Address High	GD3SRCH	136h	page 3-10
General-Purpose DMA 3 Destination Address Low	GD3DSTL	138h	page 3-11
General-Purpose DMA 3 Destination Address High	GD3DSTH	13Ah	page 3-12
General-Purpose DMA 3 Transfer Count	GD3TC	13Ch	page 3-13
SmartDMA Channel Pair 0 Registers (Am186CC and Am186CH HDLC Microcontrollers Only)			
SmartDMA Channel Pair 0 Control	SD0CON	140h	page 3-14
SmartDMA Channel Pair 0 Transmit Ring Count/ Address Low	SD0TRCAL	142h	page 3-20
SmartDMA Channel Pair 0 Transmit Ring Address High	SD0TRAH	144h	page 3-21
SmartDMA Channel Pair 0 Receive Ring Count/ Address Low	SD0RRCAL	146h	page 3-22
SmartDMA Channel Pair 0 Receive Ring Address High	SD0RRAH	148h	page 3-23
SmartDMA Channel Pair 0 Status	SD0STAT	14Ah	page 3-24
SmartDMA Channel Pair 0 Current Buffer Descriptor	SD0CBD	14Ch	page 3-26
SmartDMA Channel Pair 0 Current Transmit Address	SD0CTAD	14Eh	page 3-27
SmartDMA Channel Pair 0 Current Receive Address	SD0CRAD	150h	page 3-28
SmartDMA Channel Pair 1 Registers (Am186CC and Am186CH HDLC Microcontrollers Only)			
SmartDMA Channel Pair 1 Control	SD1CON	158h	page 3-14
SmartDMA Channel Pair 1 Transmit Ring Count/ Address Low	SD1TRCAL	15Ah	page 3-20
SmartDMA Channel Pair 1 Transmit Ring Address High	SD1TRAH	15Ch	page 3-21

Table 3-1 DMA Register Map (Continued)

Register Name	Mnemonic	Offset	Page Number
SmartDMA Channel Pair 1 Receive Ring Count/ Address Low	SD1RRCAL	15Eh	page 3-22
SmartDMA Channel Pair 1 Receive Ring Address High	SD1RRAH	160h	page 3-23
SmartDMA Channel Pair 1 Status	SD1STAT	162h	page 3-24
SmartDMA Channel Pair 1 Current Buffer Descriptor	SD1CBD	164h	page 3-26
SmartDMA Channel Pair 1 Current Transmit Address	SD1CTAD	166h	page 3-27
SmartDMA Channel Pair 1 Current Receive Address	SD1CRAD	168h	page 3-28
SmartDMA Channel Pair 2 Registers (Am186CC and Am186CU USB Microcontrollers Only)			
SmartDMA Channel Pair 2 Control	SD2CON	170h	page 3-17
SmartDMA Channel Pair 2 Transmit Ring Count/ Address Low	SD2TRCAL	172h	page 3-20
SmartDMA Channel Pair 2 Transmit Ring Address High	SD2TRAH	174h	page 3-21
SmartDMA Channel Pair 2 Receive Ring Count/ Address Low	SD2RRCAL	176h	page 3-22
SmartDMA Channel Pair 2 Receive Ring Address High	SD2RRAH	178h	page 3-23
SmartDMA Channel Pair 2 Status	SD2STAT	17Ah	page 3-24
SmartDMA Channel Pair 2 Current Buffer Descriptor	SD2CBD	17Ch	page 3-26
SmartDMA Channel Pair 2 Current Transmit Address	SD2CTAD	17Eh	page 3-27
SmartDMA Channel Pair 2 Current Receive Address	SD2CRAD	180h	page 3-28
SmartDMA Channel Pair 3 Registers (Am186CC and Am186CU USB Microcontrollers Only)			
SmartDMA Channel Pair 3 Control	SD3CON	188h	page 3-17
SmartDMA Channel Pair 3 Transmit Ring Count/ Address Low	SD3TRCAL	18Ah	page 3-20
SmartDMA Channel Pair 3 Transmit Ring Address High	SD3TRAH	18Ch	page 3-21
SmartDMA Channel Pair 3 Receive Ring Count/ Address Low	SD3RRCAL	18Eh	page 3-22
SmartDMA Channel Pair 3 Receive Ring Address High	SD3RRAH	190h	page 3-23
SmartDMA Channel Pair 3 Status	SD3STAT	192h	page 3-24
SmartDMA Channel Pair 3 Current Buffer Descriptor	SD3CBD	194h	page 3-26
SmartDMA Channel Pair 3 Current Transmit Address	SD3CTAD	196h	page 3-27
SmartDMA Channel Pair 3 Current Receive Address	SD3CRAD	198h	page 3-28

General-Purpose DMA 0 Control 0 (GD0CON0) Offset 100h
General-Purpose DMA 1 Control 0 (GD1CON0) Offset 110h
General-Purpose DMA 2 Control 0 (GD2CON0) Offset 120h
General-Purpose DMA 3 Control 0 (GD3CON0) Offset 130h

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	ST	AST	TC	INT	*Res*		P		*Res*	TS	*Res*	DSEL				
Software Read/Write	R/W	R/W	R/W	R/W	R		R/W		R	R/W	R	R/W				
Hardware Set/Clear	S/C	—	—	—	—		—		—	—	—	—				
Chip Reset Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Microcontroller:	
Am186CC	✔
Am186CH	!
Am186CU	✔

! *= See bit definitions.*

Register Description

Along with GDxCON1 (see page 3-7), this register is used to set up a general-purpose DMA channel.

Bit Definitions

Bit **Name** **Function**

15 ST **Start/Stop DMA Channel**

0 = Stop DMA transfer.

1 = Start DMA transfer.

This bit is cleared by hardware when the associated GDxTC transfer count register (see page 3-13) reaches 0 and the TC bit is set for a synchronized request. For an unsynchronized request, this bit is cleared by hardware when the terminal count reaches 0. If the AST bit is set, hardware sets the ST bit whenever the Transfer Count register is reloaded.

14 AST **Auto Start**

0 = Disable Auto Start function.

1 = The DMA channel sets the ST bit and starts the transfer whenever the associated GDxTC transfer count register (see page 3-13) register is reloaded.

13 TC **Terminal Count**

0 = DMA transfer does not stop when the associated GDxTC transfer count register (see page 3-13) reaches 0 for a synchronized request. For an unsynchronized request, DMA always stops when the transfer count reaches 0.

1 = DMA transfer stops when the associated transfer count register reaches 0.

12 INT **Interrupt**

0 = DMA does not generate an interrupt request on the completion of the transfer count.

1 = DMA generates an interrupt request on the completion of the transfer count.

The general-purpose DMA channel interrupt status is indicated by the DMA3–DMA0 bits in the INTSTS register (see page 9-42).

In addition to setting the INT bit in this register (GDxCON0), software must configure the interrupt Channel 9–10 Control registers correctly. The appropriate register must be configured to select each required interrupt channel’s internal source and to enable the interrupt channel. Interrupt channel 9 serves DMA channels 0 and 1. Interrupt channel 10 serves DMA channels 2 and 3. See Chapter 9, “Interrupt Controller Registers”.

11–10 Res **Reserved**

For compatibility with future devices, always write this bit field with its chip reset default value.

Bit	Name	Function
9–8	P	**Relative Priority** These two bits indicate the priority of this channel relative to other channels during simultaneous transfers. 00 = Low priority 01 = Medium priority 10 = High priority 11 = Reserved
7	Res	**Reserved** For compatibility with future devices, always write this bit field with its chip reset default value.
6	TS	**Transfer Size** 0 = Select byte transfer. 1 = Select word transfer. The TS bit setting affects the valid settings of the SINC and DINC bit fields in the GDxCON1 register. See page 3-7 and page 3-8.
5	Res	**Reserved** For compatibility with future devices, always write this bit field with its chip reset default value.
4–0	DSEL	**DMA Request Select** The DSEL bit selects the DMA request source and synchronization for the channel. 00000 = Unsynchronized 00001 = Reserved 00010 = Timer 2, source synchronized 00011 = Reserved 00100 = Reserved 00101 = Reserved 00110 = Reserved 00111 = Reserved 01000 = External DRQ0, source synchronized 01001 = External DRQ0, destination synchronized 01010 = External DRQ1, source synchronized 01011 = External DRQ1, destination synchronized 01100 = Reserved 01101 = Reserved 01110 = Reserved 01111 = Reserved 10000 = UART receiver, source synchronized 10001 = UART transmitter, destination synchronized 10010 = High-Speed UART receiver, source synchronized 10011 = High-Speed UART transmitter, destination synchronized 10100 = Reserved 10101 = Reserved 10110 = Reserved 10111 = Reserved 11000 = USB Endpoint A receiver, source synchronized (USB OUT endpoint) 11001 = USB Endpoint A transmitter, destination synchronized (USB IN endpoint) 11010 = USB Endpoint B receiver, source synchronized 11011 = USB Endpoint B transmitter, destination synchronized 11100 = USB Endpoint C receiver, source synchronized 11101 = USB Endpoint C transmitter, destination synchronized 11110 = USB Endpoint D receiver, source synchronized 11111 = USB Endpoint D transmitter, destination synchronized ***Am186CH Microcontroller:*** *Values 11000b–11111b are reserved on the Am186CH HDLC microcontroller.*

Programming Notes

General-Purpose DMA 0 Control 1 (GD0CON1) Offset 102h
General-Purpose DMA 1 Control 1 (GD1CON1) Offset 112h
General-Purpose DMA 2 Control 1 (GD2CON1) Offset 122h
General-Purpose DMA 3 Control 1 (GD3CON1) Offset 132h

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	SM/$\overline{\text{IO}}$	SAW			SINC				DM/$\overline{\text{IO}}$	DAW			DINC			
Software Read/Write	R/W	R/W			R/W				R/W	R/W			R/W			
Hardware Set/Clear	—	—			—				—	—			—			
Chip Reset Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Microcontroller:

Am186CC ✔

Am186CH ✔

Am186CU ✔

Register Description

Along with GDxCON0 (see page 3-4), this register is used to set up a general-purpose DMA channel.

Bit Definitions

Bit	Name	Function
15	SM/$\overline{\text{IO}}$	**Source Address Space Select** 0 = The source address is in I/O space. 1 = The source address is in memory space.
14–12	SAW	**Source Address Wrap** 000 = Normal operation; no wrap. 001 = 1K circular buffer on a 1K boundary. 010 = 2K circular buffer on a 2K boundary. 011 = 4K circular buffer on a 4K boundary. 100 = 8K circular buffer on an 8K boundary. 101 = 16K circular buffer on a 16K boundary. 110 = 32K circular buffer on a 32K boundary. 111 = 64K circular buffer on a 64K boundary.
11–8	SINC	**Source Increment (2's complement)** 0000 = No increment or decrement. 0001 = Increment by 1 byte (not valid for word transfers). 0010 = Increment by 2 bytes. 1111 = Decrement by 1 byte (not valid for word transfers). 1110 = Decrement by 2 bytes. All other values are reserved. The 0001b and 1111b values of the SINC bit field are invalid if word transfers are selected by setting the TS bit to 1 in the GDxCON0 register (see page 3-5).
7	DM/$\overline{\text{IO}}$	**Destination Address Space Select** 0 = The destination address is in I/O space. 1 = The destination address is in memory space.

Bit	Name	Function
6–4	DAW	**Destination Address Wrap** 000 = Normal operation; no wrap. 001 = 1K circular buffer on a 1K boundary. 010 = 2K circular buffer on a 2K boundary. 011 = 4K circular buffer on a 4K boundary. 100 = 8K circular buffer on an 8K boundary. 101 = 16K circular buffer on a 16K boundary. 110 = 32K circular buffer on a 32K boundary. 111 = 64K circular buffer on a 64K boundary.
3–0	DINC	**Destination Increment (2's complement)** 0000 = No increment or decrement. 0001 = Increment by 1 byte (not valid for word transfers). 0010 = Increment by 2 bytes. 1111 = Decrement by 1 byte (not valid for word transfers). 1110 = Decrement by 2 bytes. All other values are reserved. The 0001b and 1111b values of the DINC bit field are invalid if word transfers are selected by setting the TS bit to 1 in the GDxCON0 register (see page 3-5).

Programming Notes

Software must stop DMA operation before writing to this register; otherwise, the results are unpredictable.

General-Purpose DMA 0 Source Address Low (GD0SRCL) Offset 104h
General-Purpose DMA 1 Source Address Low (GD1SRCL) Offset 114h
General-Purpose DMA 2 Source Address Low (GD2SRCL) Offset 124h
General-Purpose DMA 3 Source Address Low (GD3SRCL) Offset 134h

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	DSA[15–0]															
Software Read/Write	R/W															
Hardware Set/Clear	S/C															
Chip Reset Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Microcontroller:	
Am186CC	✔
Am186CH	✔
Am186CU	✔

Register Description

This register is used to set the source address bits [15–0] for a general-purpose DMA channel. (Bits [19–16] are set in the GDxSRCH register, see page 3-10.)

Bit Definitions

Bit	Name	Function
15–0	DSA[15–0]	**DMA Source Address Low** Source Address bits [15–0]. This register and the Source Address High bits DSA[19–16] (see page 3-10) are updated after each transfer.

Programming Notes

General-Purpose DMA 0 Source Address High (GD0SRCH) Offset 106h
General-Purpose DMA 1 Source Address High (GD1SRCH) Offset 116h
General-Purpose DMA 2 Source Address High (GD2SRCH) Offset 126h
General-Purpose DMA 3 Source Address High (GD3SRCH) Offset 136h

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	*Res*												DSA[19–16]			
Software Read/Write	R												R/W			
Hardware Set/Clear	—												S/C			
Chip Reset Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Microcontroller:

Am186CC	✓
Am186CH	✓
Am186CU	✓

Register Description

This register is used to set the source address bits [19–16] for a general-purpose DMA channel. (Bits [15–0] are set in the GDxSRCL register, see page 3-9.)

Bit Definitions

Bit	Name	Function
15–4	Res	**Reserved** For compatibility with future devices, always write this bit field with its chip reset default value.
3–0	DSA[19–16]	**DMA Source Address High** Source Address bits [19–16]. This register and the Source Address Low bits DSA[15–0] (see page 3-9) are updated after each transfer.

Programming Notes

General-Purpose DMA 0 Destination Address Low (GD0DSTL) Offset 108h

General-Purpose DMA 1 Destination Address Low (GD1DSTL) Offset 118h

General-Purpose DMA 2 Destination Address Low (GD2DSTL) Offset 128h

General-Purpose DMA 3 Destination Address Low (GD3DSTL) Offset 138h

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	DDA[15–0]															
Software Read/Write	R/W															
Hardware Set/Clear	S/C															
Chip Reset Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Microcontroller:	
Am186CC	✓
Am186CH	✓
Am186CU	✓

Register Description

This register is used to set the destination address bits [15–0] for a general-purpose DMA channel. (Bits [19–16] are set in the GDxDSTH register, see page 3-12.)

Bit Definitions

Bit	Name	Function
15–0	DDA[15–0]	**DMA Destination Address Low** Destination Address bits [15–0]. This register and the Destination Address High bits DDA[19–16] (see page 3-12) are updated after each transfer.

Programming Notes

General-Purpose DMA 0 Destination Address High (GD0DSTH) Offset 10Ah
General-Purpose DMA 1 Destination Address High (GD1DSTH) Offset 11Ah
General-Purpose DMA 2 Destination Address High (GD2DSTH) Offset 12Ah
General-Purpose DMA 3 Destination Address High (GD3DSTH) Offset 13Ah

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	*Res*												DDA[19–16]			
Software Read/Write	R												R/W			
Hardware Set/Clear	—												S/C			
Chip Reset Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Microcontroller:
Am186CC ✔
Am186CH ✔
Am186CU ✔

Register Description

This register is used to set the destination address bits [19–16] for a general-purpose DMA channel. (Bits [15–0] are set in the GDxDSTL register, see page 3-11.)

Bit Definitions

Bit	Name	Function
15–4	Res	**Reserved** For compatibility with future devices, always write this bit field with its chip reset default value.
3–0	DDA[19–16]	**DMA Destination Address High** Destination Address bits [19–16]. This register and the Destination Address Low bits DDA[15–0] (see page 3-11) are updated after each transfer.

Programming Notes

General-Purpose DMA 0 Transfer Count (GD0TC) Offset 10Ch
General-Purpose DMA 1 Transfer Count (GD1TC) Offset 11Ch
General-Purpose DMA 2 Transfer Count (GD2TC) Offset 12Ch
General-Purpose DMA 3 Transfer Count (GD3TC) Offset 13Ch

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	TC															
Software Read/Write	R/W															
Hardware Set/Clear	S/C															
Chip Reset Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Microcontroller:
Am186CC ✓
Am186CH ✓
Am186CU ✓

Register Description

This register is used to set the transfer count for a general-purpose DMA channel.

Bit Definitions

Bit	Name	Function
15–0	TC	**DMA Transfer Count** The TC bit field contains the transfer count for this channel. The value is decremented by one after each transfer. If the TC bit in the GDxCON0 register is set to 1 (see page 3-4) or if unsynchronized transfers are programmed, DMA activity terminates when the transfer count value reaches 0.

Programming Notes

SmartDMA Channel Pair 0 Control (SD0CON) Offset 140h

SmartDMA Channel Pair 1 Control (SD1CON) Offset 158h

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	*Res*		TEPI	TBUI	TTCI	REPI	RBUI	RTCI	TXSO	RXSO	P		POLL	*Res*	TXST	RXST
Software Read/Write	R		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		R/W	R	R/W	R/W
Hardware Set/Clear	—		—	—	—	—	—	—	—	—	—		C	—	—	—
Chip Reset Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Microcontroller:
Am186CC ☑
Am186CH ☑
Am186CU □

Register Description

This register is used to set up SmartDMA channel pair 0 or 1. Except for bit 2, the descriptions are the same as for channel pairs 2 and 3 (see page 3-17).

Bit Definitions

Bit **Name** **Function**

15–14 Res **Reserved**
For compatibility with future devices, always write this bit field with its chip reset default value.

13 TEPI **Transmit End-of-Packet Interrupt**
0 = The SmartDMA channel does not generate an interrupt after transmitting the last byte of the packet.

1 = The SmartDMA channel generates an interrupt after transmitting the last byte of the packet.

The SmartDMA channel waits for an indication from the requesting source that the packet has been transmitted successfully before generating the interrupt.

12 TBUI **Transmit Buffer Unavailable Interrupt**
0 = The SmartDMA channel does not generate an interrupt after detecting an unavailable buffer during a transmission of a packet.

1 = The SmartDMA channel generates an interrupt after detecting an unavailable buffer during a transmission of a packet.

11 TTCI **Transmit Terminal Count Interrupt**
0 = The SmartDMA channel does not generate an interrupt after transmitting the last byte of the current buffer.

1 = The SmartDMA channel generates an interrupt after transmitting the last byte of the current buffer and the TTCE bit in word 2 of the Transmit Buffer Descriptor must be set.

If the last byte of the buffer is also the last byte of the packet, the SmartDMA channel waits for an indication from the requesting source that the packet has been transmitted successfully before generating the interrupt.

10 REPI **Receive End-of-Packet Interrupt**
0 = The SmartDMA channel does not generate an interrupt after receiving the last byte of the packet.

1 = The SmartDMA channel generates an interrupt after receiving the last byte of the packet.

9 RBUI **Receive Buffer Unavailable Interrupt**
0 = The SmartDMA channel does not generate an interrupt after detecting an unavailable buffer before receiving a packet or during the reception of a packet.

1 = The SmartDMA channel generates an interrupt after detecting an unavailable buffer before receiving a packet or during the reception of a packet.

Bit	Name	Function
8	RTCI	**Receive Terminal Count Interrupt** 0 = The SmartDMA channel does not generate an interrupt after receiving the last byte of the current buffer. 1 = The SmartDMA channel generates an interrupt after receiving the last byte of the current buffer and the RTCE bit in word 2 of the Receive Buffer Descriptor must be set.
7	TXSO	**Transmit Set Own** 0 = The SmartDMA channel clears the OWN bit of the current buffer descriptor when the SmartDMA channel finishes the current transmit buffer. 1 = The SmartDMA channel does not clear the OWN bit when the SmartDMA channel finishes the current transmit buffer.
6	RXSO	**Receive Set Own** 0 = The SmartDMA channel clears the OWN bit of the current buffer descriptor when the SmartDMA channel finishes the current receive buffer. 1 = The SmartDMA channel does not clear the OWN bit when the SmartDMA channel finishes the current receive buffer.
5–4	P	**Relative Priority** These two bits indicate the priority of this channel pair relative to other channel pairs during simultaneous transfers. 00 = Low priority 01 = Medium priority 10 = High priority 11 = Reserved
3	POLL	**Forced Poll** 0 = Writing a 0 to this bit has no effect. 1 = The SmartDMA channel polls the OWN bit of the current buffer descriptor in the next opportunity. The POLL bit is cleared by the SmartDMA channel and always reads back 0.
2	Res	**Reserved** For compatibility with future devices, always write this bit field with its chip reset default value.
1	TXST	**Start/Stop Transmit SmartDMA Channel** 0 = Stop Transmit SmartDMA channel. 1 = Start Transmit SmartDMA channel; if the channel is already started, restarting it has no effect. Clearing the TXST bit has no effect while a request is pending on the channel. Before clearing the TXST bit, make sure the requesting HDLC channel is stopped.
0	RXST	**Start/Stop Receive SmartDMA Channel** 0 = Stop Receive SmartDMA channel. 1 = Start Receive SmartDMA channel; if the channel is already started, restarting it has no effect. Clearing the RXST bit has no effect while a request is pending on the channel. Before clearing the RXST bit, make sure the requesting HDLC channel is stopped.

Programming Notes

On the Am186CC and Am186CH microcontrollers, SmartDMA channel pairs 0 and 1 can service DMA requests for HDLC channel A and B, respectively. The Am186CU USB microcontroller does not use SmartDMA channel pairs 0 and 1.

On the Am186CU USB microcontroller, SmartDMA channel pairs 2 and 3 can service USB endpoints A, B, C, and D. The Am186CH HDLC microcontroller does not use SmartDMA channel pairs 2 and 3.

On the Am186CC microcontroller only, SmartDMA channel pairs 2 and 3 can service either HDLC channels B and C or USB endpoints A, B, C, and D.

Software can enable each status bit in the SDxSTAT register to generate an interrupt by setting the corresponding bit in this register (SDxCON). In addition to setting bits in the SDxCON register, software must configure the interrupt Channel 4–7 Control registers correctly. The appropriate CH4CON, CH5CON, CH6CON, or CH7CON registers must be configured to select each required interrupt channel's internal source (SmartDMA channel pair 0, 1, 2 and 3, respectively) and to enable the interrupt channel. See Chapter 9, "Interrupt Controller Registers".

SmartDMA Channel Pair 2 Control (SD2CON) Offset 170h
SmartDMA Channel Pair 3 Control (SD3CON) Offset 188h

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	*Res*		TEPI	TBUI	TTCI	REPI	RBUI	RTCI	TXSO	RXSO	P		POLL	DSEL	TXST	RXST
Software Read/Write	R		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		R/W	R/W	R/W	R/W
Hardware Set/Clear	—		—	—	—	—	—	—	—	—	—		C	—	—	—
Chip Reset Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Microcontroller:
Am186CC ✔
Am186CH
Am186CU !
! *= See bit definitions.*

Register Description

This register is used to set up SmartDMA channel pair 2 or 3. Except for the DSEL bit, the descriptions are the same as for channel pairs 0 and 1 (see page 3-14).

Bit Definitions

Bit	Name	Function
15–14	Res	**Reserved (Same as SD0CON)** For compatibility with future devices, always write this bit field with its chip reset default value.
13	TEPI	**Transmit End-of-Packet Interrupt (Same as SD0CON)** 0 = The SmartDMA channel does not generate an interrupt after transmitting the last byte of the packet. 1 = The SmartDMA channel generates an interrupt after transmitting the last byte of the packet. The SmartDMA channel waits for an indication from the requesting source that the packet has been transmitted successfully before generating the interrupt.
12	TBUI	**Transmit Buffer Unavailable Interrupt (Same as SD0CON)** 0 = The SmartDMA channel does not generate an interrupt after detecting an unavailable buffer during a transmission of a packet. 1 = The SmartDMA channel generates an interrupt after detecting an unavailable buffer during a transmission of a packet.
11	TTCI	**Transmit Terminal Count Interrupt (Same as SD0CON)** 0 = The SmartDMA channel does not generate an interrupt after transmitting the last byte of the current buffer. 1 = The SmartDMA channel generates an interrupt after transmitting the last byte of the current buffer and the TTCE bit in word 2 of the Transmit Buffer Descriptor must be set. If the last byte of the buffer is also the last byte of the packet, the SmartDMA channel waits for an indication from the requesting source that the packet has been transmitted successfully before generating the interrupt.
10	REPI	**Receive End-of-Packet Interrupt (Same as SD0CON)** 0 = The SmartDMA channel does not generate an interrupt after receiving the last byte of the packet. 1 = The SmartDMA channel generates an interrupt after receiving the last byte of the packet.
9	RBUI	**Receive Buffer Unavailable Interrupt (Same as SD0CON)** 0 = The SmartDMA channel does not generate an interrupt after detecting an unavailable buffer before receiving a packet or during the reception of a packet. 1 = The SmartDMA channel generates an interrupt after detecting an unavailable buffer before receiving a packet or during the reception of a packet.

Bit	Name	Function
8	RTCI	**Receive Terminal Count Interrupt (Same as SD0CON)** 0 = The SmartDMA channel does not generate an interrupt after receiving the last byte of the current buffer. 1 = The SmartDMA channel generates an interrupt after receiving the last byte of the current buffer and the RTCE bit in word 2 of the Receive Buffer Descriptor must be set.
7	TXSO	**Transmit Set Own (Same as SD0CON)** 0 = The SmartDMA channel clears the OWN bit of the current buffer descriptor when the SmartDMA channel finishes the current transmit buffer. 1 = The SmartDMA channel does not clear the OWN bit when the SmartDMA channel finishes the current transmit buffer.
6	RXSO	**Receive Set Own (Same as SD0CON)** 0 = The SmartDMA channel clears the OWN bit of the current buffer descriptor when the SmartDMA channel finishes the current receive buffer. 1 = The SmartDMA channel does not clear the OWN bit when the SmartDMA channel finishes the current receive buffer.
5–4	P	**Relative Priority (Same as SD0CON)** These two bits indicate the priority of this channel pair relative to other channel pairs during simultaneous transfers. 00 = Low priority 01 = Medium priority 10 = High priority 11 = Reserved
3	POLL	**Forced Poll (Same as SD0CON)** 0 = Writing a 0 to this bit has no effect. 1 = The SmartDMA channel polls the OWN bit of the current buffer descriptor in the next opportunity. The POLL bit is cleared by the SmartDMA channel and always reads back 0.
2	DSEL	**Requesting Source Select** 0 = Select HDLC as requesting source. 1 = Select USB as requesting source. ***Am186CU Microcontroller:*** *The default value of 0b is not supported on the Am186CU USB microcontroller. Am186CU USB microcontroller software must change the DSEL bit field to 1b before using pairs 2 and 3.*
1	TXST	**Start/Stop Transmit SmartDMA Channel (Same as SD0CON)** 0 = Stop Transmit SmartDMA channel. 1 = Start Transmit SmartDMA channel; if the channel is already started, restarting it has no effect. Clearing the TXST bit has no effect while a request is pending on the channel. Before clearing the TXST bit, make sure the requesting peripheral (HDLC channel or USB endpoint) is stopped.
0	RXST	**Start/Stop Receive SmartDMA Channel (Same as SD0CON)** 0 = Stop Receive SmartDMA channel. 1 = Start Receive SmartDMA channel; if the channel is already started, restarting it has no effect. Clearing the RXST bit has no effect while a request is pending on the channel. Before clearing the RXST bit, make sure the requesting peripheral (HDLC channel or USB endpoint) is stopped.

Programming Notes

On the Am186CC and Am186CH microcontrollers, SmartDMA channel pairs 0 and 1 can service DMA requests for HDLC channel A and B, respectively. The Am186CU USB microcontroller does not use SmartDMA channel pairs 0 and 1.

On the Am186CU USB microcontroller, SmartDMA channel pairs 2 and 3 can service USB endpoints A, B, C, and D. The Am186CH HDLC microcontroller does not use SmartDMA channel pairs 2 and 3.

On the Am186CC microcontroller only, SmartDMA channel pairs 2 and 3 can service either HDLC channels B and C, or USB endpoints A, B, C, and D.

Software can enable each status bit in the SDxSTAT register to generate an interrupt by setting the corresponding bit in this register (SDxCON). In addition to setting bits in the SDxCON register, software must configure the interrupt Channel 4–7 Control registers correctly. The appropriate CH4CON, CH5CON, CH6CON, or CH7CON registers must be configured to select each required interrupt channel's internal source (SmartDMA channel pair 0, 1, 2 and 3, respectively) and to enable the interrupt channel. See Chapter 9, "Interrupt Controller Registers".

SmartDMA Channel Pair 0 Transmit Ring Count/Address Low (SD0TRCAL) Offset 142h
SmartDMA Channel Pair 1 Transmit Ring Count/Address Low (SD1TRCAL) Offset 15Ah
SmartDMA Channel Pair 2 Transmit Ring Count/Address Low (SD2TRCAL) Offset 172h
SmartDMA Channel Pair 3 Transmit Ring Count/Address Low (SD3TRCAL) Offset 18Ah

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	TRA[15–4]												*Res*	TRC		
Software Read/Write	R/W												R	R/W		
Hardware Set/Clear	—												—	—		
Chip Reset Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Microcontroller:	Pair: 0	1	2	3
Am186CC	✓	✓	✓	✓
Am186CH	✓	✓		
Am186CU			✓	✓

Register Description

This register is used to indicate the number of available buffer descriptors in a SmartDMA channel pair's transmit buffer descriptor ring. This register also contains the low bits (15–4) of the ring's base address.

Bit Definitions

Bit	Name	Function
15–4	TRA[15–4]	**SmartDMA Channel Transmit Ring Address Low** The TRA[15–4] bit field contains the low address bits of the start location of the transmit buffer descriptor ring in memory. Because the base address of the ring must be paragraph aligned, the TRA[3–0] bits are always 0. The TRA[19–16] bits are in the SDxTRAH register (see page 3-21).
3	Res	**Reserved** For compatibility with future devices, always write this bit field with its chip reset default value.
2–0	TRC	**SmartDMA Channel Transmit Ring Count** The TRC bit field indicates the number of entries in the transmit buffer descriptor ring. The actual number is encoded as follows: 000 = 1 001 = 2 010 = 4 011 = 8 100 = 16 101 = 32 110 = 64 111 = 128

Programming Notes

SmartDMA Channel Pair 0 Transmit Ring Address High (SD0TRAH) Offset 144h
SmartDMA Channel Pair 1 Transmit Ring Address High (SD1TRAH) Offset 15Ch
SmartDMA Channel Pair 2 Transmit Ring Address High (SD2TRAH) Offset 174h
SmartDMA Channel Pair 3 Transmit Ring Address High (SD3TRAH) Offset 18Ch

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	*Res*												TRA[19–16]			
Software Read/Write	R												R/W			
Hardware Set/Clear	—												—			
Chip Reset Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Microcontroller:	Pair: 0	1	2	3
Am186CC	✓	✓	✓	✓
Am186CH	✓	✓		
Am186CU			✓	✓

Register Description

This register is used set the high bits (19–16) for a SmartDMA channel pair’s transmit buffer descriptor ring base address.

Bit Definitions

Bit	Name	Function
15–4	Res	**Reserved** For compatibility with future devices, always write this bit field with its chip reset default value.
3–0	TRA[19–16]	**SmartDMA Channel Transmit Ring Address High** The TRA[19–16] bit field contains the high address bits of the start location of the transmit buffer descriptor ring in memory. The TRA[15–4] bits are in the SDxTRCAL register (see page 3-20). Because the base address of the ring must be paragraph aligned, the TRA[3–0] bits are always 0.

Programming Notes

SmartDMA Channel Pair 0 Receive Ring Count/Address Low (SD0RRCAL) Offset 146h
SmartDMA Channel Pair 1 Receive Ring Count/Address Low (SD1RRCAL) Offset 15Eh
SmartDMA Channel Pair 2 Receive Ring Count/Address Low (SD2RRCAL) Offset 176h
SmartDMA Channel Pair 3 Receive Ring Count/Address Low (SD3RRCAL) Offset 18Eh

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	RRA[15–4]												*Res*	RRC		
Software Read/Write	R/W												R	R/W		
Hardware Set/Clear	—												—	—		
Chip Reset Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Microcontroller:	Pair: 0	1	2	3
Am186CC	✓	✓	✓	✓
Am186CH	✓	✓		
Am186CU			✓	✓

Register Description

This register is used to indicate the number of available buffer descriptors in a SmartDMA channel pair's receive buffer descriptor ring. This register also contains the low bits (15–4) of the ring's base address.

Bit Definitions

Bit	Name	Function
15–4	RRA[15–4]	**SmartDMA Channel Receive Ring Address Low** The RRA[15–4] bit field contains the low address bits of the start location of the receive buffer descriptor ring in memory. Because the base address of the ring must be paragraph aligned, the RRA[3–0] bits are always 0. The RRA[19–16] bits are in the SDxRRAH register (see page 3-23).
3	Res	**Reserved** For compatibility with future devices, always write this bit field with its chip reset default value.
2–0	RRC	**SmartDMA Channel Receive Ring Count** The RRC bit field indicates the number of entries in the receive buffer descriptor ring. The actual number is encoded as follows: 000 = 1 001 = 2 010 = 4 011 = 8 100 = 16 101 = 32 110 = 64 111 = 128

Programming Notes

SmartDMA Channel Pair 0 Receive Ring Address High (SD0RRAH) Offset 148h
SmartDMA Channel Pair 1 Receive Ring Address High (SD1RRAH) Offset 160h
SmartDMA Channel Pair 2 Receive Ring Address High (SD2RRAH) Offset 178h
SmartDMA Channel Pair 3 Receive Ring Address High (SD3RRAH) Offset 190h

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	*Res*												RRA[19–16]			
Software Read/Write	R												R/W			
Hardware Set/Clear	—												—			
Chip Reset Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	Pair:			
Microcontroller:	0	1	2	3
Am186CC	✓	✓	✓	✓
Am186CH	✓	✓		
Am186CU			✓	✓

Register Description

This register is used to set the high bits (19–16) for a SmartDMA channel pair's receive buffer descriptor ring base address.

Bit Definitions

Bit	Name	Function
15–4	Res	**Reserved** For compatibility with future devices, always write this bit field with its chip reset default value.
3–0	RRA[19–16]	**SmartDMA Channel Receive Ring Address High** The RRA[19–16] bit field contains the high address bits of the start location of the receive buffer descriptor ring in memory. The RRA[15–4] bits are in the SDxRRCAL register (see page 3-22). Because the base address of the ring must be paragraph aligned, the RRA[3–0] bits are always 0.

Programming Notes

SmartDMA Channel Pair 0 Status (SD0STAT) Offset 14Ah
SmartDMA Channel Pair 1 Status (SD1STAT) Offset 162h
SmartDMA Channel Pair 2 Status (SD2STAT) Offset 17Ah
SmartDMA Channel Pair 3 Status (SD3STAT) Offset 192h

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	*Res*		TEP	TBU	TTC	REP	RBU	RTC	*Res*							
Software Read/Write	R		R/W0	R/W0	R/W0	R/W0	R/W0	R/W0	R							
Hardware Set/Clear	—		S	S	S	S	S	S	—							
Chip Reset Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Microcontroller:	Pair: 0	1	2	3
Am186CC	✓	✓	✓	✓
Am186CH	✓	✓		
Am186CU			✓	✓

Register Description

This register is used to indicate the status of each SmartDMA channel pair. Software must clear the status bits.

Bit Definitions

Bit	Name	Function
15–14	Res	**Reserved** For compatibility with future devices, always write this bit field with its chip reset default value.
13	TEP	**Transmit End-of-Packet** 0 = The event has not occurred, or software has cleared this bit. 1 = The last byte of the packet has been transmitted successfully by the transmitter.
12	TBU	**Transmit Buffer Unavailable** 0 = The event has not occurred, or software has cleared this bit. 1 = A transmit buffer is not available.
11	TTC	**Transmit Terminal Count** 0 = The event has not occurred, or software has cleared this bit. 1 = The last byte of the current buffer has been transmitted and the buffer is released.
10	REP	**Receive End-of-Packet** 0 = The event has not occurred, or software has cleared this bit. 1 = The last byte of the packet has been received.
9	RBU	**Receive Buffer Unavailable** 0 = The event has not occurred, or software has cleared this bit. 1 = A buffer is not available before receiving a packet or during the reception of a packet.
8	RTC	**Receive Terminal Count** 0 = The event has not occurred, or software has cleared this bit. 1 = The last byte of the current buffer has been received and the buffer is released.
7–0	Res	**Reserved** For compatibility with future devices, always write this bit field with its chip reset default value.

Programming Notes

Software can enable each status bit in this register (SDxSTAT) to generate an interrupt by setting the corresponding bit in the SDxCON register (see page 3-14 and page 3-17). In addition to setting bits in the SDxCON register, software must configure the interrupt Channel 4–7 Control registers correctly. The appropriate CH4CON, CH5CON, CH6CON, or CH7CON registers must be configured to select each required interrupt channel's internal source (SmartDMA channel pair 0, 1, 2 and 3, respectively) and to enable the interrupt channel. See Chapter 9, "Interrupt Controller Registers".

SmartDMA Channel Pair 0 Current Buffer Descriptor (SD0CBD) Offset 14Ch
SmartDMA Channel Pair 1 Current Buffer Descriptor (SD1CBD) Offset 164h
SmartDMA Channel Pair 2 Current Buffer Descriptor (SD2CBD) Offset 17Ch
SmartDMA Channel Pair 3 Current Buffer Descriptor (SD3CBD) Offset 194h

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	*Res*	CRBD							*Res*	CTBD						
Software Read/Write	R	R/W							R	R/W						
Hardware Set/Clear	—	S/C							—	S/C						
Chip Reset Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Microcontroller:	Pair: 0	1	2	3
Am186CC	✓	✓	✓	✓
Am186CH	✓	✓		
Am186CU			✓	✓

Register Description

This register is used to indicate the buffer descriptor currently accessed by a SmartDMA channel pair.

Bit Definitions

Bit	Name	Function
15	Res	**Reserved** For compatibility with future devices, always write this bit field with its chip reset default value.
14–8	CRBD	**SmartDMA Channel Current Receive Buffer Descriptor** This bit field indicates the currently accessed buffer descriptor in the receive ring. Writing to this bit field causes the receive SmartDMA channel to transfer data to the address pointed to by this buffer descriptor. The RXST bit in the control register (see page 3-15 and page 3-18) must be cleared prior to writing to this bit field.
7	Res	**Reserved** For compatibility with future devices, always write this bit field with its chip reset default value.
6–0	CTBD	**SmartDMA Channel Current Transmit Buffer Descriptor** This bit field indicates the currently accessed buffer descriptor in the transmit ring. Writing to this bit field causes the transmit SmartDMA channel to transfer data from the address pointed to by this buffer descriptor. The TXST bit in the control register (see page 3-15 and page 3-18) must be cleared prior to writing to this bit field.

Programming Notes

Software can write to the CRBD bit field only when the RXST bit field of the SDxCON register is cleared. Software can write to the CTBD bit field only when the TXST bit field of the SDxCON register is cleared. An inappropriate write to either bit field is ignored.

SmartDMA Channel Pair 0 Current Transmit Address (SD0CTAD) Offset 14Eh
SmartDMA Channel Pair 1 Current Transmit Address (SD1CTAD) Offset 166h
SmartDMA Channel Pair 2 Current Transmit Address (SD2CTAD) Offset 17Eh
SmartDMA Channel Pair 3 Current Transmit Address (SD3CTAD) Offset 196h

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	CTAD															
Software Read/Write	R															
Hardware Set/Clear	S/C															
Chip Reset Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Microcontroller:	Pair: 0	1	2	3
Am186CC	✔	✔	✔	✔
Am186CH	✔	✔		
Am186CU			✔	✔

Register Description

This register is used to indicate the current memory address of the data being transmitted by a transmit SmartDMA channel.

Bit Definitions

Bit	Name	Function
15–0	CTAD	**SmartDMA Channel Current Transmit Address** The CTAD bit field indicates the current memory address of the data byte being transmitted by the SmartDMA channel. Only the low 16 bits of the address are recorded in this bit field.

Programming Notes

SmartDMA Channel Pair 0 Current Receive Address (SD0CRAD) Offset 150h
SmartDMA Channel Pair 1 Current Receive Address (SD1CRAD) Offset 168h
SmartDMA Channel Pair 2 Current Receive Address (SD2CRAD) Offset 180h
SmartDMA Channel Pair 3 Current Receive Address (SD3CRAD) Offset 198h

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	CRAD															
Software Read/Write	R															
Hardware Set/Clear	S/C															
Chip Reset Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	Pair:			
Microcontroller:	0	1	2	3
Am186CC	✓	✓	✓	✓
Am186CH	✓	✓		
Am186CU			✓	✓

Register Description

This register is used to indicate the current memory address of the data being received by a receive SmartDMA channel.

Bit Definitions

Bit	Name	Function
15–0	CRAD	**SmartDMA Channel Current Receive Address** The CRAD bit field indicates the current memory address of the data byte being received by the SmartDMA channel. Only the low 16 bits of the address are recorded in this bit field.

Programming Notes

CHAPTER

4 USB REGISTERS (Am186CC and Am186CU Microcontrollers Only)

4.1 OVERVIEW

This chapter describes the Universal Serial Bus (USB) registers on the Am186CC and Am186CU microcontrollers. *The Am186CH HDLC microcontroller does not support USB.*

The Am186CC and Am186CU microcontrollers provide a total of six USB endpoints that can be used in the design of a USB peripheral device.

Four data endpoints (A–D) can be configured for bulk, isochronous, or interrupt USB transfer types, in the IN or OUT direction. Any transfer type can be configured to operate in Non-DMA mode. Bulk and isochronous transfers can use a general-purpose DMA or SmartDMA channel. (See Chapter 3 for information about DMA registers.)

In addition to the four configurable endpoints, the microcontroller provides a dedicated USB interrupt IN endpoint and a USB control endpoint.

For more information about using USB, see the *Am186CC/CH/CU Microcontrollers User's Manual*, order #21914.

This chapter describes general USB configuration registers first, then control endpoint registers, interrupt endpoint registers, and data endpoint registers. Within the data endpoint registers, the register sets are identical for each of the four data endpoints. This chapter describes registers with identical functions only once. The unique register names and offsets that apply are listed at the top of each register page. Table 4-1 lists the USB registers in offset order, with the corresponding description's page number.

Table 4-1 USB Register Map

Register Name	Mnemonic	Offset	Page Number
General Configuration Registers			
USB Interrupt Status 1	UISTAT1	1E0h	page 4-3
USB Interrupt Mask 1	UIMASK1	1E2h	page 4-5
USB Interrupt Status 2	UISTAT2	1E4h	page 4-7
USB Interrupt Mask 2	UIMASK2	1E6h	page 4-9
USB Device Miscellaneous Functions	USBMFR	1E8h	page 4-10
Real-Time Frame Monitor Count	RTFMCNT	1EAh	page 4-12
Time Stamp	TSTMP	1ECh	page 4-13
Time Stamp Match	TSTMPM	1EEh	page 4-14
Isochronous Synchronization Control	ISCTL	1F0h	page 4-15
Frame Position Monitor Count	FPMCNT	1F2h	page 4-17
Control Endpoint Registers			
Control Endpoint Control/Status	CNTCTL	200h	page 4-18
Control Endpoint Receive Packet Size	CNTSIZ	202h	page 4-21
Control Endpoint Data Port	CNTDAT	206h	page 4-22
Control Endpoint Receive Data Port Peek	CNTRPK	208h	page 4-23
Control Endpoint Definition 1	CNTDEF1	20Ah	page 4-24

Table 4-1 USB Register Map (Continued)

Register Name	Mnemonic	Offset	Page Number
Control Endpoint Definition 2	CNTDEF2	20Ch	page 4-26
Interrupt Endpoint Registers			
Interrupt Endpoint Control/Status	IEPCTL	210h	page 4-27
Interrupt Endpoint Data Port	IEPDAT	216h	page 4-29
Interrupt Endpoint Definition 1	IEPDEF1	21Ah	page 4-30
Interrupt Endpoint Definition 2	IEPDEF2	21Ch	page 4-32
A Endpoint Registers			
A Endpoint Control/Status	AEPCTL	220h	page 4-33
A Endpoint Received Packet Size	AEPSIZ	222h	page 4-39
A Endpoint Buffer Status	AEPBUFS	224h	page 4-40
A Endpoint Data Port	AEPDAT	226h	page 4-42
A Endpoint Receive Data Port Peek	ARCVPK	228h	page 4-43
A Endpoint Definition 1	AEPDEF1	22Ah	page 4-44
A Endpoint Definition 2	AEPDEF2	22Ch	page 4-46
A Endpoint Definition 3	AEPDEF3	22Eh	page 4-50
B Endpoint Registers			
B Endpoint Control/Status	BEPCTL	230h	page 4-33
B Endpoint Received Packet Size	BEPSIZ	232h	page 4-39
B Endpoint Buffer Status	BEPBUFS	234h	page 4-40
B Endpoint Data Port	BEPDAT	236h	page 4-42
B Endpoint Receive Data Port Peek	BRCVPK	238h	page 4-43
B Endpoint Definition 1	CEPDEF1	23Ah	page 4-44
B Endpoint Definition 2	BEPDEF2	23Ch	page 4-46
B Endpoint Definition 3	BEPDEF3	23Eh	page 4-50
C Endpoint Registers			
C Endpoint Control/Status	CEPCTL	240h	page 4-33
C Endpoint Received Packet Size	CEPSIZ	242h	page 4-39
C Endpoint Buffer Status	CEPBUFS	244h	page 4-41
C Endpoint Data Port	CEPDAT	246h	page 4-42
C Endpoint Receive Data Port Peek	CRCVPK	248h	page 4-43
C Endpoint Definition 1	CEPDEF1	24Ah	page 4-44
C Endpoint Definition 2	CEPDEF2	24Ch	page 4-48
C Endpoint Definition 3	CEPDEF3	24Eh	page 4-50
D Endpoint Registers			
D Endpoint Control/Status	DEPCTL	250h	page 4-33
D Endpoint Received Packet Size	DEPSIZ	252h	page 4-39
D Endpoint Buffer Status	DEPBUFS	254h	page 4-41
D Endpoint Data Port	DEPDAT	256h	page 4-42
D Endpoint Receive Data Port Peek	DRCVPK	258h	page 4-43
D Endpoint Definition 1	DEPDEF1	25Ah	page 4-44
D Endpoint Definition 2	DEPDEF2	25Ch	page 4-48
D Endpoint Definition 3	DEPDEF3	25Eh	page 4-50

USB Interrupt Status 1 (UISTAT1) Offset 1E0h

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	*Res*				D_EP_STATINT	D_EP_ACT	C_EP_STATINT	C_EP_ACT	B_EP_STATINT	B_EP_ACT	A_EP_STATINT	A_EP_ACT	OTHER_INT	INT_EP_ACT	CNT_EP_NEW	CNT_EP_ACT
Software Read/Write	R				R/W0	R/W0	R/W0	R/W0	R/W0	R/W0	R/W0	R/W0	R	R/W0	R/W0	R/W0
Hardware Set/Clear	—				S/C	S/C	S/C	S/C	S/C	S/C	S/C	S/C	S/C	S/C	S/C	S/C
Chip Reset Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Microcontroller:
Am186CC ☑
Am186CH □
Am186CU ☑

Register Description

This register is used to determine the source of the USB interrupt. Except for the OTHER_INT and reserved bits, software can enable any bit in this register as an interrupt source by setting the corresponding bit in the UIMASK1 register (see page 4-5). Software must test this register to determine the explicit interrupt source. Software should clear the individual source bits located in each endpoint's Control/Status register as part of the service routine.

Bit Definitions

Bit	Name	Function
15–12	Res	**Reserved** For compatibility with future devices, always write this bit field with its chip reset default value.
11	D_EP_STATINT	**Endpoint D Status Interrupt Set** 0 = The STAT_INT bit located in the DEPCTL register (see page 4-35) is clear. 1 = The STAT_INT bit located in the DEPCTL register is set.
10	D_EP_ACT	**Endpoint D ACT_REQ Bit Set** 0 = The ACT_REQ bit located in the DEPCTL register (see page 4-34) is clear. 1 = The ACT_REQ bit located in the DEPCTL register is set.
9	C_EP_STATINT	**Endpoint C Status Interrupt Set** 0 = The STAT_INT bit located in the CEPCTL register (see page 4-35) is clear. 1 = The STAT_INT bit located in the CEPCTL register is set.
8	C_EP_ACT	**Endpoint C ACT_REQ Bit Set** 0 = The ACT_REQ bit located in the CEPCTL register (see page 4-34) is clear. 1 = The ACT_REQ bit located in the CEPCTL register is set.
7	B_EP_STATINT	**Endpoint B Status Interrupt Set** 0 = The STAT_INT bit located in the BEPCTL register (see page 4-35) is clear. 1 = The STAT_INT bit located in the BEPCTL register is set.
6	B_EP_ACT	**Endpoint B ACT_REQ Bit Set** 0 = The ACT_REQ bit located in the BEPCTL register (see page 4-34) is clear. 1 = The ACT_REQ bit located in the BEPCTL register is set.
5	A_EP_STATINT	**Endpoint A Status Interrupt Set** 0 = The STAT_INT bit located in the AEPCTL register (see page 4-35) is clear. 1 = The STAT_INT bit located in the AEPCTL register is set.
4	A_EP_ACT	**Endpoint A ACT_REQ Bit Set** 0 = The ACT_REQ bit located in the AEPCTL register (see page 4-34) is clear. 1 = The ACT_REQ bit located in the AEPCTL register is set.

Bit	Name	Function
3	OTHER_INT	**Other Interrupt Sources Have Been Set** This bit is set when any one of the interrupt source bits in the UISTAT2 register (see page 4-7) is set, unless that source is masked in UIMASK2. Note that this bit does not generate an interrupt; therefore, the bit does not have an associated mask bit in the UIMASK1 register.
2	INT_EP_ACT	**Interrupt Endpoint ACT_REQ Bit Set** 0 = The ACT_REQ bit located in the IEPCTL register (see page 4-28) is clear. 1 = The ACT_REQ bit located in the IEPCTL register is set.
1	CNT_EP_NEW	**Control Endpoint New Command Bit Set** 0 = The NEW_COMMAND bit located in the CNTCTL register (see page 4-19) is clear. 1 = The NEW_COMMAND bit located in the CNTCTL register is set.
0	CNT_EP_ACT	**Control Endpoint ACT_REQ Bit Is Set** 0 = The ACT_REQ bit located in the CNTCTL register (see page 4-19) is clear. 1 = The ACT_REQ bit located in the CNTCTL register is set.

Programming Notes

In addition to setting bits in the UIMASK1 register, software must configure the interrupt Channel 2 Control (CH2CON) register to select the interrupt channel’s internal source (USB) and enable the channel before any USB interrupts can be generated. See Chapter 9, “Interrupt Controller Registers”.

USB Interrupt Mask 1 (UIMASK1) Offset 1E2h

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	*Res*				D_EP_STATINT	D_EP_ACT	C_EP_STATINT	C_EP_ACT	B_EP_STATINT	B_EP_ACT	A_EP_STATINT	A_EP_ACT	OI_UNM	INT_EP_ACT	CNT_EP_NEW	CNT_EP_ACT
Software Read/Write	R				R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W
Hardware Set/Clear	—				—	—	—	—	—	—	—	—	—	—	—	—
Chip Reset Default	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0

Microcontroller:
Am186CC ☑
Am186CH ☐
Am186CU ☑

Register Description

This register allows or disallows the bits located in the UISTAT1 register (see page 4-3) to generate an interrupt when they are set. Setting any mask bit in this register enables the corresponding bit in the UISTAT1 register as an interrupt source.

Bit Definitions

Bit	Name	Function
15–12	Res	**Reserved** For compatibility with future devices, always write this bit field with its chip reset default value.
11	D_EP_STATINT	**Endpoint D Status Interrupt Set Mask** 0 = The D_EP_STATINT bit interrupt in the UISTAT1 register is masked off. 1 = The D_EP_STATINT bit interrupt in the UISTAT1 register is enabled.
10	D_EP_ACT	**Endpoint D ACT_REQ Bit Set Mask** 0 = The D_EP_ACT bit interrupt in the UISTAT1 register is masked off. 1 = The D_EP_ACT bit interrupt in the UISTAT1 register is enabled.
9	C_EP_STATINT	**Endpoint C Status Interrupt Set Mask** 0 = The C_EP_STATINT bit interrupt in the UISTAT1 register is masked off. 1 = The C_EP_STATINT bit interrupt in the UISTAT1 register is enabled.
8	C_EP_ACT	**Endpoint C ACT_REQ Bit Set Mask** 0 = The C_EP_ACT bit interrupt in the UISTAT1 register is masked off. 1 = The C_EP_ACT bit interrupt in the UISTAT1 register is enabled.
7	B_EP_STATINT	**Endpoint B Status Interrupt Set Mask** 0 = The B_EP_STATINT bit interrupt in the UISTAT1 register is masked off. 1 = The B_EP_STATINT bit interrupt in the UISTAT1 register is enabled.
6	B_EP_ACT	**Endpoint B ACT_REQ Bit Set Mask** 0 = The B_EP_ACT bit interrupt in the UISTAT1 register is masked off. 1 = The B_EP_ACT bit interrupt in the UISTAT1 register is enabled.
5	A_EP_STATINT	**Endpoint A Status Interrupt Set Mask** 0 = The A_EP_STATINT bit interrupt in the UISTAT1 register is masked off. 1 = The A_EP_STATINT bit interrupt in the UISTAT1 register is enabled.
4	A_EP_ACT	**Endpoint A ACT_REQ Bit Set Mask** 0 = The A_EP_ACT bit interrupt in the UISTAT1 register is masked off. 1 = The A_EP_ACT bit interrupt in the UISTAT1 register is enabled.
3	OI_UNM	**Other Interrupt Un-Mask** The OTHER_INT bit in the UISTAT1 register does not generate interrupts, so this bit always reads back as 1.

Bit	Name	Function
2	INT_EP_ACT	**Interrupt Endpoint ACT_REQ Bit Set Mask** 0 = The INT_EP_ACT bit interrupt in the UISTAT1 register is masked off. 1 = The INT_EP_ACT bit interrupt in the UISTAT1 register is enabled.
1	CNT_EP_NEW	**Control Endpoint New Command Bit Set Mask** 0 = The CNT_EP_NEW bit interrupt in the UISTAT1 register is masked off. 1 = The CNT_EP_NEW bit interrupt in the UISTAT1 register is enabled.
0	CNT_EP_ACT	**Control Endpoint ACT_REQ Bit Set Mask** 0 = The CNT_EP_ACT bit interrupt in the UISTAT1 register is masked off. 1 = The CNT_EP_ACT bit interrupt in the UISTAT1 register is enabled.

Programming Notes

In addition to setting bits in this register (UIMASK1), software must configure the interrupt Channel 2 Control (CH2CON) register to select the interrupt channel’s internal source (USB) and enable the channel before any USB interrupts can be generated. See Chapter 9, “Interrupt Controller Registers”.

USB Interrupt Status 2 (UISTAT2) Offset 1E4h

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	USB_RST	USB_SUS	USB_RES	*Res*									TSTMP_M	POS_UP	SOF_GEN	MS_SOF
Software Read/Write	R/W0	R/W0	R/W0	R									R/W0	R/W0	R/W0	R/W0
Hardware Set/Clear	S	S	S	—									S	S	S	S
Chip Reset Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Microcontroller:
Am186CC ✓
Am186CH □
Am186CU ✓

Register Description

This register is used to determine the source of the USB interrupt. Except for reserved bits, software can enable any bit in this register as an interrupt source by setting the corresponding bit in the UIMASK2 register (see page 4-9). Software must test this register to determine the explicit interrupt source. Software should clear the individual source by writing a 0 to the corresponding bit in this register.

Bit Definitions

Bit	Name	Function
15	USB_RST	**USB Reset** 0 = Either the event has not occurred or software has cleared the bit. 1 = The USB host has reset the bus.
14	USB_SUS	**USB Suspend** 0 = Either the event has not occurred or software has cleared the bit. 1 = The USB suspend condition has been detected on the bus.
13	USB_RES	**USB Resume** 0 = Either the event has not occurred or software has cleared the bit. 1 = The USB resume condition has been detected on the bus. Note that software must clear this bit after generating a soft reset by setting the S_RES bit in the USBMFR register (see page 4-10).
12–4	Res	**Reserved** For compatibility with future devices, always write this bit field with its chip reset default value.
3	TSTMP_M	**Time Stamp Match** 0 = Either the event has not occurred or software has cleared the bit. 1 = The time stamp match hardware has been armed (i.e., the TSTMPM register has been written, see page 4-14) and the value in the TSTMP register (see page 4-13) is greater than or equal to the value in the TSTMPM register.
2	POS_UP	**Position Update** 0 = Either the event has not occurred or software has cleared the bit. 1 = A new value has been loaded into the FPMCNT register (see page 4-17).

Bit	Name	Function
1	SOF_GEN	**Start of Frame Generated** 0 = Either the event has not occurred or software has cleared the bit. 1 = This bit is set whenever a start-of-frame (SOF) packet from the host is detected on the bus or when the SOF packet is not detected and an internal SOF is synthesized. When this bit is set, the USB frame number value in the TSTMP register (see page 4-13) has been updated with the frame number received from the USB host or is updated by the previous value being incremented by one.
0	MS_SOF	**Missed Start-of-Frame Packet** 0 = Either the event has not occurred or software has cleared the bit. 1 = This bit is set whenever an SOF packet from the host is not detected within five USB clocks of when it is expected by the hardware. The MS_SOF bit is set after six USB clocks from the time an SOF was expected.

Programming Notes

In addition to setting bits in the UIMASK2 register, software must configure the interrupt Channel 2 Control (CH2CON) register to select the interrupt channel’s internal source (USB) and enable the channel before any USB interrupts can be generated. See Chapter 9, “Interrupt Controller Registers”.

USB Interrupt Mask 2 (UIMASK2) Offset 1E6h

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	USB_RST	USB_SUS	USB_RES	*Res*									TSTMP_M	POS_UP	SOF_GEN	MS_SOF
Software Read/Write	R/W	R/W	R/W	R									R/W	R/W	R/W	R/W
Hardware Set/Clear	—	—	—	—									—	—	—	—
Chip Reset Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Microcontroller:
Am186CC ☑
Am186CH □
Am186CU ☑

Register Description

This register allows or disallows the bits located in the UISTAT2 register (see page 4-7) to generate an interrupt when they are set. Setting any mask bit in this register enables the corresponding bit in the UISTAT2 register as an interrupt source.

Bit Definitions

Bit	Name	Function
15	USB_RST	**USB Reset Interrupt Mask** 0 = The USB_RST bit interrupt in the UISTAT2 register is masked off. 1 = The USB_RST bit interrupt in the UISTAT2 register is enabled.
14	USB_SUS	**USB Suspend Interrupt Mask** 0 = The USB_SUS bit interrupt in the UISTAT2 register is masked off. 1 = The USB_SUS bit interrupt in the UISTAT2 register is enabled.
13	USB_RES	**USB Resume Interrupt Mask** 0 = The USB_RES bit interrupt in the UISTAT2 register is masked off. 1 = The USB_RES bit interrupt in the UISTAT2 register is enabled.
12–4	Res	**Reserved** For compatibility with future devices, always write this bit field with its chip reset default value.
3	TSTMP_M	**Time Stamp Match Interrupt Mask** 0 = The TSTMP_M bit interrupt in the UISTAT2 register is masked off. 1 = The TSTMP_M bit interrupt in the UISTAT2 register is enabled.
2	POS_UP	**Position Update Interrupt Mask** This bit is set whenever a new value is loaded into the FPMCNT register.
1	SOF_GEN	**Start-of-Frame Generated Interrupt Mask** 0 = The SOF_GEN bit interrupt in the UISTAT2 register is masked off. 1 = The SOF_GEN bit interrupt in the UISTAT2 register is enabled.
0	MS_SOF	**Missed Start-of-Frame Packet Interrupt Mask** 0 = The MS_SOF bit interrupt in the UISTAT2 register is masked off. 1 = The MS_SOF bit interrupt in the UISTAT2 register is enabled.

Programming Notes

In addition to setting bits in this register (UIMASK2), software must configure the interrupt Channel 2 Control (CH2CON) register to select the interrupt channel’s internal source (USB) and enable the channel before any USB interrupts can be generated. See Chapter 9, “Interrupt Controller Registers”.

USB Device Miscellaneous Functions (USBMFR) Offset 1E8h

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	*Res*									PUP_XCVER	SUSP	S_RES	S_POWER	DIS_XCVER	RWAKE	RWAKE_EN
Software Read/Write	R									R/W	R	R/W1	R/W	R/W	R/W1	R
Hardware Set/Clear	—									—	S/C	—	—	—	—	S/C
Chip Reset Default	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0

Microcontroller:	
Am186CC	✔
Am186CH	□
Am186CU	✔

Register Description

This register is used to control various aspects of the transceiver, including power up, disable, and the remote wakeup features of the USB peripheral controller. This register also provides some of the device status information.

Bit Definitions

Bit **Name** **Function**

15–7 Res **Reserved**
For compatibility with future devices, always write this bit field with its chip reset default value.

6 PUP_XCVER **Power Up USB Transceiver**
0 = The USB internal transceiver is powered down and the transceiver interface is disabled.

1 = The USB internal transceiver is powered up and the transceiver interface is enabled.

The device software uses the PUP_XCVER bit to power up or power down the USB internal transceiver. Setting this bit powers up the transceiver.

Note that immediately following an external or watchdog timer reset, this bit is a 0 and therefore the internal transceiver is powered down.

The PUP_XCVER bit also enables or disables the transceiver interface. This bit must be set before an external USB transceiver can be used. An external power-up signal is not provided. However, a PIO can be configured as a software-controlled power-up signal for the external transceiver's power supply circuit.

5 SUSP **USB Suspend**
0 = The USB bus is not in Suspended mode.

1 = The USB bus is in Suspended mode.

The SUSP bit indicates the status of the USB bus.

4 S_RES **Soft Reset**
0 = This bit always reads back as 0b.

1 = Write a 1 to the S_RES bit to reset the USB controller front end logic. This should be done upon the detection of a host/hub attachment.

After using this bit to cause a soft reset, software must clear the USB_RES bit in the UISTAT2 register.

3 S_POWER **Self-Powered Device**
0 = Microcontroller reports that it is bus-powered upon receiving a GET_STATUS command.

1 = Microcontroller reports that it is self-powered upon receiving a GET_STATUS command.

This bit determines whether the USB controller reports that it is a self-powered or bus-powered peripheral. The S_POWER bit is provided for compatibility with possible future low-powered microcontrollers. Leave S_POWER = 1 (self-powered, the default) unless the design's power requirements adhere to the USB specification for bus-powered devices.

Bit	Name	Function
2	DIS_XCVER	**Disable USB Transceiver** 0 = The USB transceiver is enabled. When enabled, the transceiver is driven only when a valid packet is being transmitted from one of the endpoints to the USB host. 1 = The transceiver outputs are forced to be three-stated at all times. The device software uses the DIS_XCVER bit to force the USB transceiver drivers to be three-stated. If an external USB transceiver is used, setting this bit disables the external transceivers outputs by driving the $\overline{\text{UXVOE}}$ signal High.
1	RWAKE	**Remote Wakeup** 0 = This bit always reads back as 0b. 1 = Write a 1 to the RWAKE bit to cause a suspended USB to resume. This bit has no effect unless the RWAKE_EN bit is set and the USB is in its suspended state (the SUSP bit is 1).
0	RWAKE_EN	**Remote Wakeup Enable** 0 = The USB host has issued the CLEAR_FEATURE command with the Remote Wakeup Feature Selector set, or has not set this feature since an external or watchdog timer reset. 1 = The USB host has issued the SET_FEATURE command with the Remote Wakeup Feature Selector set. The device software uses this status bit to determine whether or not the USB host has enabled the remote wakeup feature for the device.

Programming Notes

Real-Time Frame Monitor Count (RTFMCNT) Offset 1EAh

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	*Res*		RTFCNT													
Software Read/Write	R		R													
Hardware Set/Clear	—		S/C													
Chip Reset Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Microcontroller:

Am186CC ☑
Am186CH □
Am186CU ☑

Register Description

The device software can read this register to determine approximately where in time the USB host is, relative to its last start-of-frame packet. Only the higher-order bits should be used as the hardware does not guarantee that any of the bit values are stable during the read access. The device software should be implemented so that it is required to read the same value on the higher-order bits in two consecutive read cycles prior to using the value that is returned.

Bit Definitions

Bit	Name	Function
15–14	Res	**Reserved** For compatibility with future devices, always write this bit field with its chip reset default value.
13–0	RTFCNT	**Real-Time Frame Count** The RTFCNT bit field contains the current USB frame counter value. This counter is incremented one time for every USB data clock time and is reset when an SOF packet from the host is detected or when an internal SOF is synthesized.

Programming Notes

Time Stamp (TSTMP) Offset 1ECh

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	*Res*					TSTMP										
Software Read/Write	R					R										
Hardware Set/Clear	—					S/C										
Chip Reset Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Microcontroller:
Am186CC ☑
Am186CH □
Am186CU ☑

Register Description

This register is used by device software to read the current USB time stamp value, otherwise known as the USB frame number. This value is transferred from the USB host to the device in the SOF packet. The TSTMP register is updated with the value sent by the host each time a successful SOF packet is detected by the device. The SOF_GEN interrupt source bit (see page 4-8) can be configured to generate an interrupt, if it is not masked, when an SOF packet from the host is detected on the bus or when the SOF packet is not detected and an internal SOF is generated.

When the USB hardware determines that the SOF packet was missed, this register value is incremented by one. The MS_SOF interrupt source bit (see page 4-8) can be configured to generate an interrupt, if it is not masked, when this happens.

Bit Definitions

Bit	Name	Function
15–11	Res	**Reserved** For compatibility with future devices, always write this bit field with its chip reset default value.
10–0	TSTMP	**Time Stamp** The TSTMP bit field contains the current USB frame number. This bit field is updated with the frame number value received in the SOF packet each time the device successfully receives an SOF packet from the USB host. If an SOF packet is not detected successfully by the device, this bit field is incremented by one.

Programming Notes

Time Stamp Match (TSTMPM) Offset 1EEh

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	*Res*					TSTMPM										
Software Read/Write	R					R/W										
Hardware Set/Clear	—					S/C										
Chip Reset Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Microcontroller:
Am186CC ☑
Am186CH □
Am186CU ☑

Register Description

The device software can use the TSTMPM register to trap a specific USB time stamp/frame number. The value written to this register is compared to the current frame number value in the TSTMP register (see page 4-13) each time the TSTMP register is updated. If the value in the TSTMP register is equal to or greater than the value in the TSTMPM register, the TSTMP_M bit in the UISTAT2 register is set (see page 4-7). The TSTMP_M bit can be enabled as an interrupt source by setting the corresponding bit in the UIMASK2 register (see page 4-9).

Hardware cannot set the TSTMP_M bit in UISTAT2 again until software has cleared the bit and written to the TSTMPM register again. In other words, writing this register acts as an “arming” function for the TSTMP_M bit.

Note that SOF packets can be missed by the device in any given frame. If an SOF packet is missed, the TSTMP register is incremented by one.

Bit Definitions

Bit	Name	Function
15–11	Res	**Reserved** For compatibility with future devices, always write this bit field with its chip reset default value.
10–0	TSTMPM	**Time Stamp Match Value** The TSTMPM bit field contains the USB frame number match value. When the value in the TSTMP register is greater than or equal to the value in this register, a TSTMP_M bit interrupt is generated if its source is enabled.

Programming Notes

In addition to setting bits in the UIMASK2 register, software must configure the interrupt Channel 2 Control (CH2CON) register to select the interrupt channel’s internal source (USB) and enable the channel before a TSTMP_M bit interrupt can be generated. See Chapter 9, “Interrupt Controller Registers”.

Isochronous Synchronization Control (ISCTL) Offset 1F0h

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	ESOF_EN	*Res*					BYTES_SAM		*Res*			BCNT_LRATE			SAM_CLK_SEL	
Software Read/Write	R/W	R					R/W		R			R/W			R/W	
Hardware Set/Clear	—	—					—		—			—			—	
Chip Reset Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Microcontroller:	
Am186CC	✔
Am186CH	
Am186CU	!

! = *See bit definitions.*

Register Description

The device software uses the ISCTL register to control the operation of the FPMCNT register and to select the data sample clock for the auto rate control feature. This register and the FPMCNT register (see page 4-17) are used when the device is supporting isochronous IN endpoints and is acting as an SOF master.

In hardware, a counter is implemented that is incremented once per USB data bit clock (nominally 12 MHz). This counter is reset during each USB SOF time. The FPMCNT register latches the value of this USB bit counter each time a preselected number of data source sample clocks is detected. The device software can then read the latched USB bit clock value each time it is updated to determine how the source data rate and the USB data rate are moving relative to each other. The USB host's SOF rate or data rate can then be adjusted accordingly.

The device software also uses the ISCTL register to globally select the number of bytes per data source sample for the auto rate control mechanism for endpoints A, B, C, and D.

This register is also used to enable the USB SOF indication on the UCLK pin.

Bit Definitions

Bit	Name	Function
15	ESOF_EN	**External SOF Enable** This bit can be used to configure the UCLK pin to drive out the USB Start of Frame (USBSOF) signal. 0 = The UCLK pin is configured as an input, used either as the UART baud rate clock source (UCLK signal), or as an external USB sample clock (USBSCI signal, selected by the SAM_CLK_SEL bit in this register). 1 = The UCLK pin is configured as the USBSOF output. When configured as USBSOF, the UCLK pin is pulsed by the device when either a good SOF packet has been received from the USB host or when the device hardware has detected that the SOF packet has been missed on the bus. This bit should not be set if the UCLK pin's UART external clock function is enabled by the XTRN bit in either the HSPCON1 register or the SPCON1 register (see page 5-6 and page 5-30).
14–10	Res	**Reserved** For compatibility with future devices, always write this bit field with its chip reset default value.

Bit	Name	Function
9–8	BYTES_SAM	**Bytes Per Sample** If any of the endpoints A–D are configured for isochronous IN transfers and their associated AUTO_RATE_EN bit is set, the BYTES_SAM bit field determines the number of bytes per data source sample that are moved in each data transfer for that endpoint. This bit field is encoded as follows: 00 = Multiply by 1 01 = Multiply by 2 10 = Multiply by 4 11 = Reserved Note that the configured maximum packet size for the endpoint must be greater than or equal to the largest number of data bytes that might be moved by the endpoint during a USB transaction.
7–5	Res	**Reserved** For compatibility with future devices, always write this bit field with its chip reset default value.
4–2	BCNT_LRATE	**USB Bit Count Latch Rate** The BCNT_LRATE bit field is used to select how frequently (i.e., after how many data source sample clocks) the USB bit counter value is latched into the FPMCNT register. This bit field is encoded as follows: 000 = 1 001 = 2 010 = 4 011 = 8 100 = 16 101 = 32 110 = 64 Note that software can configure an interrupt to be generated upon the FPMCNT register being updated.
1–0	SAM_CLK_SEL	**Sample Clock Source Select** These bits are used to select between two different data source sample clock sources. 00 = Disabled 01 = USBSCI input signal on the UCLK pin 10 = Channel A, PCM Highway/GCI Frame Sync 11 = Disabled ***Am186CU Microcontroller:*** *The value 10b is not supported in the Am186CU USB microcontroller.* When ESOF_EN = 0 in this register, the UCLK pin can be used as an input either for sample clock source (USBSCI) or the UART external clock function, which is enabled by the XTRN bit in either the HSPCON1 register or the SPCON1 register (see page 5-6 and page 5-30).

Programming Notes

Frame Position Monitor Count (FPMCNT) Offset 1F2h

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	*Res*		FPM_CNT													
Software Read/Write	R		R													
Hardware Set/Clear	—		S/C													
Chip Reset Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Microcontroller:	
Am186CC	✓
Am186CH	□
Am186CU	✓

Register Description

Device software uses the FPMCNT register to read the value of the latched USB bit counter. This register and the ISCTL register (see page 4-15) are used when the device is supporting isochronous IN endpoints and is acting as an SOF master. Hardware latches the value of the USB bit counter each time a preselected number of data source sample clocks is detected. The device software can then read the latched USB bit clock value each time it is updated to determine how the source data rate and the USB data rate are moving relative to each other. The USB host's SOF rate or data rate can then be adjusted accordingly.

Bit Definitions

Bit	Name	Function
15–14	Res	**Reserved** For compatibility with future devices, always write this bit field with its chip reset default value.
13–0	FPM_CNT	**Frame Position Monitor Count** The FPM_CNT bit field is read by the device software to determine the position of the data source data rate relative to the USB data rate. This value is updated each time a programmable number of data source sample clocks (set in the ISCTL register, see page 4-15) is detected. Note that software can configure an interrupt to be generated upon this register being updated. The device software can sample this value over many USB frame times to determine if the data source data rate and the USB data rate are moving relative to each other and adjust the USB host's SOF accordingly. The device software is required to read the same value from this register in two consecutive peripheral control block (PCB) read accesses before software can assume that the value is valid.

Programming Notes

Control Endpoint Control/Status (CNTCTL) Offset 200h

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	EP_EN	EP_NOT_STALLED	NOT_FLUSH	ACT_REQ	NEW_COMMAND	COMMAND_BUSY	*Res*									
Software Read/Write	R/W	R/W0	R/W0	R/W0	R/W0	R/W0	R									
Hardware Set/Clear	—	S/C	S/C	S/C	S/C	S/C	—									
Chip Reset Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Microcontroller:

Am186CC ☑
Am186CH □
Am186CU ☑

Register Description

The device software uses the CNTCTL register to communicate with the control endpoint (Endpoint 0).

Bit Definitions

Bit	Name	Function
15	EP_EN	**Endpoint Enable** The EP_EN bit is used to enable or disable the control endpoint. When this bit is cleared, the endpoint does not respond to any USB host activity. When this bit is set, this endpoint is enabled and operated as defined here. When this bit is cleared, all other bits in this register are cleared.
14	EP_NOT_STALLED	**Endpoint is Not Stalled** The EP_NOT_STALLED bit is used by the endpoint handling software to configure the control endpoint to be stalled. Per the USB protocol definition regarding control endpoints, this bit only has an effect during the data and status stages of a control transaction. This bit can only be cleared by software if the NEW_COMMAND bit is cleared. If the EP_NOT_STALLED bit is cleared during the data or status stages of a control transaction to this endpoint, a stall handshake is sent to the host on the USB in response to an OUT or IN packet being sent by the host. The EP_NOT_STALLED bit is ignored and automatically set in hardware upon the host sending a setup packet to the control endpoint (i.e., when the NEW_COMMAND bit is set). The EP_NOT_STALLED bit is set in this manner when a setup packet is received independent of the command type (i.e., some commands are handled in hardware alone and are not decoded by the device software). Note that this bit is cleared by the hardware at the termination of a control transfer.
13	NOT_FLUSH	**Endpoint Buffer is Not Flushed** The NOT_FLUSH bit is used by the endpoint handling software to flush the endpoint buffer (i.e., reset its read and write address pointer to 0d). Clearing this bit only flushes the endpoint's buffer if the ACT_REQ bit is set and the NEW_COMMAND bit is cleared. Software can always clear this bit before writing a new packet data to the CNTLDAT register. This ensures the removal of any stale data that might remain from a previous operation. Received data is preserved if software happens to clear this bit after the NEW_COMMAND bit is set. This bit always reads back the same value as the EP_EN bit.

Bit	Name	Function
12	ACT_REQ	**Endpoint Buffer Action Required Status** This bit provides status about the endpoint's data buffer ownership to the device software. This bit is automatically set after a valid and complete USB setup or data packet is received from the USB host during a control transaction. This bit is also set after a complete USB data packet has been transmitted to the USB host during a control transaction, and the host has acknowledged the packet. When this bit is set and the NEW_COMMAND bit is cleared, the software is able to write or read data to/from the endpoint data buffer. When finished with this activity, the software must clear this bit thus giving control back to the USB endpoint hardware; this allows the hardware to continue to transmit or receive data packets on the USB for this endpoint. Attempts by device software to clear this bit fail if the NEW_COMMAND bit is set. When this bit is cleared, device software accesses to the data buffer port have no effect. Software can enable the control endpoint's ACT_REQ bit to generate an interrupt by setting the CNT_EP_ACT bit in the UIMASK1 register (see page 4-4). This bit is automatically cleared by hardware when a command is aborted (i.e., when another setup packet is detected) or when the endpoint is stalled during the control transfer data stage or status stage (i.e., if the EP_NOT_STALLED bit is cleared during this time). For example, if the host transmits an OUT token during an IN transaction, the USB controller hardware automatically aborts the IN transaction, flushes the buffer, and clears the ACT_REQ bit. In this situation no device software action is required, so the ACT_REQ bit is not set.
11	NEW_COMMAND	**New Command Received Status** This bit provides status about the received USB data packet type to the device software. When the setup stage of the control transfer is detected (before any data has been written to the FIFO), this bit is set by hardware. This bit is not set upon successful reception of packets during the data stage or the status stage of a control transfer. After this bit is set by hardware and the ACT_REQ bit is set, the endpoint handling software can detect, by sampling this bit, that the packet received is a setup packet to be decoded. At this point, software must clear this bit and the device software will have read/write access to the endpoint's data buffer (software would read the data buffer in this case, decode the setup packet, and respond appropriately). Attempts by device software to clear the NEW_COMMAND bit fail unless the ACT_REQ bit is set. Software can enable an endpoint's NEW_COMMAND bit to generate an interrupt by setting the CNT_EP_NEW bit in the UIMASK1 register (see page 4-4). ***Note:*** *This bit is also set by the hardware when a setup packet/command handled entirely by the endpoint hardware is detected. Software is not required to take any action upon this type of command being received unless the CNT_EP_NEW interrupt source is enabled.*
10	COMMAND_BUSY	**Current Command Busy Status** This bit provides status about the state of the current command being processed by the control endpoint handling software. This bit is set automatically in hardware whenever a new setup packet is received by the endpoint, after the device hardware has acknowledged the setup packet to the host. For a control write transfer, software must clear this bit when it has received all data associated with the command and has completed all processing associated with the received command. The status stage of the control transfer is not completed, and thus the control transfer is not completed, on the USB until this bit is cleared. This bit is automatically cleared by hardware when a command is aborted (i.e., a setup packet is detected prior to the status stage of the previous command completing), when a control read transfer's status stage is completed, or when the endpoint is stalled during the control transfer data stage or status stage (i.e., if the EP_NOT_STALL bit is cleared during this time). Attempts by device software to clear the COMMAND_BUSY bit fail if the NEW_COMMAND bit is set.
9–0	Res	**Reserved** For compatibility with future devices, always write this bit field with its chip reset default value.

Programming Notes

All the bits in this register, except for the EP_EN and NOT_FLUSH bits, are cleared if any one of the following actions occurs:

- A USB reset condition is detected.
- A USB suspend condition is detected.

- The EP_NOT_STALLED bit is cleared by software during a data stage or status stage transfer.
- The EP_NOT_STALLED bit is cleared by hardware at the termination of a control transfer.

All the bits in the register are cleared if the EP_EN bit is cleared by software.

All the bits in this register are cleared by an external or watchdog timer reset. When software wants to enable this endpoint, it typically writes a value of FFFFh to this register. During this write, with the EP_EN bit transitioning from Low to High, hardware forces the NOT_FLUSH bit to be set and the ACT_REQ, NEW_COMMAND, and COMMAND_BUSY bits to be cleared regardless of the value that is written to these bit positions.

As a special case, software might want to enable the endpoint in its stalled condition (i.e., with the EP_NOT_STALLED bit cleared). This can be accomplished during the initial write operation to this register if the EP_NOT_STALLED bit position is written as a value of 0b with the EP_EN bit transitioning from a value of 0 to a value of 1. This allows software to enable the endpoint in the stalled state.

The EP_NOT_STALLED bit cannot be cleared by software at the same time as the EP_EN bit is being set if the EP_EN bit is currently set. For the control endpoint, the EP_NOT_STALLED bit is automatically set when the first setup packet is received from the host.

Control Endpoint Receive Packet Size (CNTSIZ) Offset 202h

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	*Res*												RCV_PKT_SIZE			
Software Read/Write	R												R			
Hardware Set/Clear	—												S/C			
Chip Reset Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Microcontroller:	
Am186CC	✓
Am186CH	
Am186CU	✓

Register Description

The device software uses the CNTSIZ register to determine the received packet size for the control endpoint.

Bit Definitions

Bit	Name	Function
15–4	Res	**Reserved** For compatibility with future devices, always write this bit field with its chip reset default value.
3–0	RCV_PKT_SIZE	**Received Packet Size (in bytes)** The RCV_PKT_SIZE bit field provides status regarding the received packet size (in bytes) to the endpoint handling software when a valid packet has been received from the USB host and resides in the data buffer. Software can use this bit field to determine the amount of valid data in the buffer prior to performing the read operations. The valid range for this bit field is as follows: 1000 = 8 bytes in data buffer 0111 = 7 bytes in data buffer 0110 = 6 bytes in data buffer 0101 = 5 bytes in data buffer 0100 = 4 bytes in data buffer 0011 = 3 bytes in data buffer 0010 = 2 bytes in data buffer 0001 = 1 bytes in data buffer 0000 = 0 bytes in data buffer This bit field contains the valid size, in bytes, of the received data packet when the ACT_REQ bit located in the CNTCTL register is set (see page 4-19).

Programming Notes

All the bits in this register are cleared if any one of the following actions occurs:

- A USB reset condition is detected.
- A USB suspend condition is detected.
- The EP_NOT_STALLED bit in the CNTCTL register is cleared by software during a data stage or status stage transfer.
- The EP_EN bit in the CNTCTL register is cleared by software.
- An external or watchdog timer reset occurs.

Control Endpoint Data Port (CNTDAT) Offset 206h

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	*Res*								D							
Software Read/Write	R								R/W							
Hardware Set/Clear	—								S/C							
Chip Reset Default	0	0	0	0	0	0	0	0	x	x	x	x	x	x	x	x

Microcontroller:	
Am186CC	✓
Am186CH	□
Am186CU	✓

Register Description

The device software uses the CNTDAT register to read and write the control endpoint first-in-first-out (FIFO) data buffer.

Bit Definitions

Bit	Name	Function
15–8	Res	**Reserved** For compatibility with future devices, always write this bit field with its chip reset default value.
7–0	D	**Control Endpoint Data Port** The D bit field is used by the control endpoint handling software to read and write data to be received by and transmitted from the control endpoint. Access to this data port is controlled by the control endpoint hardware in conjunction with the device software's use of the control endpoint status bits located in the CNTCTL register. See the CNTCTL register description starting on page 4-18 for all details about the device software access limitations to the data port. All device software accesses to this FIFO are ignored unless EP_EN is set, NEW_COMMAND is cleared, and ACT_REQ is set.

Programming Notes

Control Endpoint Receive Data Port Peek (CNTRPK) Offset 208h

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	*Res*								D							
Software Read/Write	R								R							
Hardware Set/Clear	—								S/C							
Chip Reset Default	0	0	0	0	0	0	0	0	x	x	x	x	x	x	x	x

Microcontroller:

Am186CC ☑

Am186CH □

Am186CU ☑

Register Description

Debug or emulator software uses the CNTRPK register to read the control endpoint data buffer without advancing the endpoint's data buffer address pointer.

Bit Definitions

Bit	Name	Function
15–8	Res	**Reserved** For compatibility with future devices, always write this bit field with its chip reset default value.
7–0	D	**Control Endpoint Receive Data Peek** The D bit field can be used by debug or emulator software to read data from the control endpoint without advancing the endpoint's data buffer address pointer. Access to this port is controlled by the same mechanisms that are used to control the control endpoint's data port register. See the CNTCTL register description starting on page 4-18 for all details about the device software access limitations to the data port.

Programming Notes

Control Endpoint Definition 1 (CNTDEF1) Offset 20Ah

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	EP_NUM				EP_CFG		*Res*	EP_INT		*Res*	EP_ASET			EP_DIR	EP_TYPE	
Software Read/Write	R				R		R	R		R	R			R	R	
Hardware Set/Clear	—				—		—	—		—	—			—	—	
Chip Reset Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Microcontroller:	
Am186CC	✔
Am186CH	□
Am186CU	✔

Register Description

This register is used to read the control endpoint definition. The hardware associated with the control endpoint is fixed in terms of its USB definition. Software can only read this register.

Bit Definitions

Bit	Name	Function
15–12	EP_NUM	**Endpoint Number** The EP_NUM bit field is used to set the USB endpoint number for the control endpoint. This bit field is fixed to a value of 0000b.
11–10	EP_CFG	**Endpoint Configuration** The EP_CFG bit field is used to relate this endpoint to a configuration. The control endpoint is considered a member of all configurations, and this bit field is fixed to a value of 00b.
9	Res	**Reserved** For compatibility with future devices, always write this bit field with its chip reset default value.
8–7	EP_INT	**Endpoint Interface** The EP_INT bit field is used to relate this endpoint to an interface within the selected configuration. The control endpoint is considered a member of all interfaces, and this bit field is fixed to a value of 00b.
6	Res	**Reserved** For compatibility with future devices, always write this bit field with its chip reset default value.
5–3	EP_ASET	**Alternate Setting** The EP_ASET bit field is used to relate this endpoint to an interface alternate setting. The control endpoint is considered a member of all alternate settings, and this bit field is fixed to a value of 000b.
2	EP_DIR	**Endpoint Direction (IN or OUT)** The EP_DIR bit is used to select the direction for this endpoint. Control endpoints are considered to be bidirectional, and this bit field is fixed to a value of 0b.
1–0	EP_TYPE	**Endpoint Type Selection** The EP_TYPE bit field is used to select the type of USB transfers that this endpoint supports. 00 = Control 01 = Isochronous (not valid for this endpoint) 10 = Bulk (not valid for this endpoint) 11 = Interrupt (not valid for this endpoint) For the control endpoint, this bit field is fixed to 00b.

Programming Notes

Control Endpoint Definition 2 (CNTDEF2) Offset 20Ch

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	*Res*				FIFO_SIZE		EP_MX_PCT									
Software Read/Write	R				R		R									
Hardware Set/Clear	—				—		—									
Chip Reset Default	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0

Microcontroller:
Am186CC ☑
Am186CH □
Am186CU ☑

Register Description

This register is used to read the control endpoint’s FIFO size and maximum packet size.

Bit Definitions

Bit	Name	Function
15–12	Res	**Reserved** For compatibility with future devices, always write this bit field with its chip reset default value.
11–10	FIFO_SIZE	**FIFO Size** The FIFO_SIZE bit field contains the encoded size of the FIFO associated with the endpoint. 00 = 8 bytes deep 01 = 16 bytes deep 10 = 32 bytes deep 11 = 64 bytes deep For the control endpoint, this bit field is fixed to a value of 00b.
9–0	EP_MX_PCT	**Endpoint Max Packet Value** The EP_MX_PCT bit field is used to select the maximum packet value for the control endpoint. For the control endpoint, this bit field is fixed to a value of 8d.

Programming Notes

Interrupt Endpoint Control/Status (IEPCTL) Offset 210h

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	EP_EN	EP_NOT_STALLED	NOT_FLUSH	ACT_REQ	*Res*											
Software Read/Write	R/W	R/W0	R/W0	R/W0	R											
Hardware Set/Clear	—	S/C	S/C	S/C	—											
Chip Reset Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Microcontroller:

Am186CC ☑
Am186CH □
Am186CU ☑

Register Description

The device software uses the IEPCTL register to communicate with the interrupt endpoint.

Bit Definitions

Bit **Name** **Function**

15 EP_EN **Endpoint Enable**

The EP_EN bit is used to enable or disable the interrupt endpoint. When this bit is cleared, the endpoint does not respond to any USB data transactions. When set, this endpoint is enabled and operates as defined here.

When this bit is cleared, all other bits in this register are cleared.

If the interrupt endpoint is enabled, and then disabled by clearing the EP_EN bit, software must also clear the EP_NUM bit in the IEPDEF1 register (see page 4-30).

14 EP_NOT_STALLED **Endpoint is Not Stalled**

The EP_NOT_STALLED bit is used by the endpoint handling software to configure the interrupt endpoint to be stalled. If this bit is cleared by software, then the reception of a token packet by the endpoint causes a stall handshake to be sent to the host.

If this bit is cleared by software during the data phase of an interrupt transaction (to this endpoint) a negative acknowledgment (NAK) is returned during the data phase of the transaction.

This bit is set by hardware upon the host sending a CLEAR_FEATURE command when the command explicitly contains:

- The endpoint number associated with this interrupt endpoint
- The ENDPOINT_STALL feature selector

13 NOT_FLUSH **Endpoint Buffer is Not Flushed**

The NOT_FLUSH bit is used by the endpoint handling software to flush the endpoint buffer (i.e., reset the buffer’s read and write address pointers to 0d).

Clearing this bit when the ACT_REQ bit is set guarantees that the buffer is flushed. If the ACT_REQ bit is cleared, clearing this bit causes the buffer to be flushed and the ACT_REQ bit to be set only if there is not an active data transaction from this endpoint to the USB host. This feature allows device software to update stale data that has not been sent to the USB host if necessary.

This bit always reads back the same value as the EP_EN bit.

Bit	Name	Function
12	ACT_REQ	**Endpoint Buffer Action Required Status** The ACT_REQ bit provides status about the endpoint's data buffer ownership to the device software. This bit is set by hardware after the endpoint has successfully sent a data packet to the host and the host has acknowledged it. When this bit is set, the device software is able to write data to the endpoint data buffer. Only 8 or 16 bytes can be written based on the maximum packet value that is configured for this endpoint. If more data is written when the ACT_REQ bit is set, these writes have no effect. When completed with the write activity, the device software must clear the ACT_REQ bit, thus giving control back to the USB endpoint hardware. This allows the hardware to transmit data that was written to the data buffer to the host when the host requests it. When this bit is cleared, device software accesses to the data buffer port have no effect. Software can enable the interrupt endpoint's ACT_REQ bit to generate an interrupt by setting the INT_EP_ACT bit in the UIMASK1 register (see page 4-5).
11–0	Res	**Reserved** For compatibility with future devices, always write this bit field with its chip reset default value.

Programming Notes

All the bits in this register are cleared if the EP_EN bit is cleared.

All the bits in this register are cleared upon an external or watchdog timer reset. When software wishes to enable this endpoint, it typically writes a value of FFFFh to this register. During this write, with the EP_EN bit transitioning from Low to High, hardware forces the NOT_FLUSH and ACT_REQ bits to be set regardless of the value written to these bit positions.

As a special case, software might want to enable the endpoint in its stalled condition (i.e., with the EP_NOT_STALLED bit cleared). This can be accomplished if (during the initial write operation to this register, with the EP_EN bit transitioning from a value of 0 to a value of 1) the EP_NOT_STALLED bit position is written as a value of 0b. This allows software to cause the endpoint to be in the stalled state when it is initially enabled.

The EP_NOT_STALLED bit cannot be cleared by software at the same time as the EP_EN bit is being set if EP_EN is already set. For this endpoint, the EP_NOT_STALLED bit is automatically set when the USB host issues a CLEAR_FEATURE command that specifies the stall option and this endpoint's number.

If the EP_EN bit is set, all the non-reserved bits in this register are set if any one of the following occurs:

- A USB reset condition is detected.
- A USB suspend condition is detected.

Interrupt Endpoint Data Port (IEPDAT) Offset 216h

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	*Res*								D							
Software Read/Write	R								R/W							
Hardware Set/Clear	—								S/C							
Chip Reset Default	0	0	0	0	0	0	0	0	x	x	x	x	x	x	x	x

Microcontroller:

Am186CC ☑
Am186CH □
Am186CU ☑

Register Description

The device software uses the IEPDAT register to read and write the interrupt endpoint data buffer.

Bit Definitions

Bit	Name	Function
15–8	Res	**Reserved** For compatibility with future devices, always write this bit field with its chip reset default value.
7–0	D	**Interrupt Endpoint Data Port** The D bit field is used by the interrupt endpoint handling software to write data to be transmitted from the interrupt endpoint. Access to this data port is controlled by the interrupt endpoint hardware in conjunction with the device software's use of the interrupt endpoint status bits located in the IEPCTL register. See the IEPCTL register description starting on page 4-27 for all details about the device software access limitations to the data port.

Programming Notes

Interrupt Endpoint Definition 1 (IEPDEF1) Offset 21Ah

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	EP_NUM				EP_CFG		*Res*	EP_INT		*Res*	EP_ASET			EP_DIR	EP_TYPE	
Software Read/Write	R/W				R/W		R	R/W		R	R/W			R	R	
Hardware Set/Clear	—				—		—	—		—	—			—	—	
Chip Reset Default	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	1

Microcontroller:	
Am186CC	✓
Am186CH	
Am186CU	✓

Register Description

The device software uses the IEPDEF1 register to define the interrupt endpoint.

Bit Definitions

Bit	Name	Function
15–12	EP_NUM	**Endpoint Number** The EP_NUM bit field is used to set the USB endpoint number for the interrupt endpoint. The endpoint number for this endpoint should not be programmed to a value of 0d if the endpoint is enabled. Additionally, no two enabled endpoints within the device should share the same number. If the interrupt endpoint is enabled, and then disabled by clearing the EP_EN bit in the IEPCTL register (see page 4-27), software must also clear the EP_NUM bit.
11–10	EP_CFG	**Endpoint Configuration** The EP_CFG bit field is used by software to relate this endpoint to a configuration that the USB host selected by issuing a SET_CONFIGURATION command. Configurations 0d–3d are supported.
9	Res	**Reserved** For compatibility with future devices, always write this bit field with its chip reset default value.
8–7	EP_INT	**Endpoint Interface** The EP_INT bit field is used to relate this endpoint to the host-selected interface within the selected configuration. Interfaces 0d–3d are supported.
6	Res	**Reserved** For compatibility with future devices, always write this bit field with its chip reset default value.
5–3	EP_ASET	**Alternate Setting** The EP_ASET bit field is used to relate this endpoint to the host-selected interface alternate setting. Each interface supports a total of eight alternate settings (0d–7d).
2	EP_DIR	**Endpoint Direction (IN or OUT)** The EP_DIR bit is used to select the direction for this endpoint. 0 = IN 1 = OUT The interrupt endpoint is an IN endpoint, and this bit is fixed to a value of 0b.
1–0	EP_TYPE	**Endpoint Type Selection** The EP_TYPE bit field is used to select the type of USB transfers that this endpoint supports. 00 = Control (not valid for this endpoint) 01 = Isochronous (not valid for this endpoint) 10 = Bulk (not valid for this endpoint) 11 = Interrupt For the interrupt endpoint, this bit field is fixed to 11b.

Programming Notes

Host software should only set configuration and interface values that match a device descriptor returned by the device in response to a GET_DESCRIPTOR command. However, the USB hardware accepts as valid any configuration or feature setting in the range of 0d–3d, regardless of the available descriptors. To help ensure reliable operation in any USB environment, device software can define a minimal descriptor (i.e., Endpoint 0 with no bandwidth allocation) for any configuration and interface settings that it does not define otherwise.

Interrupt Endpoint Definition 2 (IEPDEF2) Offset 21Ch

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	*Res*				FIFO_SIZE		EP_MX_PCT									
Software Read/Write	R				R		R					R/W	R	R		
Hardware Set/Clear	—				—		—					—	S/C	—		
Chip Reset Default	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0

Microcontroller:
Am186CC ☑
Am186CH ☐
Am186CU ☑

Register Description

The device software uses the IEPDEF2 register to set the interrupt endpoint's maximum packet size.

Bit Definitions

Bit	Name	Function
15–12	Res	**Reserved** For compatibility with future devices, always write this bit field with its chip reset default value.
11–10	FIFO_SIZE	**FIFO Size** The FIFO_SIZE bit field contains the encoded size of the FIFO associated with the endpoint. 00 = 8 bytes deep 01 = 16 bytes deep 10 = 32 bytes deep 11 = 64 bytes deep For the interrupt endpoint, this bit field is fixed to a value of 01b.
9–0	EP_MX_PCT	**Endpoint Max Packet Value** The EP_MX_PCT bit field is used to select the maximum packet value for the interrupt endpoint. For the interrupt endpoint, this bit field can be programmed to a value of 16d or 8d. This is accomplished by setting or clearing bit 4. Bit 3 always reads back the inverted state of bit 4.

Programming Notes

A Endpoint Control/Status (AEPCTL) Offset 220h
B Endpoint Control/Status (BEPCTL) Offset 230h
C Endpoint Control/Status (CEPCTL) Offset 240h
D Endpoint Control/Status (DEPCTL) Offset 250h

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	EP_EN	EP_NOT_STALLED	NOT_FLUSH	ACT_REQ	STAT_INT	*Res*	NOT_ZERO	NOT_LAST_BYTE	*Res*	ISO_START	ISO_STOP	ISO_MS	FULL_PKT	SHORT_PKT	BUF_ERR	OTHER_ERR
Software Read/Write	R/W	R/W0	R/W0	R/W0	R/W0	R	R/W0	R/W0	R	R	R	R/W0	R/W0	R/W0	R/W0	R/W0
Hardware Set/Clear	—	S/C	S/C	S/C	S/C	—	S/C	S/C	—	S/C	S/C	S/C	S/C	S/C	S/C	S/C
Chip Reset Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Microcontroller:
Am186CC ☑
Am186CH □
Am186CU ☑

Register Description

The device software uses the xEPCTL register to control the endpoint and track its status during its operation.

Some bit field descriptions depend on the endpoint's mode of operation (DMA or non-DMA), as set in the Endpoint Definition 3 register (see page 4-50) and direction (IN or OUT) as set in the Endpoint Definition 1 register (page 4-44).

Bit Definitions

Bit **Name** **Function**

15 EP_EN **Endpoint Enable**
The EP_EN bit is used to enable or disable the endpoint. When this bit is cleared, the endpoint does not respond to any USB data transactions. When set, this endpoint is enabled and operates as defined here.

When this bit is cleared all other bits in this register are cleared.

If an enabled endpoint is disabled by clearing the EP_EN bit, software must also clear the EP_NUM bit in the xEPDEF1 register for that endpoint.

14 EP_NOT_STALLED **Endpoint is Not Stalled**
The EP_NOT_STALLED bit is used by the endpoint handling software to configure the endpoint to be stalled. This bit is ignored if the endpoint is configured to be of type isochronous.

If this bit is cleared by software, the reception of a token packet by the endpoint causes a stall handshake to be sent to the host.

If this bit is cleared by software during the data phase of a transaction to this endpoint, the endpoint immediately becomes idle.

This bit is set by hardware upon the host sending a CLEAR_FEATURE command when the command explicitly contains:

- The endpoint number associated with this endpoint
- The ENDPOINT_STALL feature selector

Bit	Name	Function
13	NOT_FLUSH	**Endpoint Buffer is Not Flushed** The NOT_FLUSH bit is used by the endpoint handling software to flush the endpoint buffer (i.e., reset the buffer's read and write address pointers to 0d). ***Non-DMA mode/OUT direction*** Clearing this bit when the ACT_REQ bit is set guarantees that the buffer is flushed. Clearing this bit when the ACT_REQ bit is cleared has no effect. ***Non-DMA mode/IN direction*** Clearing this bit when the ACT_REQ bit is set guarantees that the buffer is flushed. When the ACT_REQ bit is cleared, clearing this bit causes the buffer to be flushed and the ACT_REQ bit to be set only if there is not an active data transaction from this endpoint to the USB host. This feature allows device software to update stale data that has not been sent to the USB host. ***DMA mode*** Clearing this bit causes the endpoint buffer read and write pointers to be reset to 0d. Any active USB transaction to the endpoint is halted. If a transaction is active when this bit is cleared, the device-to-host handshake phase consists of the NAK packet for a bulk OUT transaction and a no response handshake from the host to the device for a bulk IN transaction. In addition, the EP_NOT_STALLED bit is forced to be set, if cleared, when the NOT_FLUSH bit is cleared. This bit always reads back the same value as the EP_EN bit.
12	ACT_REQ	**Endpoint Buffer Action Required Status** ***Non-DMA mode/OUT direction*** In this mode, the ACT_REQ bit provides status about the endpoint's data buffer ownership to the device software. This bit is set by hardware after the endpoint has successfully received a data packet from the USB host, and the endpoint has acknowledged it. When this bit is set, the device software is able to read data from the endpoint data buffer. The device software can only read the amount of valid data that is present in the buffer as indicated in the Endpoint Received Packet Size register (see page 4-39). If the device software attempts more read accesses than there is valid data in the buffer, the additional reads return invalid data. When completed with the read activity, the device software must clear the ACT_REQ bit, thus giving control back to the USB endpoint hardware. This allows the hardware to receive the next data sent to it from the USB host. When this bit is cleared, device software read accesses from the data buffer port return invalid data. ***Non-DMA mode/IN direction*** The ACT_REQ bit provides status about the endpoint's data buffer ownership to the device software. This bit is set by hardware when the endpoint is initially enabled, and after the endpoint has successfully sent a data packet to the USB host and the host has acknowledged it. When this bit is set, the device software is able to write data to the endpoint data buffer at the endpoint's data port address location in the PCB address space. The device software can only successfully write as many bytes as the physical size of the buffer. When finished with write activity, the device software must clear the ACT_REQ bit, thus giving control back to the USB endpoint hardware. This allows the hardware to send the data to the USB host when it is requested. If the NOT_FLUSH bit is cleared by software prior to the USB host request for this data, the ACT_REQ bit is set and the existing data in the buffer is flushed. This gives control of the buffer back to the device software, allowing software to update the data if it has become stale. ***DMA mode*** When the endpoint is configured for DMA mode, this bit, when set, causes the endpoint control hardware to stop. When the endpoint is configured for the OUT direction, five conditions can be individually selected that cause this bit to be set. These are: isochronous packet missed, full packet received, short packet received, buffer error, and other error. When the endpoint is configured for the IN direction, three conditions can be individually selected that cause this bit to be set. These are: isochronous packet missed, buffer error, and other error. The selection of these stop conditions is configured in the STOP_MASK bit field of the endpoint's Definition 3 register (see page 4-50). The individual stop conditions can be tested by sampling bits 4 through 0 of the xEPCTL register. In addition to the three selectable conditions, an IN endpoint's ACT_REQ bit is also set if a USB suspend or USB reset is detected. These conditions are indicated by the USB_SUS and USB_RST bits in the UISTAT2 register (see page 4-7). When this bit is set and the endpoint controller is stopped, software must clear this bit to allow the hardware to respond to the next request from the USB host for this endpoint. ***General*** Software can enable an endpoint's ACT_REQ bit to generate an interrupt, when set, by setting the appropriate x_EP_ACT bit in the UIMASK1 register (see page 4-3 and page 4-5). To avoid a false interrupt from being generated, software must clear all of the stop conditions prior to or at the same time as the ACT_REQ bit is cleared.

Bit	Name	Function
11	STAT_INT	**Status Interrupt** The STAT_INT bit is set when any one of the bits 4 through 0 of this register are set and the corresponding bit in the INT_MASK bit field of the endpoint's Definition 3 register is set (see page 4-50). Software can enable an endpoint's STAT_INT bit to generate an interrupt, when set, by setting the appropriate x_EP_STATINT bit in the UIMASK1 register (see page 4-3 and page 4-5). To avoid a false interrupt from being generated, software must clear all of the stop conditions prior to or at the same time as the STAT_INT bit is cleared.
10	Res	**Reserved** For compatibility with future devices, always write this bit field with its chip reset default value.
9	NOT_ZERO	**Not Zero Data Packet** The NOT_ZERO bit is used to place a zero length data packet in the endpoint's buffer. This feature is typically used when the endpoint is configured for the general-purpose DMA mode. Software should clear this bit and write a dummy data byte to the endpoint data port. This bit is automatically set in hardware after the dummy write to the data port is detected. This action should never be performed unless the DMA channel being used for this endpoint is disabled. This bit has no functional effect when the endpoint is configured for its non-DMA mode or is configured for the OUT direction. Refer to the programming notes for this register (on page 4-38) for conditions that set this bit.
8	NOT_LAST_BYTE	**Not Last Byte of Packet** The NOT_LAST_BYTE bit is used by the device software to mark the last byte of a packet that is to be placed in the endpoint's buffer. This feature is useful when the endpoint is configured for the general-purpose DMA mode and the IN direction when the device software is directly managing the data movement into the endpoint's data buffer. Software should clear this bit and write the last data byte to the endpoint data port. This bit is automatically set in hardware after the write to the data port is detected. This action should never be performed unless the DMA channel being used for this endpoint is disabled. This bit has no functional effect when the endpoint is configured for its non-DMA mode or is configured for the OUT direction. Refer to the programming notes for this register (on page 4-38) for conditions that set this bit.
7	Res	**Reserved** For compatibility with future devices, always write this bit field with its chip reset default value.
6	ISO_START	**Isochronous Packet Transfer Has Started** The ISO_START bit is set when the endpoint is configured for isochronous transfers and the data phase of the transfer has begun. This bit is automatically cleared when the USB SOF packet from the host is detected or when the internally synthesized SOF is generated.
5	ISO_STOP	**Isochronous Packet Transfer Has Ended** The ISO_STOP bit is set when the endpoint is configured for isochronous transfers and the data phase of the transfer has ended. This bit is automatically cleared when the USB SOF packet from the host is detected or when the internally synthesized SOF is generated.
4	ISO_MS	**Isochronous Packet Missed** The ISO_MS bit is set if the endpoint is configured for isochronous transfers and a token packet from the USB host is not detected in a USB frame. That is, if the ISO_STOP bit is not set when the USB SOF packet from the host is detected or when the internally synthesized SOF is generated, this bit is set. The endpoint controller hardware can be configured to stop when the ISO_MS bit is set if the endpoint is configured for the DMA mode of operation. (See the ISO_MS_SMSK bit on page 4-52.) If the ISO_MS bit is enabled to stop the endpoint hardware, this bit should be cleared by software prior to or at the same time as the ACT_REQ bit is cleared. In either DMA or Non-DMA mode, the endpoint controller hardware can be configured to generate an interrupt, via the STAT_INT bit, when the ISO_MS bit is set. (See the ISO_MS_IMSK bit on page 4-50.)

Bit	Name	Function
3	FULL_PKT	**Full Data Packet Received** The FULL_PKT bit is set when a full USB data packet is received without error (i.e., the received packet size is exactly equal to the endpoint’s maximum packet value). The endpoint controller hardware can be configured to stop when the FULL_PKT bit is set. (See the FULL_PKT_SMSK bit on page 4-52.) If the FULL_PKT bit is enabled to stop the endpoint hardware, this bit should be cleared by software prior to or at the same time as the ACT_REQ bit is cleared. The endpoint controller hardware can also be configured to generate an interrupt, via the STAT_INT bit, when the FULL_PKT bit is set. (See the FULL_PKT_IMSK bit on page 4-51.) This bit is not valid when the endpoint is configured for its non-DMA mode or the IN direction.
2	SHORT_PKT	**Short Data Packet Received** The SHORT_PKT bit is set when a short USB data packet is received without error (i.e., the received packet size is less than the endpoint’s maximum packet value). The endpoint controller hardware can be configured to stop when the SHORT_PKT bit is set. (See the SHRT_PKT_SMSK bit on page 4-52.) If the SHORT_PKT bit is enabled to stop the endpoint hardware, this bit should be cleared by software prior to or at the same time as the ACT_REQ bit is cleared. The endpoint controller hardware can also be configured to generate an interrupt, via the STAT_INT bit, when the SHORT_PKT bit is set. (See the SHRT_PKT_IMSK bit on page 4-51.) This bit is not valid when the endpoint is configured for its non-DMA mode or the IN direction.

Bit	Name	Function
1	BUF_ERR	**Buffer Error** The BUFF_ERR bit is set when a buffer error is detected during the reception or transmission of a USB data packet. When set, this bit indicates that a buffer overflow has occurred (if the endpoint is configured for the OUT direction), or that a buffer underflow has occurred (if the endpoint is configured for the IN direction). When this bit is set, the device performs the activity as listed below for the various direction and endpoint type combinations: ■ Bulk OUT = The device sends a NAK packet to the host in the handshake phase. ■ Isochronous OUT = For isochronous transfers, there is never a device-to-host handshake phase. ■ Bulk IN and isochronous IN = The device inserts the bit stuff error into the data during the device-to-host data phase. The endpoint controller hardware can be configured to stop when the BUFF_ERR bit is set. (See the BUFF_ERR_SMSK bit on page 4-52.) If the BUFF_ERR bit is enabled to stop the endpoint hardware, this bit should be cleared by software prior to or at the same time as the ACT_REQ bit is cleared. The endpoint controller hardware can also be configured to generate an interrupt, via the STAT_INT bit, when the BUFF_ERR bit is set. (See the BUFF_ERR_IMSK bit on page 4-51.) This bit is not valid when the endpoint is configured for its non-DMA mode.
0	OTHER_ERR	**Other Error** The OTHER_ERR bit is set when a USB error is detected during the reception or transmission of a USB data packet and the endpoint is operating in DMA or non-DMA mode. When the endpoint is configured for the OUT direction, its type is bulk or isochronous, and the OTHER_ERR bit is set, this indicates that the endpoint has detected either a packet ID (PID) error, CRC error, bit stuff error, data toggle error (bulk only), or more than the maximum packet value of data bytes sent by the USB host. For a data toggle error, an ACK packet is returned to the USB host during the handshake phase. If the endpoint type is isochronous, there is never a handshake phase. When the endpoint is configured for the IN direction, its type is bulk, and the OTHER_ERR bit is set, this indicates that the USB host has detected either a CRC error, bit stuff error, or a data toggle error, and that the USB host has not returned a handshake packet. This bit is never set when the endpoint is configured for the IN direction and its type is isochronous. When the endpoint is configured for DMA mode, the endpoint controller hardware can be configured to stop when the OTHER_ERR bit is set. (See the OTH_ERR_SMSK bit on page 4-52.) If the OTHER_ERR bit is enabled to stop the endpoint hardware, this bit should be cleared by software prior to or at the same time as the ACT_REQ bit is cleared. The endpoint controller hardware can also be configured to generate an interrupt, via the STAT_INT bit, when the OTHER_ERR bit is set. (See the OTH_ERR_IMSK bit on page 4-51.) When the endpoint is configured for the OUT direction, non-DMA mode, and its type is isochronous, this bit is set upon the USB hardware detecting one of the following errors during the data transmission from the USB host: PID error, CRC error, bit stuff error, or more than the maximum packet value of data bytes sent by the USB host. The device software should test this bit upon the ACT_REQ bit being set to test for these error conditions while operating in this mode (OUT direction, non-DMA mode, isochronous transfer).

Programming Notes

If the EP_EN bit is set and a USB reset or USB suspend condition is detected, or if the endpoint is configured for its DMA mode of operation and the NOT_FLUSH bit is cleared:

- EP_NOT_STALLED = 1
- NOT_FLUSH = 1 (always reads same value as the EP_EN bit)
- ACT_REQ = 0 (OUT) or ACT_REQ = 1 (IN)
- NOT_ZERO and NOT_LAST_BYTE = 1
- The remaining bits go to their default state

If the EP_EN bit is set; the endpoint is configured as bulk or interrupt; and software clears the EP_NOT_STALLED bit:

- EP_NOT_STALLED = 0 (was just cleared by software)
- NOT_FLUSH = 1 (always reads same value as the EP_EN bit)
- ACT_REQ = 0 (OUT) or ACT_REQ = 1 (IN)
- NOT_ZERO and NOT_LAST_BYTE = 1
- The remaining bits go to their default state

All the bits in this register are cleared if the EP_EN bit is cleared.

All the bits in this register are cleared upon an external or watchdog timer reset. When software wants to enable this endpoint, it typically writes a value of FFFFh to this register. During this write, with the EP_EN bit transitioning from Low to High, hardware forces the NOT_FLUSH, NOT_ZERO, and NOT_LAST_BYTE bits to be set.

When the endpoint is configured for the Non-DMA mode, the ACT_REQ bit is set or cleared depending on the state of the EP_DIR bit located in the Endpoint Definition 1 register for this endpoint (A, B, C, or D). In non-DMA mode, ACT_REQ is set if the EP_DIR bit is 0 and is cleared if the EP_DIR bit is 1. The ACT_REQ bit is cleared when the endpoint is enabled if the endpoint is configured for its DMA mode. All other bits are in their cleared state.

As a special case, software might want to enable the endpoint in its stalled condition (i.e., with the EP_NOT_STALLED bit cleared). This can be accomplished if, during the initial write operation to this register, with the EP_EN bit transitioning from a value of 0 to a value of 1, the EP_NOT_STALLED bit position is written as a value of 0b. This allows software to cause the endpoint to be in the stalled state when it is initially enabled. Note that it is the reception of a token packet that causes a stall handshake to be sent to the host if this bit is cleared by software.

The EP_NOT_STALLED bit cannot be cleared by software at the same time as EP_EN is being set if EP_EN is already set. For this endpoint, the EP_NOT_STALLED bit is automatically set when the USB host issues a CLEAR_FEATURE command that specifies the stall option and this endpoint’s number.

A Endpoint Received Packet Size (AEPSIZ) Offset 222h
B Endpoint Received Packet Size (BEPSIZ) Offset 232h
C Endpoint Received Packet Size (CEPSIZ) Offset 242h
D Endpoint Received Packet Size (DEPSIZ) Offset 252h

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	*Res*						RPS									
Software Read/Write	R						R									
Hardware Set/Clear	—						S/C									
Chip Reset Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Microcontroller:

Am186CC	☑
Am186CH	□
Am186CU	☑

Register Description

This register provides status information about the size of the received packet (in bytes) when the endpoint is configured for the OUT direction.

Some bit field descriptions depend on the endpoint's mode of operation (DMA or non-DMA), as set in the Endpoint Definition 3 register (see page 4-50).

Bit Definitions

Bit	Name	Function
15–10	Res	**Reserved** For compatibility with future devices, always write this bit field with its chip reset default value.
9–0	RPS	**Received Packet Size** The RPS bit field contains the receive packet size value. The RPS bit field provides status regarding the received packet size (in bytes) to the endpoint handling software. When a valid packet has been received from the USB host and resides in the data buffer, software can use this bit field to determine the amount of valid data in the buffer prior to performing the read operations. When the endpoint is configured for non-DMA mode, the RPS bit field contains the valid size, in bytes, of the received data packet when the ACT_REQ bit is set. When the endpoint is configured for DMA mode and the SHORT_PKT bit in the endpoint's Control/Status register is set, the device software can use this bit field to determine the size of the short packet that was received.

Programming Notes

All the bits in this register are cleared if the EP_EN bit is set and a USB reset condition is detected, a USB suspend condition is detected, or the endpoint's buffer is flushed by software.

If the EP_EN bit is set and the endpoint's EP_NOT_STALLED bit is cleared, all the bits in this register are cleared only if the endpoint is configured to be bulk. That is, the EP_NOT_STALLED bit has no effect when the endpoint is configured to be isochronous.

All the bits in this register are cleared if the EP_EN bit is cleared.

All the bits in this register are cleared upon an external or watchdog timer reset.

A Endpoint Buffer Status (AEPBUFS) Offset 224h
B Endpoint Buffer Status (BEPBUFS) Offset 234h

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	*Res*											BUF_STAT				
Software Read/Write	R											R				
Hardware Set/Clear	—											S/C				
Chip Reset Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Microcontroller:
Am186CC ☑
Am186CH □
Am186CU ☑

Register Description

This register provides status information about the present condition of the endpoint's buffer.

The Endpoint Buffer Status registers for endpoints C and D are similar, except BUF_STAT is seven bits wide in those registers (see page 4-41).

Bit Definitions

Bit	Name	Function
15–5	Res	**Reserved** For compatibility with future devices, always write this bit field with its chip reset default value.
4–0	BUF_STAT	**Buffer Status** The device software can use the BUF_STAT bit field to determine how much data, if any, is in the buffer. This is useful for bulk IN transactions when the endpoint is using the DMA mode. If more than one USB packet's data is in the buffer at the same time and an error is detected, the DMA must stop and software must "back up" its memory pointer so that the beginning of the failed packet can be re-sent to the buffer. This bit field allows the device software to determine how far to back up its pointers. This bit field is valid when the endpoint is configured for any transfer type, in any direction, and in any mode.

Programming Notes

All the bits in this register are cleared if the EP_EN bit is set and a USB reset condition is detected, a USB suspend condition is detected, or the endpoint's buffer is flushed by software.

If the EP_EN bit is set and the endpoint's EP_NOT_STALLED bit is cleared, all the bits in this register are cleared only if the endpoint is configured to be bulk or interrupt. That is, the EP_NOT_STALLED bit has no effect when the endpoint is configured to be isochronous.

All the bits in this register are cleared if the EP_EN bit is cleared.

All the bits in this register are cleared upon an external or watchdog timer reset.

C Endpoint Buffer Status (CEPBUFS) Offset 244h

D Endpoint Buffer Status (DEPBUFS) Offset 254h

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	*Res*									BUF_STAT						
Software Read/Write	R									R						
Hardware Set/Clear	—									S/C						
Chip Reset Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Microcontroller:	
Am186CC	✔
Am186CH	
Am186CU	✔

Register Description

This register provides status information about the present condition of the endpoint's buffer.

These registers are similar to the Endpoint Buffer Status register for endpoints A and B except BUF_STAT is only five bits wide in those registers (see page 4-40).

Bit Definitions

Bit	Name	Function
15–7	Res	**Reserved** For compatibility with future devices, always write this bit field with its chip reset default value.
6–0	BUF_STAT	**Buffer Status (Same as AEPBUFS)** The device software can use the BUF_STAT bit field to determine how much data, if any, is in the buffer. This is useful for bulk IN transactions when the endpoint is using the DMA mode. If more than one USB packet's data is in the buffer at the same time and an error is detected, the DMA must stop and software must "back up" its memory pointer so that the beginning of the failed packet can be re-sent to the buffer. This bit field allows the device software to determine how far to back up its pointers. This bit field is valid when the endpoint is configured for any transfer type, in any direction, and in any mode.

Programming Notes

All the bits in this register are cleared if the EP_EN bit is set and a USB reset condition is detected, a USB suspend condition is detected, or the endpoint's buffer is flushed by software.

If the EP_EN bit is set and the endpoint's EP_NOT_STALLED bit is cleared, all the bits in this register are cleared only if the endpoint is configured to be bulk or interrupt. That is, the EP_NOT_STALLED bit has no effect when the endpoint is configured to be isochronous.

All the bits in this register are cleared if the EP_EN bit is cleared.

All the bits in this register are cleared upon an external or watchdog timer reset.

A Endpoint Data Port (AEPDAT) Offset 226h
B Endpoint Data Port (BEPDAT) Offset 236h
C Endpoint Data Port (CEPDAT) Offset 246h
D Endpoint Data Port (DEPDAT) Offset 256h

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	*Res*								D							
Software Read/Write	R								R/W							
Hardware Set/Clear	—								S/C							
Chip Reset Default	0	0	0	0	0	0	0	0	x	x	x	x	x	x	x	x

Microcontroller:	
Am186CC	✔
Am186CH	□
Am186CU	✔

Register Description

The device software or DMA controller uses the xEPDAT register to read or write the endpoint data buffer when the endpoint is configured to operate in non-DMA mode or general-purpose DMA mode.

Some bit field descriptions depend on the endpoint's mode of operation (DMA or non-DMA), as set in the endpoint's Endpoint Definition 3 register (see page 4-50).

Bit Definitions

Bit	Name	Function
15–8	Res	**Reserved** For compatibility with future devices, always write this bit field with its chip reset default value.
7–0	D	**Endpoint Data Port** ***Non-DMA mode*** The D bit field is used by the endpoint handling software to read and write data to be received by and transmitted from this endpoint. Access to this data port is controlled by the endpoint hardware in conjunction with the device software's use of the endpoint's status bits located in the xEPCTL register. See the xEPCTL (x = A, B, C, or D) register description starting on page 4-33 for details about the device software access limitations to the data port. All device software accesses to this data port are ignored unless the EP_EN and ACT_REQ bits are set. ***DMA mode*** The D bit field is used by the DMA controller to read and write data to be received by and transmitted from this endpoint.

Programming Notes

A Endpoint Receive Data Port Peek (ARCVPK) Offset 228h
B Endpoint Receive Data Port Peek (BRCVPK) Offset 238h
C Endpoint Receive Data Port Peek (CRCVPK) Offset 248h
D Endpoint Receive Data Port Peek (DRCVPK) Offset 258h

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	*Res*								D							
Software Read/Write	R								R							
Hardware Set/Clear	—								S/C							
Chip Reset Default	0	0	0	0	0	0	0	0	x	x	x	x	x	x	x	x

Microcontroller:
Am186CC ☑
Am186CH □
Am186CU ☑

Register Description

Debug or emulator software uses the xRCVPK register to read the endpoint data buffer without advancing the endpoint's data buffer address pointer when the endpoint is configured for the OUT direction.

Some bit field descriptions depend on the endpoint's mode of operation (DMA or non-DMA), as set in the endpoint's Endpoint Definition 3 register (see page 4-50).

Bit Definitions

Bit	Name	Function
15–8	Res	**Reserved** For compatibility with future devices, always write this bit field with its chip reset default value.
7–0	D	**Endpoint Receive Data Peek** ***Non-DMA mode/OUT direction*** The D bit field can be used by debug or emulator software to read data from the endpoint without advancing the endpoint's data buffer address pointer when the ACT_REQ bit is set. Access to this port is controlled by the same mechanisms that are used to control the endpoint's data port register. See the xEPCTL register description starting on page 4-33 for details about the device software access limitations to the data port.

Programming Notes

A Endpoint Definition 1 (AEPDEF1) Offset 22Ah
B Endpoint Definition 1 (CEPDEF1) Offset 23Ah
C Endpoint Definition 1 (CEPDEF1) Offset 24Ah
D Endpoint Definition 1 (DEPDEF1) Offset 25Ah

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	EP_NUM				EP_CFG		*Res*	EP_INT		*Res*	EP_ASET			EP_DIR	EP_TYPE	
Software Read/Write	R/W				R/W		R	R/W		R	R/W			R/W	R/W	
Hardware Set/Clear	—				—		—	—		—	—			—	—	
Chip Reset Default	See the bit description.				0	0	0	0	0	0	0	0	0	1	1	0

Microcontroller:
Am186CC ☑
Am186CH □
Am186CU ☑

Register Description

The device software uses the xEPDEF1 register to define endpoints A–D.

Note: *Correct endpoint operation is **not guaranteed** if this register is modified when the endpoint is enabled (i.e., while the endpoint's EP_EN bit is set, see page 4-33).*

Bit Definitions

Bit	Name	Function
15–12	EP_NUM	**Endpoint Number** The EP_NUM bit field is used to set the USB endpoint number for the endpoint. The endpoint number for this endpoint should not be programmed to a value of 0d if it is enabled. Additionally, no two enabled endpoints within the device should share the same number. The chip reset default value for this bit field is as follows: 0010 = Endpoint A 0011 = Endpoint B 0100 = Endpoint C 0101 = Endpoint D If an enabled endpoint is disabled by clearing the EP_EN bit in the xEPCTL register, software must also clear the EP_NUM bit for that endpoint.
11–10	EP_CFG	**Endpoint Configuration** The EP_CFG bit field is used by software to relate this endpoint to a configuration that the USB host selected by issuing a SET_CONFIGURATION command. Configurations 0d–3d are supported.
9	Res	**Reserved** For compatibility with future devices, always write this bit field with its chip reset default value.
8–7	EP_INT	**Endpoint Interface** The EP_INT bit field is used to relate this endpoint to a host-selected interface within the selected configuration. Interfaces 0d–3d are supported.
6	Res	**Reserved** For compatibility with future devices, always write this bit field with its chip reset default value.
5–3	EP_ASET	**Alternate Setting** The EP_ASET bit field is used to relate this endpoint to a host-selected interface alternate setting. Each interface supports a total of eight alternate settings (0d–7d).

Bit	Name	Function
2	EP_DIR	**Endpoint Direction (IN or OUT)** The EP_DIR bit is used to select the direction for this endpoint. 0 = IN 1 = OUT These endpoints can be programmed for the IN or OUT direction regardless of the endpoint type.
1–0	EP_TYPE	**Endpoint Type Selection** The EP_TYPE bit field is used to select the type of USB transfers that this endpoint supports. 00 =Control (not valid for this endpoint) 01 =Isochronous 10 =Bulk 11 =Interrupt For this endpoint, 00b is an illegal value.

Programming Notes

Host software should only set configuration and interface values that match a device descriptor returned by the device in response to a GET_DESCRIPTOR command. However, the USB hardware accepts as valid any configuration or feature setting in the range of 0d–3d, regardless of the available descriptors. To help ensure reliable operation in any USB environment, device software can define a minimal descriptor (i.e., Endpoint 0 with no bandwidth allocation) for any configuration and interface settings that it does not define otherwise.

A Endpoint Definition 2 (AEPDEF2) Offset 22Ch

B Endpoint Definition 2 (BEPDEF2) Offset 23Ch

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	*Res*					FIFO_SIZE	EP_MX_PCT									
Software Read/Write	R					R/W	R/W									
Hardware Set/Clear	—					—	—									
Chip Reset Default	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0

Microcontroller:	
Am186CC	✔
Am186CH	
Am186CU	✔

Register Description

The device software uses the xEPDEF2 register to set the endpoint's maximum packet size and effective FIFO size when configured for the IN direction.

The Endpoint Definition 2 registers for endpoints C and D are similar, except FIFO_SIZE is two bits wide and defaults to 11b in those registers (see page 4-48).

Note: *Correct endpoint operation is* ***not guaranteed*** *if this register is modified when the endpoint is enabled (i.e., while the endpoint's EP_EN bit is set, see page 4-33).*

Bit Definitions

Bit	Name	Function
15–11	Res	**Reserved** For compatibility with future devices, always write this bit field with its chip reset default value.
10	FIFO_SIZE	**FIFO Size** The FIFO_SIZE bit field is valid for DMA mode only. It is used to select the endpoint buffer depth when the endpoint is configured for the IN direction. This bit field has no effect when the endpoint is configured for the OUT direction except that its default state can serve to inform the device software about the physical size of the buffer. The following FIFO sizes can be selected for endpoints A and B: 0 = 8 bytes deep 1 = 16 bytes deep The chip reset default value for this bit field (1b) indicates the physical buffer size (16d). Endpoints C and D have a 2-bit FIFO_SIZE bit field and 64-byte physical buffers (see page 4-48). When the endpoint is configured for the IN direction, data that is requested by the USB host is not sent to the host (i.e., IN transactions receive a NAK response) unless the buffer is full or there is status such as last byte or zero packet in the buffer. To minimize NAKs when configured for the IN direction, software can adjust the endpoint's effective buffer size to optimize the endpoint for minimal delay/skew between the data source and the USB host.
9–0	EP_MX_PCT	**Endpoint Max Packet Value** The EP_MX_PCT bit field is used to select the maximum packet value for the endpoint. This bit field can be programmed with a value of 0d to 1023d. Software must follow the USB rules for the endpoint types when programming this bit field. In addition, it is illegal to program this bit field to a value of 0d and then enable the endpoint. For example, this value can only be 8d, 16d, 32d, or 64d if the endpoint is configured to be bulk. This value must be less than or equal to 64d (but not 0d) if the endpoint is configured to be interrupt. There are no logical restrictions if the endpoint is configured to be isochronous except that the value cannot be 0d. However, if a 24-MHz processor clock is used and EP_MX_PCT must be 1023 bytes, then it is necessary to use endpoint C or D and set the FIFO size to 64 bytes. At 24 MHz, software cannot handle a 1023-byte packet before a smaller FIFO is filled or emptied. In addition, if the endpoint is configured to be used in the non-DMA mode, the maximum packet value configured for the endpoint is required to be less than or equal to the physical size of the endpoint's data buffer (i.e., 16d for endpoints A and B, 64d for endpoints C and D).

Programming Notes

C Endpoint Definition 2 (CEPDEF2) Offset 24Ch
D Endpoint Definition 2 (DEPDEF2) Offset 25Ch

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	*Res*				FIFO_SIZE		EP_MX_PCT									
Software Read/Write	R				R/W		R/W									
Hardware Set/Clear	—				—		—									
Chip Reset Default	0	0	0	0	1	1	0	0	0	0	0	0	1	0	0	0

Microcontroller:

Am186CC ☑
Am186CH □
Am186CU ☑

Register Description

The device software uses the xEPDEF2 register to set the endpoint's maximum packet size and effective FIFO size when configured for the IN direction.

This register is similar to Endpoint Definition 2 for endpoints A and B (see page 4-46), except for the width and default value of the FIFO_SIZE bit field.

Note: *Correct endpoint operation is* ***not guaranteed*** *if this register is modified when the endpoint is enabled (i.e., while the endpoint's EP_EN bit is set, see page 4-33).*

Bit Definitions

Bit **Name** **Function**

15–12 Res **Reserved**

For compatibility with future devices, always write this bit field with its chip reset default value.

11–10 FIFO_SIZE **FIFO Size**

The FIFO_SIZE bit field is valid for DMA mode only. It is used to select the endpoint buffer depth when the endpoint is configured for the IN direction. This bit field has no effect when the endpoint is configured for the OUT direction except that its default state can serve to inform the device software about the physical size of the buffer.

The following FIFO sizes can be selected for endpoints C and D:

00 = 8 bytes deep
01 = 16 bytes deep
10 = 32 bytes deep
11 = 64 bytes deep

The chip reset default value for this bit field (11b) indicates the physical buffer size (64d).

Endpoints A and B have a 1-bit FIFO_SIZE bit field and 16-byte physical buffers (see page 4-46).

When the endpoint is configured for the IN direction, data that is requested by the USB host is not sent to the host (i.e., IN transactions receive a NAK response) unless the buffer is full or there is status such as last byte or zero packet in the buffer. To minimize NAKs when configured for the IN direction, software can adjust the endpoint's effective buffer size to optimize the endpoint for minimal delay/skew between the data source and the USB host.

9–0 EP_MX_PCT **Endpoint Max Packet Value (Same as AEPDEF2)**

The EP_MX_PCT bit field is used to select the maximum packet value for the endpoint.

This bit field can be programmed with a value of 0d to 1023d. Software must follow the USB rules for the endpoint types when programming this bit field. In addition, it is illegal to program this bit field to a value of 0d and then enable the endpoint.

For example, this value can only be 8d, 16d, 32d, or 64d if the endpoint is configured to be bulk.

This value must be less than or equal to 64d (but not 0d) if the endpoint is configured to be interrupt.

There are no logical restrictions if the endpoint is configured to be isochronous except that the value cannot be 0d. However, if a 24-MHz processor clock is used and EP_MX_PCT must be 1023 bytes, then it is necessary to use endpoint C or D and set the FIFO size to 64 bytes. At 24 MHz, software cannot handle a 1023-byte packet before a smaller FIFO is filled or emptied.

In addition, if the endpoint is configured to be used in the non-DMA mode, the maximum packet value configured for the endpoint is required to be less than or equal to the physical size of the endpoint's data buffer (i.e., 16d for endpoints A and B, 64d for endpoints C and D).

Programming Notes

A Endpoint Definition 3 (AEPDEF3) Offset 22Eh
B Endpoint Definition 3 (BEPDEF3) Offset 23Eh
C Endpoint Definition 3 (CEPDEF3) Offset 24Eh
D Endpoint Definition 3 (DEPDEF3) Offset 25Eh

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	*Res*		AUTO_RATE_EN	ISO_MS_IMSK	FULL_PKT_IMSK	SHRT_PKT_IMSK	BUF_ERR_IMSK	OTH_ERR_IMSK	MODE			ISO_MS_SMSK	FULL_PKT_SMSK	SHRT_PKT_SMSK	BUF_ERR_SMSK	OTH_ERR_SMSK
Software Read/Write	R		R/W	R/W					R/W			R/W				
Hardware Set/Clear	—		—	—					—			—				
Chip Reset Default	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0

Microcontroller:
Am186CC ✔
Am186CH □
Am186CU ✔

Register Description

The device software uses the xEPDEF3 register to select various operating modes for endpoints A, B, C, and D.

Note: *Correct endpoint operation is **not guaranteed** if this register is modified when the endpoint is enabled (i.e., while the endpoint's EP_EN bit is set, see page 4-33).*

Bit Definitions

Bit	Name	Function
15–14	Res	**Reserved** For compatibility with future devices, always write this bit field with its chip reset default value.
13	AUTO_RATE_EN	**Auto Rate Control Enable** The AUTO_RATE_EN bit field is used to enable and configure the auto rate feature of an endpoint that is configured for the IN direction, is of type isochronous, and has been configured for one of the DMA modes of operation. This bit field is encoded as follows: 0 = Auto rate disabled 1 = Auto rate enabled The auto rate feature allows an isochronous IN endpoint to have the number of bytes that it allows the host to read, per USB frame, controlled from a non-USB data sample clock. This clock can be selected to be the Channel A PCM highway/GCI frame synchronization, or an external device driving the USB sample clock input (USBSCI) signal on the UCLK pin. In addition, the Am186CC microcontroller provides a multiplier of 1, 2, or 4 so that data streams consisting of 1, 2, or 4 bytes per sample are supported when using this feature. Refer to the ISCTL register description (page 4-15) for additional details. Note that the configured maximum packet size for the endpoint must be greater than or equal to the largest number of data bytes that might be moved by the endpoint during a USB transaction.
12	ISO_MS_IMSK	**Isochronous Missed Packet Interrupt Mask** 0 = The ISO_MS status bit in the xEPCTL register (see page 4-35) does not affect the STAT_INT bit. 1 = Enables the ISO_MS status bit, when set, to set the endpoint's STAT_INT bit. Software can enable an endpoint's STAT_INT bit to generate an interrupt by setting the appropriate x_EP_STATINT bit in the UIMASK1 register (see page 4-5).

Bit	Name	Function
11	FULL_PKT_IMSK	**Full Data Packet Interrupt Mask** 0 = The FULL_PKT status bit in the xEPCTL register (see page 4-36) does not affect the STAT_INT bit. 1 = Enables the FULL_PKT status bit, when set, to set the endpoint's STAT_INT bit. Software can enable an endpoint's STAT_INT bit to generate an interrupt by setting the appropriate x_EP_STATINT bit in the UIMASK1 register (see page 4-5).
10	SHRT_PKT_IMSK	**Short Data Packet Interrupt Mask** 0 = The SHORT_PKT status bit in the xEPCTL register (see page 4-36) does not affect the STAT_INT bit. 1 = Enables the SHORT_PKT status bit, when set, to set the endpoint's STAT_INT bit. Software can enable an endpoint's STAT_INT bit to generate an interrupt by setting the appropriate x_EP_STATINT bit in the UIMASK1 register (see page 4-5).
9	BUFF_ERR_IMSK	**Buffer Error Interrupt Mask** 0 = The BUFF_ERR status bit in the xEPCTL register (see page 4-37) does not affect the STAT_INT bit. 1 = Enables the BUFF_ERR status bit, when set, to set the endpoint's STAT_INT bit. Software can enable an endpoint's STAT_INT bit to generate an interrupt by setting the appropriate x_EP_STATINT bit in the UIMASK1 register (see page 4-5).
8	OTH_ERR_IMSK	**Other Error Interrupt Mask** 0 = The OTHER_ERR status bit in the xEPCTL register (see page 4-37) does not affect the STAT_INT bit. 1 = Enables the OTHER_ERR status bit, when set, to set the endpoint's STAT_INT bit. Software can enable an endpoint's STAT_INT bit to generate an interrupt by setting the appropriate x_EP_STATINT bit in the UIMASK1 register (see page 4-5).
7–5	MODE	**Transfer Mode** The MODE bit field is used by the device software to configure the endpoint for a particular mode of operation relative to the way in which data is moved between the endpoint's buffer and system memory or another peripheral's data ports. This bit field is encoded as follows: 000 = Non-DMA mode 001 = Reserved 010 = The general-purpose DMA controller is used; the terminal count is ignored for the IN direction (i.e., the DMA terminal count does not mark the byte written into the endpoint buffer when terminal count is asserted as the packet's last byte). 011 = The general-purpose DMA controller is used; the terminal count is not ignored for the IN direction (i.e., the DMA terminal count does mark the byte written into the endpoint buffer when the terminal count is asserted as the packet's last byte). 100 = The SmartDMA controller is used; the packet status is not stored to the buffer descriptor if configured for the OUT direction. 101 = The SmartDMA controller is used; the packet status is stored in the buffer descriptor if configured for the OUT direction. 110 = Reserved 111 = Reserved When the endpoint is configured for the OUT direction, modes 010 or 011 result in exactly the same operation. When the endpoint is configured for the IN direction, modes 100 or 101 result in exactly the same operation. If the ACT_REQ bit is set for any reason while the endpoint is in one of the DMA modes, the endpoint hardware is forced to stop and no packet status is automatically loaded into the buffer descriptor for the SmartDMA controller mode. If the endpoint is configured as interrupt, only mode 000 is valid.

Bit	Name	Function
4	ISO_MS_SMSK	**Isochronous Missed Packet Stop Mask** 0 = The ISO_MS status bit in the xEPCTL register (see page 4-35), when set, does not stop the endpoint hardware or set the ACT_REQ bit. 1 = Enables the ISO_MS status bit, when set, to stop the endpoint hardware and set the endpoint's ACT_REQ bit. Software can enable an endpoint's ACT_REQ bit to generate an interrupt by setting the appropriate x_EP_ACT bit in the UIMASK1 register (see page 4-5).
3	FULL_PKT_SMSK	**Full Data Packet Stop Mask** 0 = The FULL_PKT status bit in the xEPCTL register (see page 4-36), when set, does not stop the endpoint hardware or set the ACT_REQ bit. 1 = Enables the FULL_PKT status bit, when set, to stop the endpoint hardware and set the endpoint's ACT_REQ bit. Software can enable an endpoint's ACT_REQ bit to generate an interrupt by setting the appropriate x_EP_ACT bit in the UIMASK1 register (see page 4-5).
2	SHRT_PKT_SMSK	**Short Data Packet Stop Mask** 0 = The SHORT_PKT status bit in the xEPCTL register (see page 4-36), when set, does not stop the endpoint hardware or set the ACT_REQ bit. 1 = Enables the SHORT_PKT status bit, when set, to stop the endpoint hardware and set the endpoint's ACT_REQ bit. Software can enable an endpoint's ACT_REQ bit to generate an interrupt by setting the appropriate x_EP_ACT bit in the UIMASK1 register (see page 4-5).
1	BUFF_ERR_SMSK	**Buffer Error Stop Mask** 0 = The BUFF_ERR status bit in the xEPCTL register (see page 4-37), when set, does not stop the endpoint hardware or set the ACT_REQ bit. 1 = Enables the BUFF_ERR status bit, when set, to stop the endpoint hardware and set the endpoint's ACT_REQ bit. Software can enable an endpoint's ACT_REQ bit to generate an interrupt by setting the appropriate x_EP_ACT bit in the UIMASK1 register (see page 4-5).
0	OTH_ERR_SMSK	**Other Error Stop Mask** 0 = The OTHER_ERR status bit in the xEPCTL register (see page 4-37), when set, does not stop the endpoint hardware or set the ACT_REQ bit. 1 = Enables the OTHER_ERR status bit, when set, to stop the endpoint hardware and set the endpoint's ACT_REQ bit. Software can enable an endpoint's ACT_REQ bit to generate an interrupt by setting the appropriate x_EP_ACT bit in the UIMASK1 register (see page 4-5).

Programming Notes

When the auto rate control feature is used (applicable for isochronous IN transfers that use DMA), the value of the endpoint maximum packet size (EP_MX_PCT bit in the xEPDEF2 register) should be equal to or greater than the expected maximum data transfer value for the frame (based on the sample-clock-source and bytes-per-sample values programmed in the ISCTL register, see page 4-15). This ensures that the auto rate feature's variable-size data transfers are not restricted by the maximum packet size.

When the MODE bit field is set to 101b (SmartDMA channel, status stored in the buffer descriptor), a bulk OUT transfer that results in a retransmission of data by the host due to handshake packet errors produces the following buffer descriptor field values: STP = 1, ENP = 1, and CRC = 1. The MCNT value in the buffer descriptor is invalid because setting the CRC bit causes the ERR bit to be set as well.

Also, when the MODE bit field is set to 101b, a bulk or isochronous OUT transfer with a message size that is an integer multiple of the maximum packet size results in the following buffer descriptor field values: STP = 1, ENP = 1, and MCNT = 0.

CHAPTER

5 ASYNCHRONOUS SERIAL PORT (UART) REGISTERS

5.1 OVERVIEW

This chapter describes the asynchronous serial port registers on the Am186CC/CH/CU microcontrollers.

These controllers provide two independent asynchronous serial ports, the Universal Asynchronous Receiver/Transmitter (UART) and the High-Speed UART.

The UART provides standard serial communications features such as full-duplex operation, 7-, 8-, or 9-bit data transfers, parity, hardware flow control, break handling, maskable interrupt generation, and DMA.

The High-Speed UART has all the UART features, plus 32-byte receive and 16-byte transmit first-in-first-out buffers (FIFOs), automatic baud rate detection, and receive character matching with maskable interrupt generation.

For more information about using the asynchronous serial ports, see the *Am186CC/CH/CU Microcontrollers User's Manual*, order #21914.

This chapter describes general High-Speed UART registers first, then the UART registers. Table 5-1 lists the UART registers in offset order, with the corresponding description's page number.

Table 5-1 Asynchronous Serial Port Register Map

Register Name	Mnemonic	Offset	Page Number
High-Speed Asynchronous Serial Port Registers			
High-Speed Serial Port Control 0	HSPCON0	0260h	page 5-3
High-Speed Serial Port Control 1	HSPCON1	0262h	page 5-5
High-Speed Serial Port Status	HSPSTAT	0264h	page 5-7
High-Speed Serial Port Interrupt Mask	HSPIMSK	0266h	page 5-10
High-Speed Serial Port Transmit Data	HSPTXD	0268h	page 5-13
High-Speed Serial Port Receive Data	HSPRXD	026Ah	page 5-14
High-Speed Serial Port Receive Data Peek	HSPRXDP	026Ch	page 5-16
High-Speed Serial Port Baud Rate Divisor	HSPBDV	026Eh	page 5-18
High-Speed Serial Port Character Match 0	HSPM0	0270h	page 5-20
High-Speed Serial Port Character Match 1	HSPM1	0272h	page 5-21
High-Speed Serial Port Character Match 2	HSPM2	0274h	page 5-22
High-Speed Serial Port Autobaud 0	HSPAB0	0276h	page 5-23
High-Speed Serial Port Autobaud 1	HSPAB1	0278h	page 5-25
High-Speed Serial Port Autobaud 2	HSPAB2	027Ah	page 5-26
High-Speed Serial Port Autobaud 3	HSPAB3	027Ch	page 5-27

Table 5-1 Asynchronous Serial Port Register Map (Continued)

Register Name	Mnemonic	Offset	Page Number
Asynchronous Serial Port Registers			
Serial Port Control 0	SPCON0	0280h	page 5-28
Serial Port Control 1	SPCON1	0282h	page 5-30
Serial Port Status	SPSTAT	0284h	page 5-31
Serial Port Interrupt Mask	SPIMSK	0286h	page 5-33
Serial Port Transmit Data	SPTXD	0288h	page 5-35
Serial Port Receive Data	SPRXD	028Ah	page 5-36
Serial Port Receive Data Peek	SPRXDP	028Ch	page 5-38
Serial Port Baud Rate Divisor	SPBDV	028Eh	page 5-40

High-Speed Serial Port Control 0 (HSPCON0) Offset 260h

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	*Res*			RSIE	BRK	AB	FC	TXIE	RXIE	TMODE	RMODE	EVN	PEN	ABEN	D7	STP2
Software Read/Write	R			R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Hardware Set/Clear	—			—	—	C	—	—	—	—	—	—	—	—	—	—
Chip Reset Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Microcontroller:

Am186CC ✔
Am186CH ✔
Am186CU ✔

Register Description

The HSPCON0 register is the first of two control registers for the High-Speed serial port. The HSPCON1 register is the second control register (see page 5-5).

Bit Definitions

Bit	Name	Function
15–13	Res	**Reserved** For compatibility with future devices, always write this bit field with its chip reset default value.
12	RSIE	**Receive Status Interrupt Enable** 0 = Receive status interrupts are disabled. 1 = Receive status interrupts are enabled. The bits in the HSPSTAT register that generate receive status interrupts are MATCH, OERIM, IDLE, PER, OER, FER, AB, and BRK (see the descriptions starting on page 5-8).
11	BRK	**Send Break** 0 = The HSP_TXD line follows the output data. 1 = The HSP_TXD line is held to the value determined by the BRKVAL bit in the HSPCON1 register (see page 5-6).
10	AB	**Address Bit** 0 = The address bit for the next frame is driven Low if address generation is enabled by setting the ABEN bit to 1. 1 = The address bit for the next frame is driven High if address generation is enabled. The AB bit clears automatically after the value is read into either the transmit FIFO or the transmit shift register. The AB bit in this register has no effect if the EXDWR bit in the HSPCON1 register is set (see page 5-6). If the EXDWR bit is set, the transmitted address bit is controlled by the AB bit in the HSPTXD register (see page 5-13).
9	FC	**Flow Control Enable** 0 = Hardware flow control (CTS/RTR) is disabled. 1 = Hardware flow control is enabled. The FC bit should not be modified when either the TMODE bit or the RMODE bit is 1.
8	TXIE	**Transmit Data Interrupt Enable** 0 = Transmit interrupts are disabled. 1 = Transmit interrupts are enabled. The bits in the HSPSTAT register that generate transmit interrupts are TTHRSH, THRE, and TEMT (see the descriptions starting on page 5-7).
7	RXIE	**Receive Data Interrupt Enable** 0 = Receive data interrupts are disabled. 1 = Receive data interrupts are enabled. The bits in the HSPSTAT register that generate receive interrupts are RTHRSH, IDLED, and RDR (see the descriptions starting on page 5-7).

Bit	Name	Function
6	TMODE	**Transmitter Enable** 0 = The transmitter portion of the serial port is disabled. 1 = The transmitter portion of the serial port is enabled. Software must configure the HSPBDV, HSPCON1, and HSPIMSK registers appropriately before setting the TMODE bit. Transmitter bits in this register (HSPCON0) must be configured either before the TMODE bit is set or at the same time the bit is set.
5	RMODE	**Receiver Enable** 0 = The receiver portion of the serial port is disabled. 1 = The receiver portion of the serial port is enabled. Software must configure the HSPBDV, HSPCON1, and HSPIMSK registers appropriately before setting the RMODE bit. Receiver bits in this register (HSPCON0) must be configured either before the RMODE bit is set or at the same time the bit is set.
4	EVN	**Even Parity** 0 = When this bit is 0 and the PEN bit is 1, odd parity is in effect for the serial port. 1 = When this bit is 1 and the PEN bit is 1, even parity is in effect for the serial port. The EVN bit should not be modified when either the TMODE bit or the RMODE bit is 1. The EVN bit has no effect when the PEN bit is 0.
3	PEN	**Parity Enable** 0 = Parity generation and checking is disabled for this port. 1 = Parity generation and checking is enabled for this port. The PEN bit should not be modified when either the TMODE bit or the RMODE bit is 1. Even or odd parity is set with the EVN bit.
2	ABEN	**Address Bit Enable** 0 = The serial port does not generate or receive address bits. 1 = The serial port generates and receives address bits. The ABEN bit should not be modified when either the TMODE bit or the RMODE bit is 1. If enabled, the address bit is added to the 7- or 8-bit serial frame selected by the D7 bit. The state of the transmitted address bit is controlled by the AB bit in this register (HSPCON0) unless the EXDWR bit in the HSPCON1 register is set (see page 5-5). If the EXDWR bit is set, the transmitted address bit is controlled by the AB bit in the HSPTXD register (see page 5-13). The state of the received address bit is indicated by the AB bit in the HSPSTAT register (see page 5-8) unless the EXDRD bit in the HSPCON1 register is set (see page 5-6). If the EXDRD bit is set, the received address bit is indicated by the AB bit in the HSPRXD register (see page 5-15).
1	D7	**Data 7 Bits** 0 = Serial frames contain eight data bits. 1 = Serial frames contain seven data bits. The D7 bit should not be modified when either the TMODE bit or the RMODE bit is 1.
0	STP2	**Stop Bit Length** 0 = The serial port transmitter holds the TXD_HU line High for one bit time to generate the stop bit. 1 = The serial port transmitter holds the TXD_HU line High for two bit times to generate the stop bit. The STP2 bit should not be modified when either the TMODE bit or the RMODE bit is 1. The STP2 bit has no effect on received frames.

Programming Notes

Do not change bits FC, EVN, PEN, ABEN, D7, or STP2 while either the transmitter or receiver is enabled (that is, if the TMODE or RMODE bits are 1).

After an external or watchdog timer reset, flow control is disabled on the High-Speed UART. To enable flow control, software must set the ITF4 bit field in the SYSCON register to 01b or 10b (see page 16-3).

High-Speed Serial Port Control1 (HSPCON1) Offset 262h

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	TFEN	RFEN	TFLUSH	RFLUSH	ABAUD	*Res*			MEN	MAB2	MAB1	MAB0	BRKVAL	EXDWR	EXDRD	XTRN
Software Read/Write	R/W	R/W	W	W	R/W	R			R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Hardware Set/Clear	—	—	C	C	C	—			—	—	—	—	—	—	—	—
Chip Reset Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Microcontroller:	
Am186CC	!
Am186CH	✔
Am186CU	!

! *= See bit definitions.*

Register Description

The HSPCON1 register is the second of two control registers for the serial port. The HSPCON0 register is the first control register (see page 5-3).

Bit Definitions

Bit **Name** **Function**

15 TFEN **Transmit FIFO Enable**
0 = The transmit FIFO is disabled.

1 = The transmit FIFO is enabled.

When setting the TFEN bit, software should also set the TFLUSH bit to ensure that the FIFO is initialized. The TFEN bit should not be modified when either the TMODE bit or the RMODE bit in the HSPCON0 register is 1 (see page 5-4).

14 RFEN **Receive FIFO Enable**
0 = The receive FIFO is disabled.

1 = The receive FIFO is enabled.

When setting the RFEN bit, software should also set the RFLUSH bit to ensure that the FIFO is initialized. The RFEN bit should not be modified when either the TMODE bit or the RMODE bit in the HSPCON0 register is 1 (see page 5-4).

13 TFLUSH **Transmit FIFO Flush**
Writing a 1 to the TFLUSH bit causes the transmit FIFO to be flushed. Writing a 0 to this bit has no effect. Because the flush occurs immediately, this bit always reads as a 0.

12 RFLUSH **Receive FIFO Flush**
Writing a 1 to this bit causes the receive FIFO to be flushed. Writing a 0 to this bit has no effect. Since the flush takes place immediately, this bit always reads as a 0.

11 ABAUD **Auto Baud Enable**
0 = The serial port is not in autobaud detect mode.

1 = The serial port is in autobaud detect mode.

The ABAUD bit is cleared automatically when the serial port completes autobaud detection.

10–8 Res **Reserved**
For compatibility with future devices, always write this bit field with its chip reset default value.

7 MEN **Character Match Enable**
0 = Character matching is disabled.

1 = Character matching is enabled.

Bit	Name	Function
6	MAB2	**Match Address Bit 2** 0 = When the MAB2 bit is 0 and the ABEN bit in the HSPCON0 register is 1 (see page 5-4), a received character must have the address bit cleared (0) in order to match characters in the HSPM2 register. 1 = When this bit is 1 and the ABEN bit in the HSPCON0 register is 1, a received character must have the address bit set (1) in order to match characters in the HSPM2 register.
5	MAB1	**Match Address Bit 1** 0 = When the MAB1 bit is 0 and the ABEN bit in the HSPCON0 register is 1 (see page 5-4), a received character must have the address bit cleared (0) in order to match characters in the HSPM1 register. 1 = When this bit is 1 and the ABEN bit in the HSPCON0 register is 1, a received character must have the address bit set (1) in order to match characters in the HSPM1 register.
4	MAB0	**Match Address Bit 0** 0 = When the MAB0 bit is 0 and the ABEN bit in the HSPCON0 register is 1 (see page 5-4), a received character must have the address bit cleared (0) in order to match characters in the HSPM0 register. 1 = When this bit is 1 and the ABEN bit in the HSPCON0 register is 1, a received character must have the address bit set (1) in order to match characters in the HSPM0 register.
3	BRKVAL	**Break Value** 0 = The TXD_HU line is held Low during breaks (when the BRK bit is 1 in the HSPCON0 register, see page 5-4). 1 = The TXD_HU line is held High during breaks. Setting the BRKVAL bit to 1 allows the serial port transmit logic to time idle times on the TXD_HU line.
2	EXDWR	**Extended Write** 0 = The value of the AB bit in the HSPCON0 register (see page 5-3) is written to the AB bit of the HSPTXD register (see page 5-13). 1 = The AB bit in the upper byte of the HSPTXD register can be written.
1	EXDRD	**Extended Read** 0 = The upper byte of the HSPRXD register (see page 5-14) always reads 0. When the EXDRD bit is 0, receive data DMA requests are inhibited while a receive status interrupting condition exists. 1 = The bit fields in the upper byte of the HSPRXD register contain valid data. When the EXDRD bit is 1, receive data DMA requests are always generated when the RDR bit is set.
0	XTRN	**External Clock** 0 = The serial port uses the internal processor clock to generate the serial clock. 1 = The serial port uses the external UCLK pin to generate the serial clock. The XTRN bit should not be modified when either the TMODE bit or the RMODE bit in the HSPCON0 register is 1 (see page 5-4). ***Am186CC and Am186CU Microcontrollers:*** *On the Am186CC and Am186CU microcontrollers, the UCLK pin is multiplexed with the USB sample clock input (USBSCI) and SOF output (USBSOF) functions, which are controlled by the ESOF_EN and SAM_CLK_SEL bits in the ISCTL register (see page 4-15 and page 4-16).*

Programming Notes

Do not change bits TFEN, RFEN, or XTRN while either the transmitter or receiver is enabled (the TMODE or RMODE bits are 1, see page 5-4).

High-Speed Serial Port Status (HSPSTAT) Offset 264h

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	RTHRSH	TTHRSH	*Res*	OERIM	*Res*	MATCH	BRK	AB	RDR	THRE	FER	OER	PER	TEMT	IDLED	IDLE
Software Read/Write	R/W0	R/W0	R	R/W0	R	R/W0	R/W0	R/W0	R	R	R/W0	R/W0	R/W0	R	R/W0	R
Hardware Set/Clear	S/C	S/C	—	S	—	S	S	S/C	S/C	S/C	S	S	S	S/C	S	S/C
Chip Reset Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Microcontroller:

- **Am186CC** ✔
- **Am186CH** ✔
- **Am186CU** ✔

Register Description

This register provides information about the current status of the serial port. Each bit in this register can be enabled as an interrupt source by setting the corresponding bit in the HSPIMSK register (see page 5-10).

Bit Definitions

Bit **Name** **Function**

15 RTHRSH **Receive FIFO Threshold Reached**

0 = The serial port has not detected that the receive FIFO has reached the threshold value.

1 = The serial port has detected that the receive FIFO has reached the threshold value (i.e., the receiver is within 16 bytes of being filled, so software should read data more quickly).

If the receive FIFO is enabled (the RFEN bit in the HSPCON1 register is set, see page 5-5), the RTHRSH bit is set by hardware whenever the 32-byte receive FIFO contains 16 or more bytes of data.

The RTHRSH bit must be cleared by software, but the bit can only be cleared when the FIFO has been drained of sufficient data so that the data is no longer at or above the threshold value.

14 TTHRSH **Transmit FIFO Threshold Reached**

0 = The serial port has not detected that the transmit FIFO has reached the threshold value.

1 = The serial port has detected that the transmit FIFO has reached the threshold value (i.e., the transmitter is within 8 bytes of being emptied, so software should write data more quickly).

If the transmit FIFO is enabled (TFEN in HSPCON1 is set, see page 5-5), TTHRSH is set by hardware whenever the 16-byte transmit FIFO contains less than 8 bytes of data.

The TTHRSH bit must be cleared by software, but the bit can only be cleared when the FIFO has been written with sufficient data so that the data is no longer at or below the threshold value.

13 Res **Reserved**

For compatibility with future devices, always write this bit field with its chip reset default value.

12 OERIM **Overrun Error Detected—Immediate**

0 = No frames have been lost due to overrun.

1 = The serial port receiver has lost data due to an overrun error.

The OERIM bit must be cleared by software.

If the receive FIFO is enabled, the OERIM bit bypasses the FIFO to provide immediate notification of overrun, while the OER bit is not set until the HSPRXD register (see page 5-14) is loaded with the character on which the error was detected (i.e., the first data character after the overrun loss).

11 Res **Reserved**

For compatibility with future devices, always write this bit field with its chip reset default value.

Bit	Name	Function
10	MATCH	**Address Match Detected** 0 = No received frames have matched a character in the HSPMx registers (see the descriptions starting on page 5-20). 1 = The serial port has detected that a received character matches one of the characters in the HSPMx registers. The MATCH bit must be cleared by software. If the receive FIFO is enabled, the MATCH bit is not set until the HSPRXD register (see page 5-14) is loaded with the character on which the match was detected.
9	BRK	**Break Detected** 0 = No break condition has been detected on the receive data line. 1 = The serial port has detected a break condition on the receive data line. The BRK bit must be cleared by software. If the receive FIFO is enabled, the BRK bit is not set until the HSPRXD register (see page 5-14) is loaded with the character on which the break was detected.
8	AB	**Address Bit Detected** 0 = The serial port has not received a frame with the address bit High. 1 = The serial port has received a frame with the address bit High (1). The AB bit is only updated when the ABEN bit in the HSPCON0 register is 1 (see page 5-4). The AB bit must be cleared by software. If the receive FIFO is enabled, the AB bit is not set until the HSPRXD register (see page 5-14) is loaded with the character on which the address bit was detected.
7	RDR	**Receive Data Ready** 0 = The RDATA bit field of the HSPRXD register (see page 5-15) is undefined. 1 = The RDATA bit field of the HSPRXD register contains valid data. The RDR bit is cleared by a read of the HSPRXD register.
6	THRE	**Transmit Holding Register Empty** 0 = Writing to the HSPTXD register (see page 5-13) results in the loss (failure to transmit) of the most recently written data (i.e., software must wait until the THRE bit is 1 before writing transmit data). 1 = The transmitter is capable of accepting more data for transmission. When the THRE bit is 1, the HSPTXD register can be written without loss of data.
5	FER	**Framing Error Detected** 0 = No framing errors have been detected. The receiver only checks for framing errors during the first stop bit time, regardless of the setting of the STP2 bit in the HSPCON0 register (see page 5-4). 1 = The serial port receiver has detected a framing error. A framing error is a Low value on the RXD_HU line during the expected stop bit time. The FER bit must be cleared by software. If the receive FIFO is enabled, the FER bit is not set until the HSPRXD register (see page 5-14) is loaded with the character on which the error was detected.
4	OER	**Overrun Error Detected** 0 = No frames have been lost due to overrun. 1 = The serial port receiver has lost data due to an overrun error between this frame and the previous frame. The OER bit must be cleared by software. If the receive FIFO is enabled, the OER bit is not set until the HSPRXD register (see page 5-14) is loaded with the character on which the error was detected (i.e., the first data character after the overrun loss). The OERIM bit can be used for immediate notification of overrun errors.
3	PER	**Parity Error Detected** 0 = The serial port receiver has not detected a parity error on a received frame. 1 = The serial port receiver has detected a parity error on a received frame. The PER bit must be cleared by software. Parity detection is configured via the EVN and PEN bits in the HSPCON0 register (see page 5-4). The PER bit is always 0 if parity is not enabled. If the receive FIFO is enabled, the PER bit is not set until the HSPRXD register (see page 5-14) is loaded with the character on which the error was detected.

Bit	Name	Function
2	TEMT	**Transmitter Empty** 0 = The transmitter contains data to be transmitted over the TXD_HU line. 1 = The transmitter does not have any data that has not been transmitted. When FIFOs are enabled, the TEMT bit indicates that the FIFO is empty and the transmit shift register is empty. At this point, the transmitter can be disabled without loss of data.
1	IDLED	**Receive Line Idle Detected** 0 = The receiver has not been idle for 40 bit times, or no receive data is available. 1 = The receiver has detected a High bit on the RXD_HU line for 40 bit times while receive data is available. This indicates that the receive message appears to be complete, but the receiver contains data that has not been read. The IDLED bit must be cleared by software. The IDLED bit can be used to detect the end of a message when its last character does not fill the receive FIFO to its threshold.
0	IDLE	**Receive Line Idle** 0 = The receiver is currently receiving a data frame. 1 = The receiver is not currently in frame. This means that the RXD_HU line has been High continuously since the completion of reception on the last frame.

Programming Notes

If the EXDRD bit is 1 in the HSPCON1 register (see page 5-6), software can read the RDR, THRE, FER, OER, PER, MATCH, BRK, and AB bits in the HSPRXD register (see page 5-14) instead of the HSPSTAT register. Software can ignore the corresponding bits in the HSPSTAT register unless interrupts are enabled for the FER, OER, PER, MATCH, or BRK bits in the HSPIMSK register. These bits must be cleared in the HSPSTAT register if they generate an interrupt.

High-Speed Serial Port Interrupt Mask (HSPIMSK) Offset 266h

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	RTHRSH	TTHRSH	*Res*	OERIM	*Res*	MATCH	BRK	AB	RDR	THRE	FER	OER	PER	TEMT	IDLED	IDLE
Software Read/Write	R/W	R/W	R	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Hardware Set/Clear	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Chip Reset Default	0	0	0	0	0	0	1	0	1	1	1	1	1	0	0	0

Microcontroller:
Am186CC ✓
Am186CH ✓
Am186CU ✓

Register Description

This register provides bits for individually enabling or disabling interrupts based on the condition of various status bits. The bits in the HSPIMSK register have a one-to-one correspondence with the associated bits in the HSPSTAT register (see page 5-7). The default settings of these bits provide compatibility with Am186ES and Am186ED microcontroller software.

Note that when the interrupt controller determines that a receive status bit has generated an interrupt condition and extended reads are not enabled, receive DMA requests are inhibited. If extended reads are enabled, DMA requests are enabled.

Bit Definitions

Bit	Name	Function
15	RTHRSH	**Receive FIFO Threshold Interrupt Enable** 0 = The RTHRSH bit in the HSPSTAT register (see page 5-7) does not generate interrupt requests. 1 = When this RTHRSH bit is 1 and the RXIE bit in the HSPCON0 register is 1 (see page 5-3), the serial port generates interrupts when the receive FIFO threshold value is reached (the RTHRSH bit in the HSPSTAT register is 1).
14	TTHRSH	**Transmit FIFO Threshold Interrupt Enable** 0 = The TTHRSH bit in the HSPSTAT register (see page 5-7) does not generate interrupt requests. 1 = When this TTHRSH bit is 1 and the TXIE bit in the HSPCON0 register is 1 (see page 5-3), the serial port generates interrupts when the transmit FIFO threshold value is reached (the TTHRSH bit in the HSPSTAT register is 1).
13	Res	**Reserved** For compatibility with future devices, always write this bit field with its chip reset default value.
12	OERIM	**Overrun Error Detected—Immediate** 0 = The OERIM bit in the HSPSTAT register (see page 5-7) does not generate interrupt requests. 1 = When this OERIM bit is 1 and the RSIE bit in the HSPCON0 register is 1 (see page 5-3), the serial port generates interrupts when a receive overrun condition is detected (the OERIM bit in the HSPSTAT register is 1).
11	Res	**Reserved** For compatibility with future devices, always write this bit field with its chip reset default value.
10	MATCH	**Character Match Interrupt Enable** 0 = The MATCH bit in the HSPSTAT register (see page 5-8) does not generate interrupt requests. 1 = When this MATCH bit is 1 and the RSIE bit in the HSPCON0 register is 1 (see page 5-3), the serial port generates interrupts when a character match is detected (the MATCH bit in the HSPSTAT register is 1).
9	BRK	**Break Interrupt Enable** 0 = The BRK bit in the HSPSTAT register (see page 5-8) does not generate interrupt requests. 1 = When this BRK bit is 1 and the RSIE bit in the HSPCON0 register is 1 (see page 5-3), the serial port generates interrupts on break detection (the BRK bit in the HSPSTAT register is 1).

Bit	Name	Function
8	AB	**Address Bit Interrupt Enable** 0 = The AB bit in the HSPSTAT register (see page 5-8) does not generate interrupt requests. 1 = When this AB bit is 1 and the RSIE bit in the HSPCON0 register is 1 (see page 5-3), the serial port generates interrupts when a character is received with the address bit 1 (the AB bit in the HSPSTAT register is 1).
7	RDR	**Receive Data Ready Interrupt Enable** 0 = The RDR bit in the HSPSTAT register (see page 5-8) does not generate interrupt requests. 1 = When this RDR bit is 1 and the RXIE bit in the HSPCON0 register is 1 (see page 5-3), the serial port generates interrupts when valid received data is available (the RDR bit in the status register is 1).
6	THRE	**Transmit Holding Register Empty Interrupt Enable** 0 = The THRE bit in the HSPSTAT register (see page 5-8) does not generate interrupt requests. 1 = When this THRE bit is 1 and the TXIE bit in the HSPCON0 register is 1 (see page 5-3), the serial port generates interrupts when the transmitter can accept more data (the THRE bit in the HSPSTAT register is 1).
5	FER	**Framing Error Interrupt Enable** 0 = The FER bit in the HSPSTAT register (see page 5-8) does not generate interrupt requests. 1 = When this FER bit is 1 and the RSIE bit in the HSPCON0 register is 1 (see page 5-3), the serial port generates interrupts when a framing error is detected (the FER bit in the HSPSTAT register is 1).
4	OER	**Overrun Error Interrupt Enable** 0 = The OER bit in the HSPSTAT register (see page 5-8) does not generate interrupt requests. 1 = When this OER bit is 1 and the RSIE bit in the HSPCON0 register is 1 (see page 5-3), the serial port generates interrupts when a receive overrun condition is detected (the OER bit in the HSPSTAT register is 1).
3	PER	**Parity Error Interrupt Enable** 0 = The PER bit in the HSPSTAT register (see page 5-8) does not generate interrupt requests. 1 = When this PER bit is 1 and the RSIE bit in the HSPCON0 register is 1 (see page 5-3), the serial port generates interrupts when a parity error is detected (the PER bit in the HSPSTAT register is 1).
2	TEMT	**Transmitter Empty Interrupt Enable** 0 = The TEMT bit in the HSPSTAT register (see page 5-9) does not generate interrupt requests. 1 = When this TEMT bit is 1 and the TXIE bit in the HSPCON0 register is 1 (see page 5-3), the serial port generates interrupts when the transmitter does not have any data to transmit over the line (the TEMT bit in the HSPSTAT register is 1). When FIFOs are enabled, setting the TEMT bit in this register (HSPIMSK) enables the serial port to generate interrupts when the FIFO is empty and the transmit shift register is empty.
1	IDLED	**Receive Line Idle Detected Interrupt Enable** 0 = The IDLED bit in the HSPSTAT register (see page 5-9) does not generate interrupt requests. 1 = When this IDLED bit is 1 and the RXIE bit in the HSPCON0 register is 1 (see page 5-3), the serial port generates interrupts when the receive data line has been idle for 40 bit times and receive data is available (the IDLED bit in the HSPSTAT register is 1).
0	IDLE	**Receive Line Idle Interrupt Enable** 0 = The IDLE bit in the HSPSTAT register (see page 5-9) does not generate interrupt requests. 1 = When this IDLE bit is 1 and the RSIE bit in the HSPCON0 register is 1 (see page 5-3), the serial port generates interrupts when the receive data line is idle (the IDLE bit in the HSPSTAT register is 1.)

Programming Notes

There are two levels of interrupt enable. This register (HSPIMSK) provides the second or lower level. The RSIE, RXIE, and TXIE bits in the control register provide the first or higher level of interrupt enable for the serial port. When the RSIE bit is 0, the IDLE, PER, OER, FER, AB, BRK, OERIM, and MATCH bit interrupt conditions are disabled, regardless of the settings of the associated bits in this register. When the RXIE bit is 0, the IDLED, RDR, and RTHRSH interrupt conditions are disabled regardless of the settings of the associated bits in this register. When the TXIE bit is 0, the TEMT, THRE, and TTHRSH bit interrupt conditions are disabled regardless of the settings of the associated bits in this register.

In addition to these requirements, software must configure the interrupt Channel 3 Control (CH3CON) register to select the interrupt channel’s internal source (High-Speed UART) and enable the channel before High-Speed UART interrupts can be generated. See Chapter 9, “Interrupt Controller Registers”.

High-Speed Serial Port Transmit Data (HSPTXD) Offset 268h

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	*Res*							AB	TDATA							
Software Read/Write	R							R/W	R/W							
Hardware Set/Clear	—							—	—							
Chip Reset Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Microcontroller:
Am186CC ✓
Am186CH ✓
Am186CU ✓

Register Description

This register provides data to the serial port transmitter. When the THRE bit in the HSPSTAT register is 1 (see page 5-8), the HSPTXD register can be written with data to be transmitted over the serial port. If the HSPTXD register is written when the THRE bit is not 1, data previously written to this register is lost. Data that has already been written to the transmit shift register (not visible to software) is not affected by a write to this register.

Bit Definitions

Bit	Name	Function
15–9	Res	**Reserved** For compatibility with future devices, always write this bit field with its chip reset default value.
8	AB	**Address Bit** The AB bit in this register contains the value to be transmitted for the address bit. This bit can only be written when the EXDWR bit is 1 in the HSPCON1 register (see page 5-6). If the EXDWR bit is not 1, the value of the AB bit in the HSPCON0 register (see page 5-3) is placed in the AB bit in this register (HSPTXD) when the TDATA bit field is written.
7–0	TDATA	**Transmit Data** The TDATA bit field contains data to be transmitted through the serial port.

Programming Notes

High-Speed Serial Port Receive Data (HSPRXD) Offset 26Ah

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	RDR	THRE	FER	OER	PER	MATCH	BRK	AB	RDATA							
Software Read/Write	R	R	R	R	R	R	R	R	R							
Hardware Set/Clear	S/C	S/C	S/C	S/C	S/C	S/C	S/C	S/C	S/C							
Chip Reset Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Microcontroller:

Am186CC	✓
Am186CH	✓
Am186CU	✓

Register Description

When the RDR bit in the HSPSTAT register (see page 5-7) is 1, the HSPRXD register contains valid data received over the serial line. When extended reads are enabled, the value of the RDR bit is available in bit 15 of this register. The RDATA, FER, OER, PER, MATCH, BRK, and AB bit fields only contain valid data when the RDR bit is 1. The THRE bit field contains valid data on all reads of this register when extended reads are enabled.

Reading the HSPRXD register advances the receive FIFO if enabled and also clears the RDR bit in the HSPSTAT register if no further receive data is available.

Bit Definitions

Bit	Name	Function
15	RDR	**Receive Data Ready** 0 = The RDATA bit field does not contain valid data, and the FER, OER, PER, MATCH, BRK, and AB bit fields all read 0. 1 = The RDATA bit field contains valid data, and the FER, OER, PER, MATCH, BRK, and AB bit fields contain valid status on that data. The RDR bit is only valid if the EXDRD bit is 1 in the HSPCON1 register (see page 5-6).
14	THRE	**Transmit Holding Register Empty** 0 = Writing to the HSPTXD register (see page 5-13) results in the loss (failure to transmit) of the most recently written data. 1 = The transmitter is capable of accepting more data for transmission. When the THRE bit is 1, the HSPTXD register can be written without loss of data. The THRE bit is only valid if the EXDRD bit is 1 in the HSPCON1 register (see page 5-6).
13	FER	**Framing Error** 0 = No framing error was detected on the received frame. The receiver only checks for framing errors during the first stop bit time, regardless of the setting of the STP2 bit in the HSPCON0 register (see page 5-4). 1 = A framing error was detected on this frame. A framing error is a Low value on the RXD_HU line during the expected stop bit time. The FER bit is only valid if the EXDRD bit is 1 in the HSPCON1 register (see page 5-6).
12	OER	**Overrun Error** 0 = No overrun error occurred between this frame and the previous frame. 1 = An overrun error occurred between this frame and the previous frame. The OER bit is only valid when the EXDRD bit is 1 in the HSPCON1 register (see page 5-6).

Bit	Name	Function
11	PER	**Parity Error** 0 = No parity error was detected on this frame. 1 = A parity error was detected on this frame. Parity detection is configured via the EVN and PEN bits in the HSPCON0 register (see page 5-4). The PER bit is always 0 if parity is not enabled. The PER bit is only valid if the EXDRD bit is 1 in the HSPCON1 register (see page 5-6).
10	MATCH	**Match** 0 = The received frame does not match any of the characters in the HSPMx registers (see the descriptions starting on page 5-20). The MATCH bit is always 0 if character matching is not enabled. 1 = The serial port detected that the received character matches one of the characters in the HSPMx registers. The MATCH bit is only valid if the EXDRD bit is 1 in the HSPCON1 register (see page 5-6).
9	BRK	**Break** 0 = The received frame has not been detected as a break. 1 = The serial port detected a break condition on the received frame. The BRK bit is only valid if the EXDRD bit is 1 in the HSPCON1 register (see page 5-6).
8	AB	**Address Bit** 0 = The address bit of the received frame was read as Low. 1 = The address bit of the received frame was read as High. The AB bit is only valid if the EXDRD bit is 1 in the HSPCON1 register (see page 5-6).
7–0	RDATA	**Received Data** The RDATA bit field holds the value of the last character received over the HSP_RXD line when FIFOs are not enabled. The RDATA bit field holds the character at the top of the FIFO when FIFOs are enabled. This bit field is valid only when the RDR bit in the status register is 1. Reading this register (HSPRXD) clears the RDR bit in the HSPSTAT register (see page 5-8) if no further receive data is available.

Programming Notes

If the EXDRD bit is 1 in the HSPCON1 register (see page 5-6), software can read the RDR, THRE, FER, OER, PER, MATCH, BRK, and AB bits in the HSPRXD register instead of the HSPSTAT register (see page 5-7). Software can ignore the corresponding bits in the HSPSTAT register unless interrupts are enabled for the FER, OER, PER, MATCH, or BRK bits in the HSPIMSK register. These bits must be cleared in the HSPSTAT register if they generate an interrupt.

High-Speed Serial Port Receive Data Peek (HSPRXDP) Offset 26Ch

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	RDR	THRE	FER	OER	PER	MATCH	BRK	AB	RDATA							
Software Read/Write	R	R	R	R	R	R	R	R	R							
Hardware Set/Clear	S/C	S/C	S/C	S/C	S/C	S/C	S/C	S/C	S/C							
Chip Reset Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Microcontroller:	
Am186CC	✔
Am186CH	✔
Am186CU	✔

Register Description

This register provides a means of reading the HSPRXD register (see page 5-14) without changing the serial port status or advancing the receive FIFO. The bit definitions are otherwise the same.

Bit Definitions

Bit **Name** **Function**

15 RDR **Receive Data Ready (Same as HSPRXD)**

0 = The RDATA bit field does not contain valid data, and the FER, OER, PER, MATCH, BRK, and AB bit fields all read 0.

1 = The RDATA bit field contains valid data, and the FER, OER, PER, MATCH, BRK, and AB bit fields contain valid status on that data.

The RDR bit is only valid if the EXDRD bit is 1 in the HSPCON1 register (see page 5-6).

14 THRE **Transmit Holding Register Empty (Same as HSPRXD)**

0 = Writing to the HSPTXD register (see page 5-13) results in the loss (failure to transmit) of the most recently written data.

1 = The transmitter is capable of accepting more data for transmission. When the THRE bit is 1, the HSPTXD register can be written without loss of data.

The THRE bit is only valid if the EXDRD bit is 1 in the HSPCON1 register (see page 5-6).

13 FER **Framing Error (Same as HSPRXD)**

0 = No framing error was detected on the received frame. The receiver only checks for framing errors during the first stop bit time, regardless of the setting of the STP2 bit in the HSPCON0 register (see page 5-4).

1 = A framing error was detected on this frame. A framing error is a Low value on the RXD_HU line during the expected stop bit time.

The FER bit is only valid if the EXDRD bit is 1 in the HSPCON1 register (see page 5-6).

12 OER **Overrun Error (Same as HSPRXD)**

0 = No overrun error occurred between this frame and the previous frame.

1 = An overrun error occurred between this frame and the previous frame.

The OER bit is only valid when the EXDRD bit is 1 in the HSPCON1 register (see page 5-6).

11 PER **Parity Error (Same as HSPRXD)**

0 = No parity error was detected on this frame.

1 = A parity error was detected on this frame.

Parity detection is configured via the EVN and PEN bits in the HSPCON0 register (see page 5-4). The PER bit is always 0 if parity is not enabled.

The PER bit is only valid if the EXDRD bit is 1 in the HSPCON1 register (see page 5-6).

Bit	Name	Function
10	MATCH	**Match (Same as HSPRXD)** 0 = The received frame does not match any of the characters in the HSPMx registers (see the descriptions starting on page 5-20). The MATCH bit is always 0 if character matching is not enabled. 1 = The serial port detected that the received character matches one of the characters in the HSPMx registers. The MATCH bit is only valid if the EXDRD bit is 1 in the HSPCON1 register (see page 5-6).
9	BRK	**Break (Same as HSPRXD)** 0 = The received frame has not been detected as a break. 1 = The serial port detected a break condition on the received frame. The BRK bit is only valid if the EXDRD bit is 1 in the HSPCON1 register (see page 5-6).
8	AB	**Address Bit (Same as HSPRXD)** 0 = The address bit of the received frame was read as Low. 1 = The address bit of the received frame was read as High. The AB bit is only valid if the EXDRD bit is 1 in the HSPCON1 register (see page 5-6).
7–0	RDATA	**Received Data** The RDATA bit field holds the value of the last character received over the HSP_RXD line when FIFOs are not enabled. The RDATA bit field holds the character at the top of the FIFO when FIFOs are enabled. This bit field is valid only when the RDR bit in the HSPSTAT register is 1 (see page 5-8).

Programming Notes

If the EXDRD bit is 1 in the HSPCON1 register (see page 5-6), software can read the RDR, THRE, FER, OER, PER, MATCH, BRK, and AB bits in the HSPRXD register instead of the HSPSTAT register (see page 5-7). Software can ignore the corresponding bits in the HSPSTAT register unless interrupts are enabled for the FER, OER, PER, MATCH, or BRK bits in the HSPIMSK register. These bits must be cleared in the HSPSTAT register if they generate an interrupt.

High-Speed Serial Port Baud Rate Divisor (HSPBDV) Offset 26Eh

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	BAUDDIV															
Software Read/Write	R/W															
Hardware Set/Clear	S/C															
Chip Reset Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Microcontroller:	
Am186CC	✓
Am186CH	✓
Am186CU	✓

Register Description

This register specifies a clock divisor for the generation of the serial clock. The HSPBDV register should not be modified while either the RMODE bit or the TMODE bit in the HSPCON0 register is 1 (see page 5-4). In a startup situation, a non-zero value must be written into this register (HSPBDV) before setting the RMODE bit or the TMODE bit to 1. If the BAUDDIV bit field is cleared to 0, data is neither received nor transmitted.

The baud divisor register determines the baud rate using the following formula:

BAUDDIV = (UART frequency / (16 · baud rate))

Where UART frequency is either the processor frequency or the UCLK frequency if external clocking is enabled.

Bit Definitions

Bit	Name	Function
15–0	BAUDDIV	**Baud Divisor** The BAUDDIV bit field specifies the divisor used to generate the serial clock. When autobaud is enabled in the HSPCON1 register, the BAUDDIV bit field is loaded automatically with the detected baud rate divisor, or with a corresponding valid divisor as programmed in registers HSPAB0–HSPAB3 (see the descriptions starting on page 5-23).

Programming Notes

Software can set baud rates up to 1/16th of the UART frequency (by setting the BAUDDIV bit field to 1) if the connecting device’s baud rate formula is known. But it may be necessary to use special UART frequencies and non-integer baud rates.

Table 5-2 shows the value of the BAUDDIV bit field, in both decimal and hexadecimal, for common baud rates and clock frequencies. The actual baud rate varies from the nominal rate by the amount shown in the % Error column.

Table 5-2 Baud Rate Table for High-Speed UART

	Serial Port Clock Frequency (Processor Frequency or UCLK Frequency)											
	24 MHz		25 MHz		40 MHz		44.2 MHz		48 MHz		50 MHz	
Baud Rate	Divisor	% Error	Divisor	% Error	Divisor	% Error	Divisor	% Error	Divisor	% Error	Divisor	% Error
300	5000d 1388h	0	5208d 1458h	0	8333d 208Dh	0	9208d 23F8h	0	10000d 2710h	0	10417d 28B1h	0
600	2500d 09C4h	0	2604d 0A2Ch	0	4167d 1047h	0	4604d 11FCh	0	5000d 1388h	0	5208d 1458h	0
1200	1250d 04E2h	0	1302d 0516h	0	2083d 0823h	0	2302d 08FEh	0	2500d 09C4h	0	2604d 0A2Ch	0
2400	625d 0271h	0	651d 028Bh	0	1042d 0412h	0	1151d 047Fh	0	1250d 04E2h	0	1302d 0516h	0
9600	156d 9Ch	0.2	163d A3h	-0.1	260d 0104h	0.2	288d 0120h	-0.1	313d 0139h	-0.2	326d 0146h	-0.1
19200	78d 4Eh	0.2	81d 51h	0.5	130d 82h	0.2	144d 90h	-0.1	156d 9Ch	0.2	163d A3h	-0.1
38400	39d 27h	0.2	41d 29h	-0.8	65d 41h	0.2	72d 48h	-0.1	78d 4Eh	0.2	81d 51h	0.5
57600	26d 1Ah	0.2	27d 1Bh	0.5	43d 2Bh	0.9	48d 30h	-0.1	52d 34h	0.2	54d 36h	0.5
115200	13d 0Dh	0.2	14d 0Eh	-3.2	22d 16h	-1.4	24d 18h	-0.1	26d 1Ah	0.2	27d 1Bh	0.5
230400	7d 07h	-7.5	7d 07h	-3.2	11d 0Bh	-1.4	12d 0Ch	-0.1	13d 0Dh	0.2	14d 0Eh	-3.2
460800	3d 03h	7.8	3d 03h	11.5	5d 05h	7.8	6d 06h	-0.1	7d 07h	-7.5	7d 07h	-3.2

High-Speed Serial Port Character Match 0 (HSPM0) Offset 270h

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	MCHR1								MCHR0							
Software Read/Write	R/W								R/W							
Hardware Set/Clear	—								—							
Chip Reset Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Microcontroller:	
Am186CC	✔
Am186CH	✔
Am186CU	✔

Register Description

This register can be programmed with two characters for use with automatic character matching. Match characters should also be programmed into the HSPM1 and HSPM2 registers (see page 5-21 and page 5-22).

Bit Definitions

Bit	Name	Function
15–8	MCHR1	**Match Character 1** When the MEN bit in the HSPCON1 register is 1 (see page 5-5), incoming characters are compared against the MCHR1 bit field (and against the MAB0 bit if the ABEN bit in the HSPCON0 register is 1). If the incoming character matches, the MATCH bit in the HSPSTAT register is set to 1, indicating that a match has occurred. If the MEN bit in the HSPCON1 register is 0, the MCHR1 bit field is ignored.
7–0	MCHR0	**Match Character 0** When the MEN bit in the HSPCON1 register is 1, incoming characters are compared against the MCHR0 bit field (and against the MAB0 bit if the ABEN bit in the HSPCON0 register is 1). If the incoming character matches, the MATCH bit in the HSPSTAT register is set to 1, indicating that a match has occurred. If the MEN bit in the HSPCON1 register is 0, the MCHR0 bit field is ignored.

Programming Notes

All six match characters should be programmed unless the default value (00h) is a character to be matched. If fewer than six unique characters need to be matched, program the remaining match character bit fields with one of the characters to be matched.

For example, if the characters 11h, 13h, and 04h are to be matched, software can program the MCHR0, MCHR1, and MCHR2 bit fields with these values. But if the remaining match bit fields are left alone, their default value (00h) is also matched. To avoid matching the default value, you can use one of the three already programmed values to fill the remaining bit fields. (For this example, software can set MCHR3=MCHR4=MCHR5=04h.)

High-Speed Serial Port Character Match 1 (HSPM1) Offset 272h

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	MCHR3								MCHR2							
Software Read/Write	R/W								R/W							
Hardware Set/Clear	—								—							
Chip Reset Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Microcontroller:	
Am186CC	✔
Am186CH	✔
Am186CU	✔

Register Description

This register can be programmed with two characters for use with automatic character matching. Match characters should also be programmed into the HSPM0 and HSPM2 registers (see page 5-20 and page 5-22).

Bit Definitions

Bit	Name	Function
15–8	MCHR3	**Match Character 3** When the MEN bit in the HSPCON1 register is 1 (see page 5-5), incoming characters are compared against the MCHR3 bit field (and against the MAB1 bit if the ABEN bit in the HSPCON0 register is 1). If the incoming character matches, the MATCH bit in the HSPSTAT register is set to 1, indicating that a match has occurred. If the MEN bit in the HSPCON1 register is 0, the MCHR3 bit field is ignored.
7–0	MCHR2	**Match Character 2** When the MEN bit in the HSPCON1 register is 1, incoming characters are compared against the MCHR2 bit field (and against the MAB1 bit if the ABEN bit in the HSPCON0 register is 1). If the incoming character matches, the MATCH bit in the HSPSTAT register is set to 1, indicating that a match has occurred. If the MEN bit in the HSPCON1 register is 0, the MCHR2 bit field is ignored.

Programming Notes

All six match characters should be programmed unless the default value (00h) is a character to be matched. If fewer than six unique characters need to be matched, program the remaining match character bit fields with one of the characters to be matched.

For example, if the characters 11h, 13h, and 04h are to be matched, software can program the MCHR0, MCHR1, and MCHR2 bit fields with these values. But if the remaining match bit fields are left alone, their default value (00h) is also matched. To avoid matching the default value, you can use one of the three already programmed values to fill the remaining bit fields. (For this example, software can set MCHR3=MCHR4=MCHR5=04h.)

High-Speed Serial Port Character Match 2 (HSPM2) Offset 274h

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	MCHR5								MCHR4							
Software Read/Write	R/W								R/W							
Hardware Set/Clear	—								—							
Chip Reset Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Microcontroller:	
Am186CC	✔
Am186CH	✔
Am186CU	✔

Register Description

This register can be programmed with two characters for use with automatic character matching. Match characters should also be programmed into the HSPM0 and HSPM1 registers (see page 5-20 and page 5-21).

Bit Definitions

Bit	Name	Function
15–8	MCHR5	**Match Character 5** When the MEN bit in the HSPCON1 register is 1 (see page 5-5), incoming characters are compared against the MCHR5 bit field (and against the MAB2 bit if the ABEN bit in HSPCON0 is 1). If the incoming character matches, the MATCH bit in the HSPSTAT register is set to 1, indicating that a match has occurred. If the MEN bit in the HSPCON1 register is 0, the MCHR5 bit field is ignored.
7–0	MCHR4	**Match Character 4** When the MEN bit in the HSPCON1 register is 1, incoming characters are compared against the MCHR4 bit field (and against the MAB2 bit if the ABEN bit in HSPCON0 is 1). If the incoming character matches, the MATCH bit in the HSPSTAT register is set to 1, indicating that a match has occurred. If the MEN bit in the HSPCON1 register is 0, the MCHR4 bit field is ignored.

Programming Notes

All six match characters should be programmed unless the default value (00h) is a character to be matched. If fewer than six unique characters need to be matched, program the remaining match character bit fields with one of the characters to be matched.

For example, if the characters 11h, 13h, and 04h are to be matched, software can program the MCHR0, MCHR1, and MCHR2 bit fields with these values. But if the remaining match bit fields are left alone, their default value (00h) is also matched. To avoid matching the default value, you can use one of the three already programmed values to fill the remaining bit fields. (For this example, software can set MCHR3=MCHR4=MCHR5=04h.)

High-Speed Serial Port Autobaud 0 (HSPAB0) Offset 276h

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	ABDIV0								ABTHRSH0							
Software Read/Write	R/W								R/W							
Hardware Set/Clear	—								—							
Chip Reset Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Microcontroller:	
Am186CC	✔
Am186CH	✔
Am186CU	✔

Register Description

Registers HSPAB0–HSPAB3 can be used to specify four baud rate divisors to be used in place of the autobaud-calculated divisor if the calculated divisor falls between specified thresholds.

Programming these registers with valid baud rate divisors can help compensate for the increased effect of signal distortion on autobaud detection at high baud rates.

Bit Definitions

Bit	Name	Function
15–8	ABDIV0	**Autobaud Valid Divisor 0** The valid baud divisor programmed in the HSPAB0 register must be the smallest of the four divisors being programmed. This value is selected as the baud divisor during autobaud if the calculated baud divisor is less than or equal to the ABTHRSH0 bit field value.
7–0	ABTHRSH0	**Autobaud Threshold 0** The ABTHRSH0 bit field threshold value must be the smallest of the four threshold values being programmed. The value must be smaller than the ABDIV1 bit field value and larger than the ABDIV0 bit field value.

Programming Notes

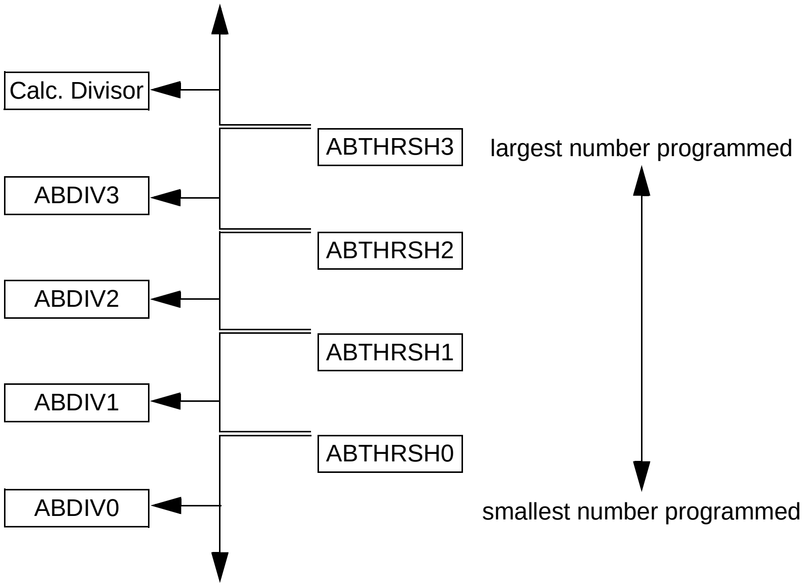
Do not change any HSPABx register while autobaud is enabled (i.e., while the ABAUD bit is 1 in the HSPCON1 register, see page 5-5).

To always use the calculated baud divisor when autobaud is enabled, clear the ABTHRSH3 bit field in the HSPAB3 register, or leave the field at its default value (00h).

Software must always program the HSPAB3 register with the largest value, program the HSPAB2 register with the next largest value, and so on. The HSPAB3 register must be programmed with the highest value even when using fewer than four valid divisor values. Software must clear the unused HSPABx registers (starting with HSPAB0) or leave them at their default values (00h).

Figure 5-1 on page 5-24 illustrates the relationship between the programmed ABDIVx and ABTHRSHx values.

Figure 5-1 Autobaud Enhancement



High-Speed Serial Port Autobaud 1 (HSPAB1) Offset 278h

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	ABDIV1								ABTHRSH1							
Software Read/Write	R/W								R/W							
Hardware Set/Clear	—								—							
Chip Reset Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Microcontroller:
Am186CC ✔
Am186CH ✔
Am186CU ✔

Register Description

Registers HSPAB0–HSPAB3 can be used to specify four baud rate divisors to be used in place of the autobaud-calculated divisor if the calculated divisor falls between specified thresholds.

Programming these registers with valid baud rate divisors can help compensate for the increased effect of signal distortion on autobaud detection at high baud rates.

Bit Definitions

Bit	Name	Function
15–8	ABDIV1	**Autobaud Valid Divisor 1** This valid baud divisor value must be larger than the ABDIV0 bit field value and smaller than the ABDIV2 bit field value. The ABDIV1 bit field value is selected as the baud divisor during autobaud if the calculated baud divisor is less than or equal to the ABTHRSH1 bit field value and greater than the ABTHRSH0 bit field value.
7–0	ABTHRSH1	**Autobaud Threshold 1** The ABTHRSH1 bit field threshold value must be the second smallest of the four threshold values being programmed. The value must be smaller than the ABDIV2 bit field value and larger than the ABDIV1 bit field value.

Programming Notes

Do not change any HSPABx register while autobaud is enabled (i.e., while the ABAUD bit is 1 in the HSPCON1 register, see page 5-5).

To always use the calculated baud divisor when autobaud is enabled, clear the ABTHRSH3 bit field in the HSPAB3 register, or leave the field at its default value (00h).

Software must always program the HSPAB3 register with the largest value, program the HSPAB2 register with the next largest value, and so on. The HSPAB3 register must be programmed with the highest value even when using fewer than four valid divisor values. Software must clear the unused HSPABx registers (starting with HSPAB0) or leave them at their default values (00h).

Figure 5-1 on page 5-24 illustrates the relationship between the programmed ABDIVx and ABTHRSHx values.

High-Speed Serial Port Autobaud 2 (HSPAB2) Offset 27Ah

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	ABDIV2								ABTHRSH2							
Software Read/Write	R/W								R/W							
Hardware Set/Clear	—								—							
Chip Reset Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Microcontroller:	
Am186CC	✓
Am186CH	✓
Am186CU	✓

Register Description

Registers HSPAB0–HSPAB3 can be used to specify four baud rate divisors to be used in place of the autobaud-calculated divisor if the calculated divisor falls between specified thresholds.

Programming these registers with valid baud rate divisors can help compensate for the increased effect of signal distortion on autobaud detection at high baud rates.

Bit Definitions

Bit	Name	Function
15–8	ABDIV2	**Autobaud Valid Divisor 2** This valid baud divisor value must be larger than the ABDIV1 bit field value and smaller than the ABDIV3 bit field value. This value is selected as the baud divisor during autobaud if the calculated baud divisor is less than or equal to the ABTHRSH2 bit field value and greater than the ABTHRSH1 bit field value.
7–0	ABTHRSH2	**Autobaud Threshold 2** The ABTHRSH2 bit field threshold value must be the second largest of the four threshold values being programmed. The value must be smaller than the ABDIV3 bit field value and larger than the ABDIV2 bit field value.

Programming Notes

Do not change any HSPABx register while autobaud is enabled (i.e., while the ABAUD bit is 1 in the HSPCON1 register, see page 5-5).

To always use the calculated baud divisor when autobaud is enabled, clear the ABTHRSH3 bit field in the HSPAB3 register, or leave the field at its default value (00h).

Software must always program the HSPAB3 register with the largest value, program the HSPAB2 register with the next largest value, and so on. The HSPAB3 register must be programmed with the highest value even when using fewer than four valid divisor values. Software must clear the unused HSPABx registers (starting with HSPAB0) or leave them at their default values (00h).

Figure 5-1 on page 5-24 illustrates the relationship between the programmed ABDIVx and ABTHRSHx values.

High-Speed Serial Port Autobaud 3 (HSPAB3) Offset 27Ch

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	ABDIV3								ABTHRSH3							
Software Read/Write	R/W								R/W							
Hardware Set/Clear	—								—							
Chip Reset Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Microcontroller:

Am186CC	✔
Am186CH	✔
Am186CU	✔

Register Description

Registers HSPAB0–HSPAB3 can be used to specify four baud rate divisors to be used in place of the autobaud-calculated divisor if the calculated divisor falls between specified thresholds.

Programming these registers with valid baud rate divisors can help compensate for the increased effect of signal distortion on autobaud detection at high baud rates.

Bit Definitions

Bit	Name	Function
15–8	ABDIV3	**Autobaud Valid Divisor 3** This valid baud divisor value must be the largest of the four divisors programmed. This value is selected as the baud divisor during autobaud if the calculated baud divisor is less than or equal to the ABTHRSH3 bit field value and greater than the ABTHRSH2 bit field value.
7–0	ABTHRSH3	**Autobaud Threshold 3** The ABTHRSH3 bit field threshold value must be the largest of the four threshold values being programmed. The value must be larger than the ABDIV3 bit field value. If the calculated baud divisor value is larger than the ABTHRSH3 bit field threshold value, then the calculated value is used rather than one of the programmed values. Thus, if the programmed value of the ABTHRSH3 bit field is 0, the calculated baud divisor value is used.

Programming Notes

Do not change any HSPABx register while autobaud is enabled (i.e., while the ABAUD bit is 1 in the HSPCON1 register, see page 5-5).

To always use the calculated baud divisor when autobaud is enabled, clear the ABTHRSH3 bit field in the HSPAB3 register, or leave the field at its default value (00h).

Software must always program the HSPAB3 register with the largest value, program the HSPAB2 register with the next largest value, and so on. The HSPAB3 register must be programmed with the highest value even when using fewer than four valid divisor values. Software must clear the unused HSPABx registers (starting with HSPAB0) or leave them at their default values (00h).

Figure 5-1 on page 5-24 illustrates the relationship between the programmed ABDIVx and ABTHRSHx values.

Serial Port Control 0 (SPCON0) Offset 280h

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	*Res*			RSIE	BRK	AB	FC	TXIE	RXIE	TMODE	RMODE	EVN	PEN	ABEN	D7	STP2
Software Read/Write	R			R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Hardware Set/Clear	—			—	—	C	—	—	—	—	—	—	—	—	—	—
Chip Reset Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Microcontroller:	
Am186CC	✔
Am186CH	✔
Am186CU	✔

Register Description

The SPCON0 register is the first of two control registers for the serial port. The SPCON1 register is the second control register (see page 5-30).

Bit Definitions

Bit **Name** **Function**

15–13 Res **Reserved**
For compatibility with future devices, always write this bit field with its chip reset default value.

12 RSIE **Receive Status Interrupt Enable**
0 = Receive status interrupts are disabled.

1 = Receive status interrupts are enabled.

The bits in the SPSTAT register that generate receive status interrupts are IDLE, PER, OER, FER, AB, and BRK (see the descriptions starting on page 5-31).

11 BRK **Send Break**
0 = The SP_TXD line follows the output data.

1 = The SP_TXD line is held to the value determined by the BRKVAL bit in the SPCON1 register (see page 5-30).

10 AB **Address Bit**
0 = The address bit for the next frame is driven Low if address generation is enabled by setting the ABEN bit to 1.

1 = The address bit for the next frame is driven High if address generation is enabled.

The AB bit clears automatically after the value is read into the transmit shift register.

The AB bit in this register has no effect if the EXDWR bit in the SPCON1 register is set (see page 5-30). If the EXDWR bit is set, the transmitted address bit is controlled by the AB bit in the SPTXD register (see page 5-35).

9 FC **Flow Control Enable**
0 = Hardware flow control (CTS/RTR) is disabled.

1 = Hardware flow control is enabled.

The FC bit should not be modified when either the TMODE bit or the RMODE bit is 1.

8 TXIE **Transmit Data Interrupt Enable**
0 = Transmit interrupts are disabled.

1 = Transmit interrupts are enabled.

The bits in the SPSTAT register that generate transmit interrupts are THRE and TEMT (see the descriptions starting on page 5-31).

7 RXIE **Receive Data Interrupt Enable**
0 = Receive data interrupts are disabled.

1 = Receive data interrupts are enabled.

The bits in the SPSTAT register that generate receive interrupts are IDLED and RDR (see the descriptions starting on page 5-31).

Bit	Name	Function
6	TMODE	**Transmitter Enable** 0 = The transmitter portion of the serial port is disabled. 1 = The transmitter portion of the serial port is enabled. Software must configure the SPBDV, SPCON1, and SPIMSK registers appropriately before setting the TMODE bit. Transmitter bits in this register (SPCON0) must be configured either before the TMODE bit is set or at the same time the bit is set.
5	RMODE	**Receiver Enable** 0 = The receiver portion of the serial port is disabled. 1 = The receiver portion of the serial port is enabled. Software must configure the SPBDV, SPCON1, and SPIMSK registers appropriately before setting the RMODE bit. Receiver bits in this register (SPCON0) must be configured either before the RMODE bit is set or at the same time the bit is set.
4	EVN	**Even Parity** 0 = When this bit is 0 and the PEN bit is 1, odd parity is in effect for the serial port. 1 = When this bit is 1 and the PEN bit is 1, even parity is in effect for the serial port. The EVN bit should not be modified when either the TMODE bit or the RMODE bit is 1. The EVN bit has no effect when the PEN bit is 0.
3	PEN	**Parity Enable** 0 = Parity generation and checking is disabled for this port. 1 = Parity generation and checking is enabled for this port. The PEN bit should not be modified when either the TMODE bit or the RMODE bit is 1. Even or odd parity is set with the EVN bit.
2	ABEN	**Address Bit Enable** 0 = The serial port does not generate or receive address bits. 1 = The serial port generates and receives address bits. The ABEN bit should not be modified when either the TMODE bit or the RMODE bit is 1. If enabled, the address bit is added to the 7- or 8-bit serial frame selected by the D7 bit. The state of the transmitted address bit is controlled by the AB bit in this register (SPCON0) unless the EXDWR bit in the SPCON1 register is set (see page 5-30). If the EXDWR bit is set, the transmitted address bit is controlled by the AB bit in the SPTXD register (see page 5-35). The state of the received address bit is indicated by the AB bit in the SPSTAT register (see page 5-35) unless the EXDRD bit in the SPCON1 register is set (see page 5-30). If the EXDRD bit is set, the received address bit is indicated by the AB bit in the SPRXD register (see page 5-37).
1	D7	**Data 7 Bits** 0 = Serial frames contain eight data bits. 1 = Serial frames contain seven data bits. The D7 bit should not be modified when either the TMODE bit or the RMODE bit is 1.
0	STP2	**Stop Bit Length** 0 = The serial port transmitter holds the TXD_U line High for one bit time to generate the stop bit. 1 = The serial port transmitter holds the TXD_U line High for two bit times to generate the stop bit. The STP2 bit should not be modified when either the TMODE bit or the RMODE bit is 1. The STP2 bit has no effect on received frames.

Programming Notes

Do not change bits FC, EVN, PEN, ABEN, D7, or STP2 while either the transmitter or receiver is enabled (that is, if the TMODE bit is 1 or the RMODE bit is 1).

The serial port (UART) is initially disabled after an external or watchdog timer reset. To enable the serial port, software must set the ITF4 bit field in the SYSCON register to 10b (see page 16-3).

Serial Port Control 1 (SPCON1) Offset 282h

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	*Res*												BRKVAL	EXDWR	EXDRD	XTRN
Software Read/Write	R												R/W	R/W	R/W	R/W
Hardware Set/Clear	—												—	—	—	—
Chip Reset Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Microcontroller:	
Am186CC	!
Am186CH	✔
Am186CU	!

! *= See bit definitions.*

Register Description

The SPCON1 register is the second of two control registers for the serial port. The SPCON0 register is the first control register (see page 5-28).

Bit Definitions

Bit **Name** **Function**

15–4 Res **Reserved**
For compatibility with future devices, always write this bit field with its chip reset default value.

3 BRKVAL **Break Value**
0 = The TXD_U line is held Low during breaks (when the BRK bit is 1 in the SPCON0 register, see page 5-30).

1 = The TXD_U line is held High during breaks.

Setting the BRKVAL bit to 1 allows the serial port transmit logic to time idle times on the TXD_U line.

2 EXDWR **Extended Write**
0 = The value of the AB bit in the SPCON0 register (see page 5-28) is written to the AB bit of the SPTXD register (see page 5-35).

1 = The AB bit in the upper byte of the SPTXD register can be written.

1 EXDRD **Extended Read**
0 = The upper byte of the SPRXD register (see page 5-36) always reads 0. When the EXDRD bit is 0, receive data DMA requests are inhibited while a receive status interrupting condition exists.

1 = The bit fields in the upper byte of the SPRXD register contain valid data. When the EXDRD bit is 1, receive data DMA requests are always generated when the RDR bit is set.

0 XTRN **External Clock**
0 = The serial port uses the internal processor clock to generate the serial clock.

1 = The serial port uses the external UCLK pin to generate the serial clock.

The XTRN bit should not be modified while either the TMODE bit or the RMODE bit is 1 (see page 5-29).

Am186CC and Am186CU Microcontrollers: *On the Am186CC and Am186CU microcontrollers, the UCLK pin is multiplexed with the USB sample clock input (USBSCI) and SOF output (USBSOF) functions, which are controlled by the ESOF_EN and SAM_CLK_SEL bits in the ISCTL register (see page 4-15 and page 4-16).*

Programming Notes

The XTRN bit should not be modified while either the transmitter or receiver is enabled (the TMODE bit is 1 or the RMODE bit is 1, see page 5-29).

Serial Port Status (SPSTAT) Offset 284h

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	*Res*						BRK	AB	RDR	THRE	FER	OER	PER	TEMT	IDLED	IDLE
Software Read/Write	R						R/W0	R/W0	R	R	R/W0	R/W0	R/W0	R	R/W0	R
Hardware Set/Clear	—						S	S/C	S/C	S/C	S	S	S	S/C	S	S/C
Chip Reset Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Microcontroller:
Am186CC ✔
Am186CH ✔
Am186CU ✔

Register Description

This register provides information about the current status of the serial port. Each bit in this register can be enabled as an interrupt source by setting the corresponding bit in the SPIMSK register (see page 5-33).

Bit Definitions

Bit	Name	Function
15–10	Res	**Reserved** For compatibility with future devices, always write this bit field with its chip reset default value.
9	BRK	**Break Detected** 0 = No break condition has been detected on the receive data line. 1 = The serial port has detected a break condition on the receive data line. The BRK bit must be cleared by software.
8	AB	**Address Bit Detected** 0 = The serial port has not received a frame with the address bit High. 1 = The serial port has received a frame with the address bit High (1). The AB bit is only updated when the ABEN bit in the SPCON0 register is 1 (see page 5-29). The AB bit must be cleared by software.
7	RDR	**Receive Data Ready** 0 = The RDATA bit field of the SPRXD register (see page 5-37) is undefined. 1 = The RDATA bit field of the SPRXD register contains valid data. The RDR bit is cleared by a read of the SPRXD register.
6	THRE	**Transmit Holding Register Empty** 0 = Writing to the SPTXD register (see page 5-35) results in the loss (failure to transmit) of the most recently written data (i.e., software must wait until the THRE bit is 1 before writing transmit data). 1 = The transmitter is capable of accepting more data for transmission. When the THRE bit is 1, the SPTXD register can be written without loss of data.
5	FER	**Framing Error Detected** 0 = No framing errors have been detected. The receiver only checks for framing errors during the first stop bit time, regardless of the setting of the STP2 bit in the SPCON0 register (see page 5-29). 1 = The serial port receiver has detected a framing error. A framing error is a Low value on the RXD_U line during the expected stop bit time. The FER bit must be cleared by software.

Bit	Name	Function
4	OER	**Overrun Error Detected** 0 = No frames have been lost due to overrun. 1 = The serial port receiver has lost data due to an overrun error between this frame and the previous frame. The OER bit must be cleared by software.
3	PER	**Parity Error Detected** 0 = The serial port receiver has not detected a parity error on a received frame. 1 = The serial port receiver has detected a parity error on a received frame. The PER bit must be cleared by software. Parity detection is configured via the EVN and PEN bits in the SPCON0 register (see page 5-29). The PER bit is always 0 if parity is not enabled.
2	TEMT	**Transmitter Empty** 0 = The transmitter contains data to be transmitted over the TXD_U line. 1 = The transmitter does not have any data that has not been transmitted. At this point, the transmitter can be disabled without loss of data.
1	IDLED	**Receive Line Idle Detected** 0 = The receiver has not been idle for 40 bit times, or no receive data is available. 1 = The receiver has detected a High bit on the RXD_U line for 40 bit times while receive data is available. This indicates that the receive message appears to be complete, but the receiver contains data that has not been read. The IDLED bit must be cleared by software.
0	IDLE	**Receive Line Idle** 0 = The receiver is currently receiving a data frame. 1 = The receiver is not currently in frame. This means that the RXD_U line has been High continuously since the completion of reception on the last frame.

Programming Notes

If the EXDRD bit is 1 in the SPCON1 register (see page 5-30), software can read the RDR, THRE, FER, OER, PER, BRK, and AB bits in the SPRXD register (see page 5-36) instead of the SPSTAT register. Software can ignore the corresponding bits in the SPSTAT register unless interrupts are enabled for the FER, OER, PER, or BRK bits in the SPIMSK register. These bits must be cleared in the SPSTAT register if they generate an interrupt.

Serial Port Interrupt Mask (SPIMSK) Offset 286h

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	*Res*						BRK	AB	RDR	THRE	FER	OER	PER	TEMT	IDLED	IDLE
Software Read/Write	R						R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Hardware Set/Clear	—						—	—	—	—	—	—	—	—	—	—
Chip Reset Default	0	0	0	0	0	0	1	0	1	1	1	1	1	0	0	0

Microcontroller:	
Am186CC	✔
Am186CH	✔
Am186CU	✔

Register Description

This register provides bits for individually enabling or disabling interrupts based on the condition of various status bits. The bits in the SPIMSK register have a one-to-one correspondence with the associated bits in the SPSTAT register (see page 5-31). The default settings of these bits provide compatibility with Am186ES and Am186ED microcontroller software.

Note that when the interrupt controller determines that a receive status bit has generated an interrupt condition and extended reads are not enabled, receive DMA requests are inhibited. If extended reads are enabled, DMA requests are enabled.

Bit Definitions

Bit	Name	Function
15–10	Res	**Reserved** For compatibility with future devices, always write this bit field with its chip reset default value.
9	BRK	**Break Interrupt Enable** 0 = The BRK bit in the SPSTAT register (see page 5-31) does not generate interrupt requests. 1 = When this BRK bit is 1 and the RSIE bit in the SPCON0 register is 1 (see page 5-28), the serial port generates interrupts on break detection (the BRK bit in the SPSTAT register is 1).
8	AB	**Address Bit Interrupt Enable** 0 = The AB bit in the SPSTAT register (see page 5-31) does not generate interrupt requests. 1 = When this AB bit is 1 and the RSIE bit in the SPCON0 register is 1 (see page 5-28), the serial port generates interrupts when a character is received with the address bit 1 (the AB bit in the SPSTAT register is 1).
7	RDR	**Receive Data Ready Interrupt Enable** 0 = The RDR bit in the SPSTAT register (see page 5-31) does not generate interrupt requests. 1 = When this RDR bit is 1 and the RXIE bit in the SPCON0 register is 1 (see page 5-28), the serial port generates interrupts when valid received data is available (the RDR bit in the status register is 1).
6	THRE	**Transmit Holding Register Empty Interrupt Enable** 0 = The THRE bit in the SPSTAT register (see page 5-31) does not generate interrupt requests. 1 = When this THRE bit is 1 and the TXIE bit in the SPCON0 register is 1 (see page 5-28), the serial port generates interrupts when the transmitter can accept more data (the THRE bit in the SPSTAT register is 1).
5	FER	**Framing Error Interrupt Enable** 0 = The FER bit in the SPSTAT register (see page 5-31) does not generate interrupt requests. 1 = When this FER bit is 1 and the RSIE bit in the SPCON0 register is 1 (see page 5-28), the serial port generates interrupts when a framing error is detected (the FER bit in the SPSTAT register is 1).

Bit	Name	Function
4	OER	**Overrun Error Interrupt Enable** 0 = The OER bit in the SPSTAT register (see page 5-32) does not generate interrupt requests. 1 = When this OER bit is 1 and the RSIE bit in the SPCON0 register is 1 (see page 5-28), the serial port generates interrupts when a receive overrun condition is detected (the OER bit in the SPSTAT register is 1).
3	PER	**Parity Error Interrupt Enable** 0 = The PER bit in the SPSTAT register (see page 5-32) does not generate interrupt requests. 1 = When this PER bit is 1 and the RSIE bit in the SPCON0 register is 1 (see page 5-28), the serial port generates interrupts when a parity error is detected (the PER bit in the SPSTAT register is 1).
2	TEMT	**Transmitter Empty Interrupt Enable** 0 = The TEMT bit in the SPSTAT register (see page 5-32) does not generate interrupt requests. 1 = When this TEMT bit is 1 and the TXIE bit in the SPCON0 register is 1 (see page 5-28), the serial port generates interrupts when the transmitter does not have any data to transmit over the line (the TEMT bit in the SPSTAT register is 1).
1	IDLED	**Receive Line Idle Detected Interrupt Enable** 0 = The IDLED bit in the SPSTAT register (see page 5-32) does not generate interrupt requests. 1 = When this IDLED bit is 1 and the RXIE bit in the SPCON0 register is 1 (see page 5-28), the serial port generates interrupts when the receive data line has been idle for 40 bit times and receive data is available (the IDLED bit in the SPSTAT register is 1).
0	IDLE	**Receive Line Idle Interrupt Enable** 0 = The IDLE bit in the SPSTAT register (see page 5-32) does not generate interrupt requests. 1 = When this IDLE bit is 1 and the RSIE bit in the SPCON0 register is 1 (see page 5-28), the serial port generates interrupts when the receive data line is idle (the IDLE bit in the SPSTAT register is 1.)

Programming Notes

There are two levels of interrupt enable. This register (SPIMSK) provides the second or lower level. The RSIE, RXIE, and TXIE bits in the control register provide the first or higher level of interrupt enable for the serial port. When the RSIE bit is 0, the IDLE, PER, OER, FER, AB, and BRK bit interrupt conditions are disabled, regardless of the settings of the associated bits in this register. When the RXIE bit is 0, the IDLED and RDR interrupt conditions are disabled regardless of the settings of the associated bits in this register. When the TXIE bit is 0, the TEMT and THRE bit interrupt conditions are disabled regardless of the settings of the associated bits in this register.

In addition to these requirements, software must configure the interrupt Channel 11 Control (CH11CON) register to select the interrupt channel’s internal source (UART) and enable the channel before UART interrupts can be generated. See Chapter 9, “Interrupt Controller Registers”.

Serial Port Transmit Data (SPTXD) Offset 288h

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	*Res*							AB	TDATA							
Software Read/Write	R							R/W	R/W							
Hardware Set/Clear	—							—	—							
Chip Reset Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Microcontroller:	
Am186CC	✓
Am186CH	✓
Am186CU	✓

Register Description

This register provides data to the serial port transmitter. When the THRE bit in the SPSTAT register is 1 (see page 5-31), the SPTXD register can be written with data to be transmitted over the serial port. If the SPTXD register is written when the THRE bit is not 1, data previously written to this register is lost. Data that has already been written to the transmit shift register (not visible to software) is not affected by a write to this register.

Bit Definitions

Bit	Name	Function
15–9	Res	**Reserved** For compatibility with future devices, always write this bit field with its chip reset default value.
8	AB	**Address Bit** The AB bit in this register contains the value to be transmitted for the address bit. This bit can only be written when the EXDWR bit is 1 in the SPCON1 register (see page 5-30). If the EXDWR bit is not 1, the value of the AB bit in the SPCON0 register (see page 5-28) is placed in the AB bit in this register (SPTXD) when the TDATA bit field is written.
7–0	TDATA	**Transmit Data** The TDATA bit field contains data to be transmitted through the serial port.

Programming Notes

Serial Port Receive Data (SPRXD) Offset 28Ah

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	RDR	THRE	FER	OER	PER	*Res*	BRK	AB	RDATA							
Software Read/Write	R	R	R	R	R	R	R	R	R							
Hardware Set/Clear	S/C	S/C	S/C	S/C	S/C	—	S/C	S/C	S/C							
Chip Reset Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Microcontroller:
Am186CC ✔
Am186CH ✔
Am186CU ✔

Register Description

When the RDR bit in the SPSTAT register is 1 (see page 5-31), the SPRXD register contains valid data received over the serial line. When extended reads are enabled, the value of the RDR bit is available in bit 15 of this register. The RDATA, FER, OER, PER, BRK, and AB bit fields only contain valid data when the RDR bit is 1. The THRE bit field contains valid data on all reads of this register when extended reads are enabled.

Reading the SPRXD register clears the RDR bit in the SPSTAT register if no further receive data is available.

Bit Definitions

Bit **Name** **Function**

15 RDR **Receive Data Ready**

0 = The RDATA bit field does not contain valid data, and the FER, OER, PER, BRK, and AB bit fields all read 0.

1 = The RDATA bit field contains valid data, and the FER, OER, PER, BRK, and AB bit fields contain valid status on that data.

The RDR bit is only valid if the EXDRD bit is 1 in the SPCON1 register (see page 5-30).

14 THRE **Transmit Holding Register Empty**

0 = Writing to the SPTXD register (see page 5-35) results in the loss (failure to transmit) of the most recently written data.

1 = The transmitter is capable of accepting more data for transmission. When the THRE bit is 1, the SPTXD register can be written without loss of data.

The THRE bit is only valid if the EXDRD bit is 1 in the SPCON1 register (see page 5-30).

13 FER **Framing Error**

0 = No framing error was detected on the received frame. The receiver only checks for framing errors during the first stop bit time, regardless of the setting of the STP2 bit in the SPCON0 register (see page 5-29).

1 = A framing error was detected on this frame. A framing error is a Low value on the RXD_U line during the expected stop bit time.

The FER bit is only valid if the EXDRD bit is 1 in the SPCON1 register (see page 5-30).

12 OER **Overrun Error**

0 = No overrun error occurred between this frame and the previous frame.

1 = An overrun error occurred between this frame and the previous frame.

The OER bit is only valid when the EXDRD bit is 1 in the SPCON1 register (see page 5-30).

11 PER **Parity Error**

0 = No parity error was detected on this frame.

1 = A parity error was detected on this frame.

Parity detection is configured via the EVN and PEN bits in the SPCON0 register (see page 5-29). The PER bit is always 0 if parity is not enabled.

The PER bit is only valid if the EXDRD bit is 1 in the SPCON1 register (see page 5-30).

Bit	Name	Function
10	Res	**Reserved** For compatibility with future devices, always write this bit field with its chip reset default value.
9	BRK	**Break** 0 = The received frame has not been detected as a break. 1 = The serial port detected a break condition on the received frame. The BRK bit is only valid if the EXDRD bit is 1 in the SPCON1 register (see page 5-30).
8	AB	**Address Bit** 0 = The address bit of the received frame was read as Low. 1 = The address bit of the received frame was read as High. The AB bit is only valid if the EXDRD bit is 1 in the SPCON1 register (see page 5-30).
7–0	RDATA	**Received Data** The RDATA bit field holds the value of the last character received over the RXD line. This bit field is valid only when the RDR bit in the status register is 1. Reading this register (SPRXD) clears the RDR bit in the SPSTAT register (see page 5-31) if no further receive data is available.

Programming Notes

If the EXDRD bit is 1 in the SPCON1 register (see page 5-30), software can read the RDR, THRE, FER, OER, PER, BRK, and AB bits in the SPRXD register instead of the SPSTAT register (see page 5-31). Software can ignore the corresponding bits in the SPSTAT register unless interrupts are enabled for the FER, OER, PER, or BRK bits in the SPIMSK register. These bits must be cleared in the SPSTAT register if they generate an interrupt.

Serial Port Receive Data Peek (SPRXDP) Offset 28Ch

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	RDR	THRE	FER	OER	PER	*Res*	BRK	AB	RDATA							
Software Read/Write	R	R	R	R	R	R	R	R	R							
Hardware Set/Clear	S/C	S/C	S/C	S/C	S/C	—	S/C	S/C	S/C							
Chip Reset Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Microcontroller:
Am186CC ✔
Am186CH ✔
Am186CU ✔

Register Description

This register provides a means of reading the SPRXD register (see page 5-36) without changing the serial port status. The bit definitions are otherwise the same.

Bit Definitions

Bit **Name** **Function**

15 RDR **Receive Data Ready (Same as SPRXD)**

0 = The RDATA bit field does not contain valid data, and the FER, OER, PER, BRK, and AB bit fields all read 0.

1 = The RDATA bit field contains valid data, and the FER, OER, PER, BRK, and AB bit fields contain valid status on that data.

The RDR bit is only valid if the EXDRD bit is 1 in the SPCON1 register (see page 5-30).

14 THRE **Transmit Holding Register Empty (Same as SPRXD)**

0 = Writing to the SPTXD register (see page 5-35) results in the loss (failure to transmit) of the most recently written data.

1 = The transmitter is capable of accepting more data for transmission. When the THRE bit is 1, the SPTXD register can be written without loss of data.

The THRE bit is only valid if the EXDRD bit is 1 in the SPCON1 register (see page 5-30).

13 FER **Framing Error (Same as SPRXD)**

0 = No framing error was detected on the received frame. The receiver only checks for framing errors during the first stop bit time, regardless of the setting of the STP2 bit in the SPCON0 register (see page 5-29).

1 = A framing error was detected on this frame. A framing error is a Low value on the RXD_U line during the expected stop bit time.

The FER bit is only valid if the EXDRD bit is 1 in the SPCON1 register (see page 5-30).

12 OER **Overrun Error (Same as SPRXD)**

0 = No overrun error occurred between this frame and the previous frame.

1 = An overrun error occurred between this frame and the previous frame.

The OER bit is only valid when the EXDRD bit is 1 in the SPCON1 register (see page 5-30).

11 PER **Parity Error (Same as SPRXD)**

0 = No parity error was detected on this frame.

1 = A parity error was detected on this frame.

Parity detection is configured via the EVN and PEN bits in the SPCON0 register (see page 5-29). The PER bit is always 0 if parity is not enabled.

The PER bit is only valid if the EXDRD bit is 1 in the SPCON1 register (see page 5-30).

10 Res **Reserved**

For compatibility with future devices, always write this bit field with its chip reset default value.

Bit	Name	Function
9	BRK	**Break (Same as SPRXD)** 0 = The received frame has not been detected as a break. 1 = The serial port detected a break condition on the received frame. The BRK bit is only valid if the EXDRD bit is 1 in the SPCON1 register (see page 5-30).
8	AB	**Address Bit (Same as SPRXD)** 0 = The address bit of the received frame was read as Low. 1 = The address bit of the received frame was read as High. The AB bit is only valid if the EXDRD bit is 1 in the SPCON1 register (see page 5-30).
7–0	RDATA	**Received Data** The RDATA bit field holds the value of the last character received over the RXD line. This bit field is valid only when the RDR bit in the status register is 1.

Programming Notes

If the EXDRD bit is 1 in the SPCON1 register (see page 5-30), software can read the RDR, THRE, FER, OER, PER, BRK, and AB bits in the SPRXD register instead of the SPSTAT register (see page 5-31). Software can ignore the corresponding bits in the SPSTAT register unless interrupts are enabled for the FER, OER, PER, or BRK bits in the SPIMSK register. These bits must be cleared in the SPSTAT register if they generate an interrupt.

Serial Port Baud Rate Divisor (SPBDV) Offset 28Eh

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	BAUDDIV															
Software Read/Write	R/W															
Hardware Set/Clear	—															
Chip Reset Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Microcontroller:

Am186CC	✓
Am186CH	✓
Am186CU	✓

Register Description

This register specifies a clock divisor for the generation of the serial clock. The SPBDV register should not be modified while either the RMODE bit or the TMODE bit in the SPCON0 register is 1 (see page 5-29). In a startup situation, a non-zero value must be written into this register (SPBDV) before setting the RMODE bit or the TMODE bit to 1. If the BAUDDIV bit field is cleared to 0, data is neither received nor transmitted.

The baud divisor register determines the baud rate using the following formula:

BAUDDIV = (UART frequency / (16 · baud rate))

Where UART frequency is either the processor frequency or the UCLK frequency if external clocking is enabled.

Bit Definitions

Bit	Name	Function
15–0	BAUDDIV	**Baud Divisor** The BAUDDIV bit field specifies the divisor used to generate the serial clock.

Programming Notes

Software can set baud rates up to 1/16th of the UART frequency (by setting the BAUDDIV bit field to 1) if the connecting device's baud rate formula is known. But it may be necessary to use special UART frequencies and non-integer baud rates.

Table 5-3 shows the value of the BAUDDIV bit field, in both decimal and hexadecimal, for common baud rates and clock frequencies. The actual baud rate varies from the nominal rate by the amount shown in the % Error column.

Table 5-3 Baud Rate Table for UART (Same as High-Speed UART)

	Serial Port Clock Frequency (Processor Frequency or UCLK Frequency)											
	24 MHz		25 MHz		40 MHz		44.2 MHz		48 MHz		50 MHz	
Baud Rate	Divisor	% Error	Divisor	% Error	Divisor	% Error	Divisor	% Error	Divisor	% Error	Divisor	% Error
300	5000d 1388h	0	5208d 1458h	0	8333d 208Dh	0	9208d 23F8h	0	10000d 2710h	0	10417d 28B1h	0
600	2500d 09C4h	0	2604d 0A2Ch	0	4167d 1047h	0	4604d 11FCh	0	5000d 1388h	0	5208d 1458h	0
1200	1250d 04E2h	0	1302d 0516h	0	2083d 0823h	0	2302d 08FEh	0	2500d 09C4h	0	2604d 0A2Ch	0
2400	625d 0271h	0	651d 028Bh	0	1042d 0412h	0	1151d 047Fh	0	1250d 04E2h	0	1302d 0516h	0
9600	156d 9Ch	0.2	163d A3h	-0.1	260d 0104h	0.2	288d 0120h	-0.1	313d 0139h	-0.2	326d 0146h	-0.1
19200	78d 4Eh	0.2	81d 51h	0.5	130d 82h	0.2	144d 90h	-0.1	156d 9Ch	0.2	163d A3h	-0.1
38400	39d 27h	0.2	41d 29h	-0.8	65d 41h	0.2	72d 48h	-0.1	78d 4Eh	0.2	81d 51h	0.5
57600	26d 1Ah	0.2	27d 1Bh	0.5	43d 2Bh	0.9	48d 30h	-0.1	52d 34h	0.2	54d 36h	0.5
115200	13d 0Dh	0.2	14d 0Eh	-3.2	22d 16h	-1.4	24d 18h	-0.1	26d 1Ah	0.2	27d 1Bh	0.5
230400	7d 07h	-7.5	7d 07h	-3.2	11d 0Bh	-1.4	12d 0Ch	-0.1	13d 0Dh	0.2	14d 0Eh	-3.2
460800	3d 03h	7.8	3d 03h	11.5	5d 05h	7.8	6d 06h	-0.1	7d 07h	-7.5	7d 07h	-3.2

CHAPTER

6 GCI REGISTERS (Am186CC Microcontroller Only)

6.1 OVERVIEW

This chapter describes the General Circuit Interface (GCI) registers on the Am186CC microcontroller. *The Am186CH and Am186CU microcontrollers do not support GCI.*

The Am186CC microcontroller supports the terminal version of GCI, which is a standard serial bus for interconnecting telecommunications integrated circuits for linecard, NT1, and ISDN terminals.

For more information about using GCI, see the *Am186CC/CH/CU Microcontrollers User's Manual*, order #21914.

Table 6-1 lists the GCI registers in offset order, with the corresponding description's page number.

Table 6-1 GCI Register Map

Register Name	Mnemonic	Offset	Page Number
GCI Peripheral Control	GPCON	2A0h	page 6-2
GCI Interrupt Status	GISTAT	2A2h	page 6-4
GCI Interrupt Mask	GIMSK	2A4h	page 6-6
GCI TIC Bus Address	GTIC	2A6h	page 6-8
GCI Intercommunication Transmit Data	GICTD	2A8h	page 6-9
GCI Intercommunication Receive Data	GICRD	2AAh	page 6-10
GCI Intercommunication Receive Data Peek	GICRDP	2ACh	page 6-11
GCI Command/Indicate Transmit Data 0	GCITD0	2AEh	page 6-12
GCI Command/Indicate Receive Data 0	GCIRD0	2B0h	page 6-13
GCI Command/Indicate Receive Data 0 Peek	GCIRD0P	2B2h	page 6-14
GCI Command/Indicate Transmit Data 1	GCITD1	2B4h	page 6-15
GCI Command/Indicate Receive Data 1	GCIRD1	2B6h	page 6-16
GCI Command/Indicate Receive Data 1 Peek	GCIRD1P	2B8h	page 6-17
GCI Monitor Transmit Data	GMTD	2BAh	page 6-18
GCI Monitor Receive Data	GMRD	2BCh	page 6-19
GCI Monitor Receive Data Peek	GMRDP	2BEh	page 6-20

GCI Peripheral Control (GPCON) Offset 2A0h

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	*Res*							PCMFSC	MCARV	MARQ	MCHEN	MCHSEL	MEOMRQ	ICSEL	GCIACT	BRDIS
Software Read/Write	R							R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Hardware Set/Clear	—							—	C	—	—	—	C	—	—	—
Chip Reset Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Microcontroller:
Am186CC ☑
Am186CH □
Am186CU □

Register Description

This register is used to control the General Circuit Interface.

Bit Definitions

Bit **Name** **Function**

15–9 Res **Reserved**
For compatibility with future devices, always write this bit field with its chip reset default value.

8 PCMFSC **Pulse Code Modulation (PCM) B Channel Frame Sync Select**
0 = The PCM frame sync is placed one clock period prior to GCI B1 channel data.

1 = The PCM frame sync is placed one clock period prior to GCI B2 channel data. This allows PCM devices that follow a one-clock-prior-to-8-bit-data protocol to multiplex directly onto the GCI bus.

7 MCARV **Monitor Channel Address Received Valid**
0 = The MCARV bit is automatically cleared to 0 before every monitor first byte is received.

1 = Software sets the MCARV bit in response to the first monitor-data-available interrupt, which indicates to the local monitor receiver that software has determined the first byte address is valid.

6 MARQ **Monitor Channel Abort Request**
0 = An abort is not being requested (the default state). Software must clear the MARQ bit.

1 = When the MARQ bit is set by software, the remote transmitter is forced to abort.

Whenever the MARQ bit is set to 1, the local receiver is placed in its reset state. There is no indication that the remote transmitter has aborted.

Software can set and then immediately clear the MARQ bit to force an abort. After toggling this bit, software should read the GMRD register immediately (which clears the MRDA status bit) to ensure that the next MRDA bit interrupt is set for the first byte of the non-aborted transmission and not for the aborted frame.

5 MCHEN **Monitor Channel Enable**
0 = Both monitor channels are disabled. Whenever the monitor channel is disabled, the monitor channel is placed in its reset state, and the GMTD and GMRD registers are updated to their default states: GMTD = 00FFh, and GMRD = 0000h.

1 = The monitor channel selected by the MCHSEL bit is enabled.

To disable and then re-enable the monitor channel during a transmission, software should clear the MCHEN bit to 0, wait at least two frames, and then set the MCHEN bit to 1.

4 MCHSEL **Monitor Channel Select**
0 = Monitor channel 0 is used (first GCI subframe).

1 = Monitor channel 1 is used (second GCI subframe).

During the unused channel's transmit time, the data pin (DU or DD) is always placed in a high-impedance state. This is equivalent to transmitting a 1 for every bit in the unused channel (including its MR/MX handshake bits).

Bit	Name	Function
3	MEOMRQ	**Monitor Channel End-of-Message (EOM) Request** 0 = The MEOMRQ bit is cleared by hardware when the EOM is sent. 1 = When set, the MEOMRQ bit forces the monitor channel transmitter to send an EOM when all data written into the Monitor Transmit Data register has been transmitted. This tells the receiving device that the message is complete.
2	ICSEL	**Interchip Communication (IC) Channel Select** 0 = IC channel 1 is used (fifth octet after the frame sync). 1 = IC channel 2 is used (sixth octet after frame sync). During the unused channel's transmit time, the data pin (DU or DD) is always placed in a high-impedance state. This is equivalent to transmitting a 1 for every bit in the unused channel.
1	GCIACT	**GCI Activation** 0 = GCI activation is not requested by this device. 1 = Software sets the GCIACT bit to initiate an activation request to the clock master. The GCIACT bit controls the GCI Timing Request mechanism. When this bit is set to 1, the DU pin is driven Low and held Low until software clears the bit. When a device on the GCI bus drives DU Low, the clock master starts the data clock (on the microcontroller's GCI_DCL_x input signal), which generates the microcontroller's DCLST interrupt. If the microcontroller requested the activation, its DCLST interrupt handler clears the GCIACT bit and initiates a transfer. The DCL clock remains active until the upstream device stops the clock.
0	BRDIS	**Bus Reversal Disable** 0 = The bus reverses (i.e., the DD and DU pin functions are reversed during the second GCI subframe). This is the default. 1 = The bus reversal function is disabled.

Programming Notes

In order to enable the GCI interface, software must set the MODE bit field to 10b in the TSA Channel A Configuration (TSACON) register (see page 7-3). This is necessary regardless of whether TSA channel A is being used.

GCI Interrupt Status (GISTAT) Offset 2A2h

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	*Res*						IC	DCLST	CHGCI1	CHGCI0	MRAD	MCD	MTARD	MEOMRD	MXBA	MRDA
Software Read/Write							R	R/W0	R	R	R/W0	R/W0	R/W0	R/W0	R	R
Hardware Set/Clear	—						S/C	S	S/C	S/C	S	S	S	S	S/C	S/C
Chip Reset Default	0	0	0	0	0	0	0	?	0	0	0	0	0	0	1	0

Microcontroller:
Am186CC ☑
Am186CH □
Am186CU □

Register Description

This register provides GCI status information.

Bit Definitions

Bit	Name	Function
15–10	Res	**Reserved** For compatibility with future devices, always write this bit field with its chip reset default value.
9	IC	**IC Buffer Empty/Available Interrupt** 0 = The IC bit is cleared when the GICRD register is read (see page 6-10) or the GICTD register is written (see page 6-9). 1 = Hardware sets the IC bit to indicate that the current IC channel (selected by the ICSEL bit in the GPCON register, see page 6-3) has either an empty transmit buffer or available data in its receive buffer.
8	DCLST	**DCL Start** 0 = DCL has not started or software has cleared the DCLST bit since the last time DCL went High. 1 = Hardware sets the DCLST bit to indicate that DCL has started (goes High) from a deactivated GCI bus state. After the DCL clock has started, this bit is set on every Low-to-High DCL transition.
7	CHGCI1	**Change in C/I1 Channel Status** 0 = The CHGCI1 bit is cleared when the GCIRD1 register is read (see page 6-16). 1 = Hardware sets the CHGCI1 bit to indicate that the contents on the receive side of C/I channel 1 have changed since the GCIRD1 register was last read.
6	CHGCI0	**Change in C/I0 Channel Status** 0 = The CHGCI0 bit is cleared when the GCIRD0 register is read (see page 6-13). 1 = Hardware sets the CHGCI0 bit to indicate that the contents on the receive side of the C/I channel 0 have changed since the GCIRD0 register was last read.
5	MRAD	**Monitor Channel Receive Abort Detected** 0 = The event has not occurred, or software has cleared the MRAD bit. 1 = A transmitter abort has been implied by out of sequence MX bits or MX bit transmission errors.
4	MCD	**Monitor Channel Collision Detected** 0 = The event has not occurred, or software has cleared the MCD bit. 1 = Hardware sets this bit if a collision occurs on any transmitted byte of monitor data.
3	MTARD	**Monitor Channel Transmit Abort Request Received** 0 = The event has not occurred, or software has cleared the MTARD bit. 1 = Hardware sets the MTARD bit to indicate that an abort request has been received on the monitor channel. This indicates that the receiver on the other end of the monitor channel has failed to receive the transmitted data correctly and wants the current transmission discontinued and the data transmission repeated via software.

Bit	Name	Function
2	MEOMRD	**Monitor Channel End-of-Message Indication Received** 0 = The event has not occurred, or software has cleared the MEOMRD bit. 1 = Hardware sets the MEOMRD bit to indicate that an EOM has been received on the monitor channel. This indicates that the message currently being received has concluded.
1	MXBA	**Monitor Channel Transmit Buffer Available** 0 = The MXBA bit is cleared when the GMTD register is written (see page 6-18), unless the write occurs during a monitor channel transmit abort request. 1 = Hardware sets the MXBA bit to indicate that a new byte of data can be loaded into the GMTD register. This bit is set to 1 during a monitor channel transmit abort request.
0	MRDA	**Monitor Channel Receive Data Available Interrupt** 0 = The MRDA bit is cleared when the GMRD register is read (see page 6-19). 1 = Hardware sets the MRDA bit to indicate that a byte of data has been received on the monitor channel and is available in the GMRD register.

Programming Notes

These individual interrupt status bits are masked via the GCI Interrupt Mask (GIMSK) register. The state of the respective interrupt mask bit does not affect the setting of the bits in the GISTAT register. That is, the status bits continue to be updated regardless of the mask setting; the mask setting only prevents the interrupt to the microcontroller. When the monitor channel is disabled (via bit 5 of the GPCON register), bits 5, 4, 3, 2, and 0 of the GISTAT register are cleared, and bit 1 is set.

In addition to setting bits in the GIMSK register, software must configure the interrupt Channel 8 Control (CH8CON) register to select the interrupt channel’s internal source (GCI) and enable the interrupt channel before any GCI interrupts can be generated. See Chapter 9, “Interrupt Controller Registers”.

GCI Interrupt Mask (GIMSK) Offset 2A4h

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	*Res*						IC	DCLST	CHGCI1	CHGCIO	MRAD	MCD	MTARD	MEOMRD	MXBA	MRDA
Software Read/Write	R						R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Hardware Set/Clear	—						—	—	—	—	—	—	—	—	—	—
Chip Reset Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Microcontroller:
Am186CC ☑
Am186CH □
Am186CU □

Register Description

This register provides an individual interrupt mask bit corresponding with each of the GCI status conditions in the GISTAT register (see page 6-4). When a mask bit is set, the corresponding source is allowed to cause an interrupt. Clearing the bit masks the individual source and prevents the source from causing an interrupt.

Bit Definitions

Bit	Name	Function
15–10	Res	**Reserved** For compatibility with future devices, always write this bit field with its chip reset default value.
9	IC	**IC Interrupt Enable** 0 = The IC bit interrupt is masked. 1 = The IC bit interrupt is enabled.
8	DCLST	**DCL Start Interrupt Enable** 0 = The DCLST bit interrupt is masked. 1 = The DCLST bit interrupt is enabled.
7	CHGCI1	**Change in C/I1 Channel Status Interrupt Enable** 0 = The CHGCI1 bit interrupt is masked. 1 = The CHGCI1 bit interrupt is enabled.
6	CHGCI0	**Change in C/I0 Channel Status Interrupt Enable** 0 = The CHGCI0 bit interrupt is masked. 1 = The CHGCI0 bit interrupt is enabled.
5	MRAD	**Monitor Channel Receive Abort Detected Interrupt Enable** 0 = The MRAD bit interrupt is masked. 1 = The MRAD bit interrupt is enabled.
4	MCD	**Monitor Channel Collision Detected Interrupt Enable** 0 = The MCD bit interrupt is masked. 1 = The MCD bit interrupt is enabled.
3	MTARD	**Monitor Channel Abort Request Received Interrupt Enable** 0 = The MTARD bit interrupt is masked. 1 = The MTARD bit interrupt is enabled.
2	MEOMRD	**Monitor Channel End-of-Message Indication Received Interrupt Enable** 0 = The MEOMRD bit interrupt is masked. 1 = The MEOMRD bit interrupt is enabled.

Bit	Name	Function
1	MXBA	**Monitor Channel Transmit Buffer Available Interrupt Enable** 0 = The MXBA bit interrupt is masked. 1 = The MXBA bit interrupt is enabled.
0	MRDA	**Monitor Channel Receive Data Available Interrupt Enable** 0 = The MRDA bit interrupt is masked. 1 = The MRDA bit interrupt is enabled.

Programming Notes

In addition to setting bits in this register (GIMSK), software must configure the interrupt Channel 8 Control (CH8CON) register to select the interrupt channel's internal source (GCI) and enable the interrupt channel before any GCI interrupts can be generated. See Chapter 9, "Interrupt Controller Registers".

GCI TIC Bus Address (GTIC) Offset 2A6h

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	*Res*								TICEN	ECHOEN	*Res*			TICAD		
Software Read/Write	R								R/W	R/W	R			R/W		
Hardware Set/Clear	—								—	—	—			—		
Chip Reset Default	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1

Microcontroller:
Am186CC ☑
Am186CH □
Am186CU □

Register Description

This register contains the Terminal Interchip Communication (TIC) bus control and address fields.

Bit Definitions

Bit	Name	Function
15–8	Res	**Reserved** For compatibility with future devices, always write this bit field with its chip reset default value.
7	TICEN	**TIC Bus Enable** 0 = TIC bus access is disabled. 1 = TIC bus access is enabled.
6	ECHOEN	**TIC Echo Enable** 0 = D channel echo compares are disabled. 1 = D channel echo compares are enabled.
5–3	Res	**Reserved** For compatibility with future devices, always write this bit field with its chip reset default value.
2–0	TICAD	**TIC Bus Address** The TICAD bit field defines the individual TIC address specific to the device trying to obtain access on the GCI bus. This address is used to access the C/I0 and D channels on the GCI bus.

Programming Notes

GCI Intercommunication Transmit Data (GICTD) Offset 2A8h

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	*Res*								IC12T							
Software Read/Write	R								R/W							
Hardware Set/Clear	—								—							
Chip Reset Default	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

Microcontroller:
Am186CC ☑
Am186CH □
Am186CU □

Register Description

This is the write register for IC channel transmit data.

Bit Definitions

Bit	Name	Function
15–8	Res	**Reserved** For compatibility with future devices, always write this bit field with its chip reset default value.
7–0	IC12T	**Intercommunication Transmit Channel 1/2** The IC12T bit field contains user-defined transmission data for either GCI IC Channel 1 (IC1) or GCI IC Channel 2 (IC2). The ICSEL bit in the GPCON register (see page 6-3) determines the specific IC channel these bits reference. Data is written into these bits by the user in response to an IC Buffer Empty/Available interrupt.

Programming Notes

The GICTD register must be serviced every 125 µs to guarantee noncorrupted data.

GCI Intercommunication Receive Data (GICRD) Offset 2AAh

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	*Res*								IC12R							
Software Read/Write	R								R							
Hardware Set/Clear	—								S/C							
Chip Reset Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Microcontroller:
Am186CC ☑
Am186CH □
Am186CU □

Register Description

This is the read register for IC channel receive data.

Bit Definitions

Bit	Name	Function
15–8	Res	**Reserved** For compatibility with future devices, always write this bit field with its chip reset default value.
7–0	IC12R	**Intercommunication Receive Channel 1/2** The IC12R bit field contains received data for either GCI IC Channel 1 (IC1) or GCI IC Channel 2 (IC2). The ICSEL bit in the GPCON register (see page 6-3) determines the specific IC channel this bit field affects. Data is written into these bits by hardware and is indicated valid by an IC Buffer Empty/Available interrupt. Software can then read the data.

Programming Notes

In every GCI frame, the GICRD register has its data replaced by the new data being received. Nothing inhibits the hardware from writing over old received data, so the GCIRD register should be serviced before new data arrives (every 125 µs).

GCI Intercommunication Receive Data Peek (GICRDP) Offset 2ACh

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	*Res*								IC12P							
Software Read/Write	R								R							
Hardware Set/Clear	—								S/C							
Chip Reset Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Microcontroller:
Am186CC ☑
Am186CH □
Am186CU □

Register Description

This register provides a means of reading the GICRD register (see page 6-10) without affecting the status of the GCI controller.

Bit Definitions

Bit	Name	Function
15–8	Res	**Reserved** For compatibility with future devices, always write this bit field with its chip reset default value.
7–0	IC12P	**Intercommunication Receive Channel 1/2 Peek** The IC12P bit field contains a copy of received data for either GCI IC Channel 1 (IC1) or GCI IC Channel 2 (IC2). The ICSEL bit in the GPCON register (see page 6-3) determines the specific IC channel this bit field affects.

Programming Notes

This register is used primarily for debugging purposes.

GCI Command/Indicate Transmit Data 0 (GCITD0) Offset 2AEh

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	*Res*								BAR	*Res*			CI0T			
Software Read/Write	R								R/W	R			R/W			
Hardware Set/Clear	—								—	—			—			
Chip Reset Default	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1

Microcontroller:
Am186CC ☑
Am186CH □
Am186CU □

Register Description

This is the transmit data register for Command/Indicate Channel 0.

Bit Definitions

Bit	Name	Function
15–8	Res	**Reserved** For compatibility with future devices, always write this bit field with its chip reset default value.
7	BAR	**Bus Access Request** When software sets the BAR bit to 1, the GCI controller attempts to gain access to the C/I0 channel if the TIC bus is enabled.
6–4	Res	**Reserved** For compatibility with future devices, always write this bit field with its chip reset default value.
3–0	CI0T	**Command/Indicate Transmit Channel 0** The CI0T bit field contains user defined transmission data for the GCI C/I0 Channel. Software can write data into the CI0T bit field at any time. The data is transmitted continuously during each subsequent frame until software writes a new value to the CI0T bit field.

Programming Notes

GCI Command/Indicate Receive Data 0 (GCIRD0) Offset 2B0h

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	*Res*												CI0R			
Software Read/Write	R												R			
Hardware Set/Clear	—												S/C			
Chip Reset Default	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1

Microcontroller:

Am186CC ☑
Am186CH □
Am186CU □

Register Description

This is the receive data register for Command/Indicate Channel 0.

Bit Definitions

Bit	Name	Function
15–4	Res	**Reserved** For compatibility with future devices, always write this bit field with its chip reset default value.
3–0	CI0R	**Command/Indicate Receive Channel 0** The CI0R bit field contains received GCI C/I0 channel data that has passed the GCI valid data integrity check. Data is written into the GCIRD0 register by hardware and contains data that has been valid for two frames. A Change in C/I0 Channel Status interrupt is generated to indicate that the contents of the GCIRD0 register have changed since it was last read.

Programming Notes

GCI Command/Indicate Receive Data 0 Peek (GCIRD0P) Offset 2B2h

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	*Res*												CI0P			
Software Read/Write	R												R			
Hardware Set/Clear	—												S/C			
Chip Reset Default	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1

Microcontroller:
Am186CC ☑
Am186CH □
Am186CU □

Register Description

This register provides a means of reading the GCIRD0 register (see page 6-13) without affecting the status of the GCI controller.

Bit Definitions

Bit	Name	Function
15–4	Res	**Reserved** For compatibility with future devices, always write this bit field with its chip reset default value.
3–0	CI0P	**Command/Indicate Receive Channel 0 Peek** The CI0P bit field contains a copy of received GCI C/I0 channel data that has passed the GCI valid data integrity check.

Programming Notes

This register is used primarily for debugging purposes.

GCI Command/Indicate Transmit Data 1 (GCITD1) Offset 2B4h

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	*Res*										CI1T					
Software Read/Write											R/W					
Hardware Set/Clear	—										—					
Chip Reset Default	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1

Microcontroller:
Am186CC ☑
Am186CH □
Am186CU □

Register Description

This is the transmit data register for Command/Indicate Channel 1.

Bit Definitions

Bit	Name	Function
15–6	Res	**Reserved** For compatibility with future devices, always write this bit field with its chip reset default value.
5–0	CI1T	**Command/Indicate Transmit Channel 1** The CI1T bit field contains user defined transmission data for the GCI C/I1 Channel. Software can write data into these bits at any time. The data is transmitted continuously during each subsequent frame until software writes a new value to the CI1T bit field.

Programming Notes

GCI Command/Indicate Receive Data 1 (GCIRD1) Offset 2B6h

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	*Res*										CI1R					
Software Read/Write	R										R					
Hardware Set/Clear	—										S/C					
Chip Reset Default	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1

Microcontroller:
Am186CC ☑
Am186CH □
Am186CU □

Register Description

This is the receive data register for Command/Indicate Channel 1.

Bit Definitions

Bit	Name	Function
15–6	Res	**Reserved** For compatibility with future devices, always write this bit field with its chip reset default value.
5–0	CI1R	**Command/Indicate Receive Channel 1** The CI1R bit field contains received GCI C/I1 Channel data that has passed the GCI valid data integrity check. Data is written into the GCIRD1 register by hardware and contains data that has been valid for two frames. A Change in C/I1 Channel Status interrupt is generated to indicate that the contents of the GCIRD1 register have changed since it was last read.

Programming Notes

GCI Command/Indicate Receive Data 1 Peek (GCIRD1P) Offset 2B8h

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	*Res*										CI1P					
Software Read/Write	R										R					
Hardware Set/Clear	—										S/C					
Chip Reset Default	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1

Microcontroller:

Am186CC ☑
Am186CH □
Am186CU □

Register Description

This register provides a means of reading the GCIRD1 register (see page 6-16) without affecting the status of the GCI controller.

Bit Definitions

Bit	Name	Function
15–6	Res	**Reserved** For compatibility with future devices, always write this bit field with its chip reset default value.
5–0	CI1P	**Command/Indicate Receive Channel 1 Peek** The CI1P bit field contains a copy of received GCI C/I1 Channel data that has passed the GCI valid data integrity check.

Programming Notes

This register is used primarily for debugging purposes.

GCI Monitor Transmit Data (GMTD) Offset 2BAh

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	*Res*								MON01T							
Software Read/Write	R								R/W							
Hardware Set/Clear	—								S/C							
Chip Reset Default	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

Microcontroller:
Am186CC ☑
Am186CH □
Am186CU □

Register Description

This is the transmit data register for monitor channels 0 and 1.

Bit Definitions

Bit	Name	Function
15–8	Res	**Reserved** For compatibility with future devices, always write this bit field with its chip reset default value.
7–0	MON01T	**Monitor 0/1 Transmit Data** The MON01T bit field contains user-defined transmission data for either the GCI monitor channel 0 (Mon0) or the GCI monitor channel 1 (Mon1). The MCHSEL bit in the GPCON register (see page 6-2) determines the specific monitor channel this bit field affects. Data is written into these bits by the user in response to a Monitor Transmit Buffer Available interrupt. The data is then transmitted to the receiver on the other side of the GCI bus. During an unused channel’s transmit time, the data pin (DU or DD) is always placed in a high-impedance state. This is equivalent to transmitting a 1 for every bit in the unused channel.

Programming Notes

GCI Monitor Receive Data (GMRD) Offset 2BCh

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	*Res*								MON01R							
Software Read/Write									R							
Hardware Set/Clear	—								C							
Chip Reset Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Microcontroller:

Am186CC	☑
Am186CH	□
Am186CU	□

Register Description

This is the receive data register for monitor channels 0 and 1.

Bit Definitions

Bit	Name	Function
15–8	Res	**Reserved** For compatibility with future devices, always write this bit field with its chip reset default value.
7–0	MON01R	**Monitor 0/1 Receive Data** The MON01R bit field contains received data for either the GCI monitor channel 0 (Mon0) or the GCI monitor channel 1 (Mon1). The MCSEL bit in the GPCON register (see page 6-2) determines the specific monitor channel this bit field affects. Data is written into the MON01R bit field by hardware as it is received over the monitor channel. A Monitor Data Available interrupt is generated when the GMRD register is loaded. Reading the GMRD register indicates to the hardware that the next byte can be placed in this bit field. Hardware clears the MON01R bit field if the selected channel is disabled by the MCHEN bit in the GPCON register (see page 6-2).

Programming Notes

GCI Monitor Receive Data Peek (GMRDP) Offset 2BEh

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	*Res*								MON01P							
Software Read/Write	Res								R							
Hardware Set/Clear	—								C							
Chip Reset Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Microcontroller:

- Am186CC ☑
- Am186CH □
- Am186CU □

Register Description

This register provides a means of reading the GMRD register (see page 6-19) without affecting the status of the GCI controller.

Bit Definitions

Bit	Name	Function
15–8	Res	**Reserved** For compatibility with future devices, always write this bit field with its chip reset default value.
7–0	MON01P	**Monitor 0/1 Receive Data Peek** The MON01P bit field contains a copy of received data for either the GCI monitor channel 0 (Mon0) or the GCI monitor channel 1 (Mon1). The MCSEL bit in the GPCON register determines the specific monitor channel this bit field affects.

Programming Notes

This register is used primarily for debugging purposes.

CHAPTER

7 TSA REGISTERS (Am186CC and Am186CH Microcontrollers Only)

7.1 OVERVIEW

This chapter describes the Time Slot Assigner (TSA) registers on the Am186CC and Am186CH microcontrollers. *The Am186CU USB microcontroller does not support TSAs.*

Each TSA allows the transmission and reception of data between an individual High-Level Data Link Control (HDLC) channel and an external time-division multiplexed (TDM) data stream (see Chapter 2, “HDLC Registers (Am186CC and Am186CH Microcontrollers Only)”). The external data stream can be either PCM Highway, raw DCE, or (on the Am186CC microcontroller only) General Circuit Interface (GCI).

For more information about using TSAs, see the *Am186CC/CH/CU Microcontrollers User’s Manual*, order #21914.

The TSA register sets are identical for each of the channels. This chapter describes registers with identical functions only once. The unique register names and offsets that apply are listed at the top of each register page. Table 7-1 lists the TSA registers in offset order, with the corresponding description’s page number.

Table 7-1 TSA Register Map

Register Name	Mnemonic	Offset	Page Number
TSA Channel A Configuration	TSACON	2C0h	page 7-2
TSA Channel A Bit Start Position	TSASTART	2C2h	page 7-5
TSA Channel A Bit Stop Position	TSASTOP	2C4h	page 7-7
TSA Channel B Configuration	TSBCON	2C8h	page 7-2
TSA Channel B Bit Start Position	TSBSTART	2CAh	page 7-5
TSA Channel B Bit Stop Position	TSBSTOP	2CCh	page 7-7
TSA Channel C Configuration	TSCCON	2D0h	page 7-2
TSA Channel C Bit Start Position	TSCSTART	2D2h	page 7-5
TSA Channel C Bit Stop Position	TSCSTOP	2D4h	page 7-7
TSA Channel D Configuration	TSDCON	2D8h	page 7-2
TSA Channel D Bit Start Position	TSDSTART	2DAh	page 7-5
TSA Channel D Bit Stop Position	TSDSTOP	2DCh	page 7-7

TSA Channel A Configuration (TSACON) Offset 2C0h
TSA Channel B Configuration (TSBCON) Offset 2C8h
TSA Channel C Configuration (TSCCON) Offset 2D0h
TSA Channel D Configuration (TSDCON) Offset 2D8h

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	EN	*Res*					MODE		*Res*		FSCP	DRVLVL	*Res*	ESADJ		
Software Read/Write	R/W	R					R/W		R		R/W	R/W	R	R/W		
Hardware Set/Clear	—	—					—		—		—	—	—	—		
Chip Reset Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Microcontroller:	Channel: A	B	C	D
Am186CC	✓	✓	✓	✓
Am186CH	!	!		
Am186CU				

! = *See bit definitions.*

Register Description

This register enables and sets operating modes for the TSA channel.

Bit Definitions

Bit **Name** **Function**

15 EN **TSA Channel Enable**

0 = The channel is disabled.

1 = The channel is enabled.

The EN bit enables the TSA channel. When the channel is disabled, bit fields BPSTART in the TSxSTART register (see page 7-5), BPSTOP in the TSxSTOP register (see page 7-7), and DRVLVL, FSCP, and ESADJ of this register (TSxCON) have no meaning. If the EN bit is cleared and then set, no data is transmitted to or received from the HDLC channel until the next full frame has been reacquired.

If the MODE bit field is 00 (raw DCE mode), the EN bit has no effect. Other MODE bit settings have no meaning when the channel is disabled (EN = 0).

14–10 Res **Reserved**

For compatibility with future devices, always write this bit field with its chip reset default value.

Bit	Name	Function
9–8	MODE	**Channel Mode** The MODE bit field selects the channel configuration: 00 = Raw DCE 01 = Raw PCM highway 10 = For channel A: GCI mode For channels B–D: GCI mode multiplexed on channel A 11 = For channel A: Reserved For channels B–D: PCM highway mode multiplexed on channel A ***Am186CH Microcontroller:*** *The value 10b is not supported on the Am186CH HDLC microcontroller. The value 11b is supported for multiplexing channel B only. HDLC and TSA channels C and D are not supported in the Am186CH HDLC microcontroller.* Changing of the MODE bits should be done only when the channel is disabled (i.e., the EN bit is 0). After any channel mode switch, the HDLC channel becomes stable within no more than six clock periods, measured by the new mode's clock. During this stabilization period, inputs to the HDLC are ignored. In raw DCE mode, the channel's time slot assigner is bypassed. Programming bits DRVLVL, FSCP, and ESADJ of this register (TSxCON) and bit fields BPSTART in the TSxSTART register (see page 7-5) and BPSTOP in the TSxSTOP register (see page 7-7) has no effect on the channel in raw DCE mode. Several bits in the SYSCON register (see page 16-2) affect channel mode operation. If an external PCM codec is configured on channel C by setting SYSCON register bit EXSYNC, HDLC channel C can be programmed for GCI time-slotting on channel A with the MODE bit field set to 10b. If the UART is selected by setting SYSCON register bit field ITF4 to 10b, use of HDLC channel D is limited to GCI or PCM time-slotting on channel A via programming the MODE bit field to 10b or 11b respectively. In this case, no HDLC full flow control on channel D is possible because the pins are used for the UART. Programming the MODE bit field to 00b or 01b with the UART selected is illegal. If the High-Speed UART is selected by setting SYSCON register bit field ITF4 to 01b, HDLC channel D loses full flow control when programmed with the MODE bit field set to 00b or 01b. However, if the High-Speed UART is selected with the ITF4 bit set to 00b, HDLC channel D has full pin flow control with the MODE bit field set to 00b or 01b. If both the High-Speed UART and the UART are selected by setting SYSCON register bit field ITF4 to 10b, use of HDLC channel D is limited to GCI or PCM time-slotting on channel A (i.e., the MODE bit is 10b or 11b).
7–6	Res	**Reserved** For compatibility with future devices, always write this bit field with its chip reset default value.
5	FSCP	**Channel Frame Sync Pulse Polarity Detect** The FSCP bit selects the polarity for detecting the channel's frame sync pulse. This bit has no effect in raw DCE mode. 0 = Frame Sync pulse detect is asserted High. 1 = Frame Sync pulse detect is asserted Low.
4	DRVLVL	**Channel Adjust Bit Drive Level** The DRVLVL bit sets the drive level of bits transmitted after the bit stop position if the ESADJ bit field is set to any value other than 000b. The DRVLVL and ESADJ bits have no effect in raw DCE mode. 0 = Drive ESADJ bits to Low. 1 = Drive EASDJ bits to High.

Bit	Name	Function
3	Res	**Reserved** For compatibility with future devices, always write this bit field with its chip reset default value.
2–0	ESADJ	**Channel End Slot Bit Adjust** The ESADJ bit field specifies the additional number of bits to transmit starting from the bit immediately after the bit stop position programmed in the TSxSTOP register’s BPSTOP bit field (see page 7-7). The drive level for these bits is specified by the DRVLVL bit. The ESADJ and DRVLVL bit fields have no effect in raw DCE mode. 000 = No bits are driven after the bit stop position. 001 = 1 bit is driven after the bit stop position. 010 = 2 bits are driven after the bit stop position. 011 = 3 bits are driven after the bit stop position. 100 = 4 bits are driven after the bit stop position. 101 = 5 bits are driven after the bit stop position. 110 = 6 bits are driven after the bit stop position. 111 = 7 bits are driven after the bit stop position. Note that if the ESADJ bit field is 000b, then the bit position programmed in the BPSTOP bit field is the last bit position transmitted for the frame.

Programming Notes

TSA Channel A Bit Start Position (TSASTART) Offset 2C2h
TSA Channel B Bit Start Position (TSBSTART) Offset 2CAh
TSA Channel C Bit Start Position (TSCSTART) Offset 2D2h
TSA Channel D Bit Start Position (TSDSTART) Offset 2DAh

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	*Res*				BPSTART											
Software Read/Write	R				R/W											
Hardware Set/Clear	—				—											
Chip Reset Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Microcontroller:	Channel: A	B	C	D
Am186CC	✓	✓	✓	✓
Am186CH	✓	✓		
Am186CU				

Register Description

This register defines the time-slot bit start position for the receive and transmit data frame.

Bit Definitions

Bit	Name	Function
15–12	Res	**Reserved** For compatibility with future devices, always write this bit field with its chip reset default value.
11–0	BPSTART	**Channel Bit Start Position**

The BPSTART bits define the time-slot bit start position for receive and transmit data frame. Together with the BPSTOP bit field (see page 7-7), they configure the contiguous number of bits for the data frame. The BPSTART bit field value can exceed the value of the BPSTOP bit field. In this case, frame boundaries are crossed. The BPSTART and BPSTOP bit fields have no effect in raw DCE mode.

000000000000 = Start frame 0 bit time position after frame sync

000000000001 = Start frame1 bit time position after frame sync

000000000010 = Start frame 2 bit time position after frame sync

. . . . .

111111111110 = Start frame 4094 bit time position after frame sync

111111111111 = Start frame 4095 bit time position after frame sync

The total number of bits transmitted or received per every frame sync pulse can be calculated from the formula [(BPSTOP + 1) – BPSTART + ESADJ]. If the BPSTART bit field value exceeds the BPSTOP bit field value, the formula [MaxFrameSize – BPSTART + (BPSTOP +1) + ESADJ] can be used. MaxFrameSize is the frame's maximum bit count and is system specific. If the BPSTART bit field value equals the BPSTOP bit field value, then only one bit is transferred. To transfer the entire frame, set the BPSTART bit field to 0 and the BPSTOP bit field to (MaxFrameSize –1).

(For example, if the BPSTART bit field value is 000001001b and the BPSTOP bit field value is 000110000b, a contiguous 40 bits, equivalent to five 8-bit contiguous time-slots, is transmitted and received for the channel. The data is transmitted/received nine bit times after frame sync detect. Furthermore, if the ESADJ bit field value is 100 and the DRVLVL bit field value is 1, bit positions 49–52 are driven High. In this case, a total of 40 + 4 = 44 bits per every 8 kHz frame are transmitted for the channel. Only 40 bits are received (i.e., bit positions 9–48).

If the BPSTART bit field value exceeds the maximum bit-position possible for the time division multiplexed (TDM) frame, no data is received or transmitted for that channel. For example, if 256 bits exist for every TDM frame, programming the BPSTART bit field to 300d results in no data transfer.

Programming Notes

TSA Channel A Bit Stop Position (TSASTOP) Offset 2C4h
TSA Channel B Bit Stop Position (TSBSTOP) Offset 2CCh
TSA Channel C Bit Stop Position (TSCSTOP) Offset 2D4h
TSA Channel D Bit Stop Position (TSDSTOP) Offset 2DCh

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	*Res*				BPSTOP											
Software Read/Write	R				R/W											
Hardware Set/Clear	—				—											
Chip Reset Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Microcontroller:	Channel: A	B	C	D
Am186CC	✓	✓	✓	✓
Am186CH	✓	✓		
Am186CU				

Register Description

This register defines the bit stop position for the receive and transmit data frame.

Bit Definitions

Bit	Name	Function
15–12	Res	**Reserved** For compatibility with future devices, always write this bit field with its chip reset default value.
11–0	BPSTOP	**Channel Bit Stop Position**

The BPSTOP bits define the bit stop position for the receive and transmit data frame. The data to be transferred is inclusive of this bit. Together with the BPSTART bit field, the BPSTOP bit field configures the contiguous number of bits for the data frame. The BPSTART bit field value can exceed the value of the BPSTOP bit field. In this case, frame boundaries are crossed. The BPSTART and BPSTOP bits have no effect in raw DCE mode.

000000000000 = Stop frame 0 bit time position after frame sync

000000000001 = Stop frame1 bit time position after frame sync

000000000010 = Stop frame 2 bit time position after frame sync

. . . . .

111111111110 = Stop frame 4094 bit time position after frame sync

111111111111 = Stop frame 4095 bit time position after frame sync

Refer to the BPSTART bit description on page 7-5 for a detailed description and example.

If the BPSTOP bit field value exceeds the maximum bit-position possible for the TDM frame, data is continuously received or transmitted for that channel because the stop point is never reached. For example, if 256 bits exist for every TDM frame, programming the BPSTOP bit field to 313d results in continuous data transfer.

Programming Notes

CHAPTER

8 SYNCHRONOUS SERIAL INTERFACE (SSI) REGISTERS

8.1 OVERVIEW

This chapter describes the Synchronous Serial Interface (SSI) registers on the Am186CC/CH/CU microcontrollers.

The SSI provides a half-duplex, bidirectional communications interface between the microcontroller and other integrated circuits.

For more information about using the SSI, see the *Am186CC/CH/CU Microcontrollers User's Manual*, order #21914.

Table 8-1 lists the SSI registers in offset order, with the corresponding description's page number.

Table 8-1 SSI Register Map

Register Name	Mnemonic	Offset	Page Number
SSI Mode/Status	SSSTAT	2F0h	page 8-2
SSI Control	SSCON	2F2h	page 8-3
SSI Transmit 1	SSTXD1	2F4h	page 8-5
SSI Transmit 0	SSTXD0	2F6h	page 8-5
SSI Receive Data	SSRXD	2F8h	page 8-6

SSI Mode/Status (SSSTAT) Offset 2F0h

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	ENHCTL	*Res*												RE/TE	DR/DT	PB
Software Read/Write	R/W	R												R	R	R
Hardware Set/Clear	—	—												S/C	S/C	S/C
Chip Reset Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Microcontroller:	
Am186CC	✔
Am186CH	✔
Am186CU	✔

Register Description

This register defines Synchronous Serial Interface (SSI) control mode and status bits

Bit Definitions

Bit	Name	Function
15	ENHCTL	**Enhanced Control** 0 = Am186EM Microcontroller Control mode (/2 and /4 clock divisors only). Only bits 0, 1, and 4 of the SSCON register (see page 8-4 and page 8-4) are operational. This bit defaults to 0. 1 = Enhanced Control mode. All bits of the SSCON register are operational.
14–3	Res	**Reserved** For compatibility with future devices, always write this bit field with its chip reset default value.
2	RE/TE	**Receive/Transmit Error Detect** 0 = The event has not occurred, or the bit was cleared. The RE/TE bit is cleared when bits DE1 and DE0 in the SSCON register are both cleared to 0. 1 = A receive or transmit was attempted while the SSI was busy with a previous transaction.
1	DR/DT	**Data Receive/Transmit Complete** 0 = The event has not occurred, or the bit was cleared. The DR/DT bit is cleared by reading the SSSTAT or SSRXD register, or by writing to the SSTXD1 or SSTXD0 register, or by clearing bits DE1 and DE0 in the SSCON register. 1 = The SSI has gone non-busy following a receive or transmit operation (i.e., the software can write the next byte out, or read the byte that was just received). The DR/DT bit is read-only.
0	PB	**Port Busy** 0 = The port is not busy. 1 = A receive or transmit operation is in progress. The PB bit is read-only.

Programming Notes

SSI Control (SSCON) Offset 2F2h

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	*Res*				CLKP	DENP	*Res*	MSBF	*Res*	CLKEXP			*Res*		DE1	DE0
Software Read/Write	R				R/W	R/W	R	R/W	R	R/W			R		R/W	R/W
Hardware Set/Clear	—				—	—	—	—	—	—			—		—	—
Chip Reset Default	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0

Microcontroller:

Am186CC ✔

Am186CH ✔

Am186CU ✔

Register Description

This register controls SSI features.

Bit Definitions

Bit	Name	Function
15–12	Res	**Reserved** For compatibility with future devices, always write this bit field with its chip reset default value.
11	CLKP	**Active Clock Polarity** 0 = The SCLK signal is normally High, and is Low when active. 1 = The SCLK signal is normally Low, and is High when active. Before using the CLKP bit, software must set the ENHCTL bit in the SSSTAT register (see page 8-2). The transmitting device should always drive new data on the inactive-to-active transition, and the receiving device should always latch data on the active-to-inactive transition. The state of the CLKP bit should be changed only when the SSI system is completely quiescent (i.e., when the PB bit in the SSSTAT register is clear and all devices' enables are inactive).
10	DENP	**Device Enable Polarity** 0 = The DE0 bit of the SSCON register is inverted and sent to the SDEN pin output (active Low). 1 = The SDEN pin output is active High. Before using the DENP bit, software must set the ENHCTL bit in the SSSTAT register (see page 8-2). The DENP bit is only used to invert the polarity of the SDEN bit to the pin. The DE0 bit of the SSCON register is always active High. The default is 1 (active High).
9	Res	**Reserved** For compatibility with future devices, always write this bit field with its chip reset default value.
8	MSBF	**Most Significant Bit First** 0 = The LSB is transmitted and received first. 1 = The MSB is transmitted and received first. Before using the MSBF bit, software must set the ENHCTL bit in the SSSTAT register (see page 8-2). The state of the MSBF bit should not be changed when the PB bit in the SSSTAT register is set.
7	Res	**Reserved** For compatibility with future devices, always write this bit field with its chip reset default value.

Bit	Name	Function
6–4	CLKEXP	**Clock Exponent** The CLKEXP bit field defines how the bit clock is derived from the CPU clock: 000 = CPUCLK/2 001 = CPUCLK/4 010 = CPUCLK/8 011 = CPUCLK/16 100 = CPUCLK/32 101 = CPUCLK/64 110 = CPUCLK/128 111 = CPUCLK/256 Before selecting clock divisors other than /2 and /4, software must set the ENHCTL bit in the SSSTAT register (see page 8-2).
3–2	Res	**Reserved** For compatibility with future devices, always write this bit field with its chip reset default value.
1	DE1	**Device Enable 1 (SSI Enable Only)** When either bit DE1 or bit DE0 is set, the SSI state machine is enabled for operation. The DE1 bit can be used to enable and disable the SSI state machine when using software-controlled PIO pins instead of the SDEN signal to enable additional external devices for SSI communication.
0	DE0	**Device Enable 0 (SSI and SDEN Enable)** When either bit DE1 or bit DE0 is set, the SSI state machine is enabled for operation. The DE0 bit also controls the state of the SDEN signal, which is used to enable an external device for SSI communication. The SDEN signal polarity is controlled by the DENP bit in this register (SSCON).

Programming Notes

Bits 4, 1 and 0 of the SSI control register are backward-compatible with the Am186EM and Am186ER microcontrollers SSI Control register. Bits 11, 10, 8, 6, and 5 are changed. See the ENHCTL bit description on page 8-2.

SSI Transmit 1 (SSTXD1) Offset 2F4h

SSI Transmit 0 (SSTXD0) Offset 2F6h

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	*Res*								TXDATA							
Software Read/Write	R								R/W							
Hardware Set/Clear	—								—							
Chip Reset Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Microcontroller:	
Am186CC	✓
Am186CH	✓
Am186CU	✓

Register Description

Writing to a transmit register initiates a transmit transaction on the SSI bus if the bus is not currently busy. Writing to either SSTXDx register clears the DR/DT bit in the SSSTAT register (see page 8-2). If the PB bit in the SSSTAT register is set, writing to either SSTXDx register sets the RE/TE bit in the SSSTAT register. If both the DE1 and DE0 bits in the SSCON register are 0 (see page 8-4), a write to either SSTXDx register does not initiate a transmit transaction.

Bit Definitions

Bit	Name	Function
15–8	Res	**Reserved** For compatibility with future devices, always write this bit field with its chip reset default value.
7–0	TXDATA	**Transmit Data** The TXDATA 8-bit field is written with the data to be sent across the SSI interface. The TXDATA bit field always reads back the last value that was written. When the SSI system is in a state where it is ready to accept data for transmission (all non-busy states, and busy states that are caused solely by external ready or internal interbyte delay), writing to the transmit register immediately loads the shift register with the data, initiating a transmit transaction. When the SSI system is not ready to accept data, a write to the SSTXDx register does not load the shift register or otherwise adversely affect previously initiated transactions. But the SSSTAT register RE/TE bit is set to show the current transaction was rejected. Even in this case, the current (failed) transaction's data is available for read-back from this register.

Programming Notes

SSI Receive Data (SSRXD) Offset 2F8h

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	*Res*								RXDATA							
Software Read/Write	R								R							
Hardware Set/Clear	—								—							
Chip Reset Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Microcontroller:	
Am186CC	✓
Am186CH	✓
Am186CU	✓

Register Description

Reading this register serves two purposes; it returns data which was previously received from the external SSI device, and it initiates a new SSI receive, clocking new data in from the SDATA pin.

These purposes are served simultaneously; previously received data is returned, and a new read is initiated. Note that the first read of a series of reads never returns valid data. Instead, the first read simply starts the read process so that a subsequent read can start valid data.

Software should poll the SSSTAT register (see page 8-2) between reads in order to wait for valid data to be available. Reading the SSRXD register clears the DR/DT bit in the SSSTAT register. Also, if the PB bit in the status register is set, reading the SSRXD register sets RE/TE in the SSSTAT register.

Each read of valid data from the SSRXD register initiates a new SSI receive transaction. This is a potential problem when reading the last byte sent because a new transaction might not be desired. To avoid starting a new transaction unintentionally, clear both the DE1 and DE0 bits in the SSCON register (see page 8-4) after the last byte of the data has been received, then read the SSRXD register. A receive transaction is not initiated if the SSI port is not enabled.

Bit Definitions

Bit	Name	Function
15–8	Res	**Reserved** For compatibility with future devices, always write this bit field with its chip reset default value.
7–0	RXDATA	**Receive Data** The RXDATA 8-bit field contains data shifted in from the SSI interface.

Programming Notes

CHAPTER

9 INTERRUPT CONTROLLER REGISTERS

9.1 OVERVIEW

This chapter describes the Interrupt Controller registers on the Am186CC/CH/CU microcontrollers.

The Am186CC microcontroller provides a total of 15 interrupt channels to support a total of 36 maskable interrupt sources. There are 19 internal sources (timers, asynchronous ports, HDLCs, GCI, DMA, and USB) and 17 external interrupt sources. Of the external sources, nine are dedicated pins (INT8–0) and eight are interrupt-capable programmable I/O pins.

The Am186CH and Am186CU microcontrollers also provide 17 external interrupt sources, but fewer internal interrupts are used, as compared to the Am186CC microcontroller. The Am186CH HDLC microcontroller provides only 15 internal sources (omitting the GCI, USB, and two HDLC sources), and the Am186CU USB microcontroller provides only 14 internal sources (omitting the GCI and four HDLC sources).

Each interrupt channel has a control register, an in-service bit, a request bit, and a mask bit. The control register for each channel contains a 3-bit programmable priority field, a mask bit, and if necessary, a Level-Triggered Mode (LTM) bit and a source selection bit. The Interrupt Mask register contains a copy of the mask bit from each control register. The In-Service and Interrupt Request registers contain the in-service bit and request bit for each interrupt channel. The Interrupt Polarity and PIO Polarity registers are provided to control the polarity (active High or active Low) of external interrupt sources. Because channel 14 is shared by interrupt sources, additional registers are used to control the channel 14 interrupt sources; these are the Shared Mask and Shared Request registers.

An interrupt channel’s in-service bit is set when the interrupt is generated. The End-of-Interrupt (EOI) register is used by software to clear the current interrupt’s in-service bit either specifically by type or non-specifically according to priority.

The Interrupt Status register provides status for the timer and general-purpose DMA interrupt sources. The DMA Halt register is used to control DMA activity during interrupts.

If interrupts are not desired, the Poll and Poll Status registers can be used to poll the interrupt channels while interrupts are disabled.

For more information about using interrupts, see the *Am186CC/CH/CU Microcontrollers User’s Manual*, order #21914.

Table 9-1 lists the Interrupt Controller registers in offset order, with the corresponding description's page number.

Table 9-1 Interrupt Register Map

Register Name	Mnemonic	Offset	Page Number
Channel 0 Control	CH0CON	300h	page 9-4
Channel 1 Control	CH1CON	302h	page 9-6
Channel 2 Control	CH2CON	304h	page 9-8
Channel 3 Control	CH3CON	306h	page 9-10
Channel 4 Control	CH4CON	308h	page 9-12
Channel 5 Control	CH5CON	30Ah	page 9-14
Channel 6 Control	CH6CON	30Ch	page 9-16
Channel 7 Control	CH7CON	30Eh	page 9-18
Channel 8 Control	CH8CON	310h	page 9-20
Channel 9 Control	CH9CON	312h	page 9-22
Channel 10 Control	CH10CON	314h	page 9-24
Channel 11 Control	CH11CON	316h	page 9-26
Channel 12 Control	CH12CON	318h	page 9-28
Channel 13 Control	CH13CON	31Ah	page 9-30
Channel 14 Control	CH14CON	31Ch	page 9-32
End-of-Interrupt	EOI	320h	page 9-33
Poll	POLL	322h	page 9-35
Poll Status	POLLST	324h	page 9-36
Interrupt Mask	IMASK	326h	page 9-37
Priority Mask	PRIMSK	328h	page 9-39
In-Service	INSERV	32Ah	page 9-40
Interrupt Request	REQST	32Ch	page 9-41
Interrupt Status	INTSTS	32Eh	page 9-42
DMA Halt	DMAHLT	330h	page 9-43
Shared Request	SHREQ	332h	page 9-44
Shared Mask	SHMASK	334h	page 9-45
Interrupt Polarity	INTPOL	336h	page 9-46
PIO Polarity	PIOPOL	338h	page 9-47

Table 9-2 Interrupt Channel Sources

Interrupt Channel	Default Source	Optional Source
0	Timer 0,Timer 1, and Timer 2	—
1	INT0	—
2	INT1	USB[1]
3	INT2	High-Speed UART
4	HDLC Channel A[2] and SmartDMA Channel Pair 0[2]	—
5	HDLC Channel B[2] and SmartDMA Channel Pair 1[2]	—
6	HDLC Channel C[3] and SmartDMA Channel Pair 2[1]	—
7	HDLC Channel D[3] and SmartDMA Channel Pair 3[1]	—
8	INT3	GCI[3]
9	INT4	General-Purpose DMA 0 and 1
10	INT5	General-Purpose DMA 2 and 3
11	INT6	UART
12	INT7	PWD[4]
13	INT8	PWD[4]
14	Shared Request[5]	—

Notes:

1. Am186CC and Am186CU microcontrollers only.

2. Am186CC and Am186CH microcontrollers only.

3. Am186CC microcontroller only.

4. The PWD source is selected by setting the PWD bit in the SYSCON register. See page 16-2.

5. The Shared Request source is controlled by the SHREQ and SHMASK registers. The following sources can be enabled to use the Shared Request channel: PIO pins 35, 34, 33, 30, 29, 27, 15, and 5; and INT pins 7–1.

Channel 0 Control (CH0CON) Offset 300h

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	*Res*												MSK	PR		
Software Read/Write	R												R/W	R/W		
Hardware Set/Clear	—												—	—		
Chip Reset Default	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1

Register Description

Microcontroller:

Am186CC	✔
Am186CH	✔
Am186CU	✔

This is the Channel 0 Interrupt Control register. Channel 0 serves the timer sources. It behaves as a level-triggered channel.

Bit Definitions

Bit	Name	Function
15–4	Res	**Reserved** For compatibility with future devices, always write this bit field with its chip reset default value.
3	MSK	**Mask** 0 = Enables interrupt. 1 = Masks (disables) interrupt.
2–0	PR	**Priority Level** Software can use the PR bit field to program the interrupt channel's priority level relative to other maskable interrupts. If two or more interrupts are requested at the same time, the interrupt priority determines which interrupt is serviced first. If the PR bit fields are the same, the interrupts' overall priority is used (see Table 9-3 on page 9-34). Nonmaskable interrupts always have priority over maskable interrupts. 000 = Priority 0 (Highest) 001 = Priority 1 010 = Priority 2 011 = Priority 3 100 = Priority 4 101 = Priority 5 110 = Priority 6 111 = Priority 7 (Lowest) The PRIMSK register (see page 9-39) can be used to mask interrupts according to their PR bit field settings.

Programming Notes

In addition to clearing the MSK bit in the CH0CON register, software must also set the INT bit in the appropriate Timer Mode/Control (TxCON) register to enable a timer as an interrupt source (see page 10-2, page 10-8, and page 10-14).

All maskable interrupts are disabled if the IF bit is cleared in the Processor Status Flags (FLAGS) register. The IF bit is cleared automatically when a maskable interrupt is generated, and it is restored to its previous state (set or cleared) when the IRET processor instruction is executed. Software can also set or clear the IF bit by using the STI and CLI processor instructions, respectively. The FLAGS register is an internal processor register, not a peripheral control block register, so it is described in the *Am186 and Am188 Family Instruction Set Manual*, order #21267.

Channel 1 Control (CH1CON) Offset 302h

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	*Res*											LTM	MSK	PR		
Software Read/Write	R											R/W	R/W	R/W		
Hardware Set/Clear	—											—	—	—		
Chip Reset Default	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1

Microcontroller:
Am186CC ☑
Am186CH ☑
Am186CU ☑

Register Description

This is the Channel 1 Interrupt Control register. Channel 1 serves the INT0 pin source. The polarity, active High versus active Low, of the external interrupt on this channel can be set via a bit in the INTPOL register (see page 9-46).

Bit Definitions

Bit	Name	Function
15–5	Res	**Reserved** For compatibility with future devices, always write this bit field with its chip reset default value.
4	LTM	**Level-Triggered Mode** 0 = Edge-triggered 1 = Level-triggered
3	MSK	**Mask** 0 = Enables interrupt. 1 = Masks (disables) interrupt.
2–0	PR	**Priority Level** Software can use the PR bit field to program the interrupt channel's priority level relative to other maskable interrupts. If two or more interrupts are requested at the same time, the interrupt priority determines which interrupt is serviced first. If the PR bit fields are the same, the interrupts' overall priority is used (see Table 9-3 on page 9-34). Nonmaskable interrupts always have priority over maskable interrupts. 000 = Priority 0 (Highest) 001 = Priority 1 010 = Priority 2 011 = Priority 3 100 = Priority 4 101 = Priority 5 110 = Priority 6 111 = Priority 7 (Lowest) The PRIMSK register (see page 9-39) can be used to mask interrupts according to their PR bit field settings.

Programming Notes

All maskable interrupts are disabled if the IF bit is cleared in the Processor Status Flags (FLAGS) register. The IF bit is cleared automatically when a maskable interrupt is generated, and it is restored to its previous state (set or cleared) when the IRET processor instruction is executed. Software can also set or clear the IF bit by using the STI and CLI processor instructions, respectively. The FLAGS register is an internal processor register, not a peripheral control block register, so it is described in the *Am186 and Am188 Family Instruction Set Manual*, order #21267.

Channel 2 Control (CH2CON) Offset 304h

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	*Res*										SRC	LTM	MSK	PR		
Software Read/Write	R										R/W	R/W	R/W	R/W		
Hardware Set/Clear	—										—	—	—	—		
Chip Reset Default	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1

Microcontroller:	
Am186CC	✔
Am186CH	!
Am186CU	✔

! *= See bit definitions.*

Register Description

This is the Channel 2 Interrupt Control register. Channel 2 serves the USB (internal) or the INT1 pin (external) sources. The polarity, active High versus active Low, of the external interrupt on this channel can be set via a bit in the INTPOL register (see page 9-46).

Bit Definitions

Bit	Name	Function
15–6	Res	**Reserved** For compatibility with future devices, always write this bit field with its chip reset default value.
5	SRC	**Source** 0 = External source 1 = Internal source ***Am186CH Microcontroller:*** *The internal source (USB) is not supported on the Am186CH HDLC microcontroller.*
4	LTM	**Level-Triggered Mode** 0 = Edge-triggered 1 = Level-triggered; held at 1 if the SRC bit is 1
3	MSK	**Mask** 0 = Enables interrupt. 1 = Masks (disables) interrupt.
2–0	PR	**Priority Level** Software can use the PR bit field to program the interrupt channel's priority level relative to other maskable interrupts. If two or more interrupts are requested at the same time, the interrupt priority determines which interrupt is serviced first. If the PR bit fields are the same, the interrupts' overall priority is used (see Table 9-3 on page 9-34). Nonmaskable interrupts always have priority over maskable interrupts. 000 = Priority 0 (Highest) 001 = Priority 1 010 = Priority 2 011 = Priority 3 100 = Priority 4 101 = Priority 5 110 = Priority 6 111 = Priority 7 (Lowest) The PRIMSK register (see page 9-39) can be used to mask interrupts according to their PR bit field settings.

Programming Notes

In addition to setting the SRC bit and clearing the MSK bit in the CH1CON register, software must also set the appropriate bit in the USB Interrupt Mask 1 (UIMASK1) or USB Interrupt Mask 2 (UIMASK2) register to enable a USB event as an interrupt source (see page 4-5 and page 4-9).

All maskable interrupts are disabled if the IF bit is cleared in the Processor Status Flags (FLAGS) register. The IF bit is cleared automatically when a maskable interrupt is generated, and it is restored to its previous state (set or cleared) when the IRET processor instruction is executed. Software can also set or clear the IF bit by using the STI and CLI processor instructions, respectively. The FLAGS register is an internal processor register, not a peripheral control block register, so it is described in the *Am186 and Am188 Family Instruction Set Manual*, order #21267.

Channel 3 Control (CH3CON) Offset 306h

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	*Res*										SRC	LTM	MSK	PR		
Software Read/Write	R										R/W	R/W	R/W	R/W		
Hardware Set/Clear	—										—	—	—	—		
Chip Reset Default	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1

Microcontroller:

Am186CC ☑
Am186CH ☑
Am186CU ☑

Register Description

This is the Channel 3 Interrupt Control register. Channel 3 serves the High-Speed UART (internal) or the INT2 pin (external) sources. The polarity, active High versus active Low, of the external interrupt on this channel can be set via a bit in the INTPOL register (see page 9-46).

Bit Definitions

Bit	Name	Function
15–6	Res	**Reserved** For compatibility with future devices, always write this bit field with its chip reset default value.
5	SRC	**Source** 0 = External source 1 = Internal source
4	LTM	**Level-Triggered Mode** 0 = Edge-triggered 1 = Level-triggered; held at 1 if the SRC bit is 1
3	MSK	**Mask** 0 = Enables interrupt. 1 = Masks (disables) interrupt.
2–0	PR	**Priority Level** Software can use the PR bit field to program the interrupt channel's priority level relative to other maskable interrupts. If two or more interrupts are requested at the same time, the interrupt priority determines which interrupt is serviced first. If the PR bit fields are the same, the interrupts' overall priority is used (see Table 9-3 on page 9-34). Nonmaskable interrupts always have priority over maskable interrupts. 000 = Priority 0 (Highest) 001 = Priority 1 010 = Priority 2 011 = Priority 3 100 = Priority 4 101 = Priority 5 110 = Priority 6 111 = Priority 7 (Lowest) The PRIMSK register (see page 9-39) can be used to mask interrupts according to their PR bit field settings.

Programming Notes

In addition to setting the SRC bit and clearing the MSK bit in the CH3CON register, software must also set the appropriate bits in the High-Speed Serial Port Control 0 (HSPCON0) and High-Speed Serial Port Interrupt Mask (HSPIMSK) registers to enable a High-Speed UART event as an interrupt source (see page 5-3 and page 5-10).

All maskable interrupts are disabled if the IF bit is cleared in the Processor Status Flags (FLAGS) register. The IF bit is cleared automatically when a maskable interrupt is generated, and it is restored to its previous state (set or cleared) when the IRET processor instruction is executed. Software can also set or clear the IF bit by using the STI and CLI processor instructions, respectively. The FLAGS register is an internal processor register, not a peripheral control block register, so it is described in the *Am186 and Am188 Family Instruction Set Manual*, order #21267.

Channel 4 Control (CH4CON) Offset 308h

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	*Res*												MSK	PR		
Software Read/Write	R												R/W	R/W		
Hardware Set/Clear	—												—	—		
Chip Reset Default	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1

Microcontroller:

Am186CC	✓
Am186CH	✓
Am186CU	□

Register Description

This is the Channel 4 Interrupt Control register. Channel 4 serves the HDLC channel A and SmartDMA pair 0 sources. Interrupt channel 4 behaves as a level-triggered channel.

Bit Definitions

Bit	Name	Function
15–4	Res	**Reserved** For compatibility with future devices, always write this bit field with its chip reset default value.
3	MSK	**Mask** 0 = Enables interrupt. 1 = Masks (disables) interrupt.
2–0	PR	**Priority Level** Software can use the PR bit field to program the interrupt channel's priority level relative to other maskable interrupts. If two or more interrupts are requested at the same time, the interrupt priority determines which interrupt is serviced first. If the PR bit fields are the same, the interrupts' overall priority is used (see Table 9-3 on page 9-34). Nonmaskable interrupts always have priority over maskable interrupts. 000 = Priority 0 (Highest) 001 = Priority 1 010 = Priority 2 011 = Priority 3 100 = Priority 4 101 = Priority 5 110 = Priority 6 111 = Priority 7 (Lowest) The PRIMSK register (see page 9-39) can be used to mask interrupts according to their PR bit field settings.

Programming Notes

In addition to clearing the MSK bit in the CH4CON register, software must also set the appropriate bit in the source peripheral's register set to enable an event as an interrupt source. For HDLC channel A, interrupts are controlled by the HDLC Channel A Interrupt Mask 0 (HAIMSK0) and HDLC Channel A Interrupt Mask 1 (HAIMSK1) registers (see page 2-18 and page 2-22). For SmartDMA channel pair 0, interrupts are controlled by the SmartDMA Channel Pair 0 Control (SD0CON) register (see page 3-14).

All maskable interrupts are disabled if the IF bit is cleared in the Processor Status Flags (FLAGS) register. The IF bit is cleared automatically when a maskable interrupt is generated, and it is restored to its previous state (set or cleared) when the IRET processor instruction is executed. Software can also set or clear the IF bit by using the STI and CLI processor instructions, respectively. The FLAGS register is an internal processor register, not a peripheral control block register, so it is described in the *Am186 and Am188 Family Instruction Set Manual*, order #21267.

Channel 5 Control (CH5CON) Offset 30Ah

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	*Res*												MSK	PR		
Software Read/Write	R												R/W	R/W		
Hardware Set/Clear	—												—	—		
Chip Reset Default	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1

Microcontroller:	
Am186CC	✓
Am186CH	✓
Am186CU	□

Register Description

This is the Channel 5 Interrupt Control register. Channel 5 serves the HDLC channel B and SmartDMA pair 1 sources. Interrupt channel 5 behaves as a level-triggered channel.

Bit Definitions

Bit	Name	Function
15–4	Res	**Reserved** For compatibility with future devices, always write this bit field with its chip reset default value.
3	MSK	**Mask** 0 = Enables interrupt. 1 = Masks (disables) interrupt.
2–0	PR	**Priority Level** Software can use the PR bit field to program the interrupt channel's priority level relative to other maskable interrupts. If two or more interrupts are requested at the same time, the interrupt priority determines which interrupt is serviced first. If the PR bit fields are the same, the interrupts' overall priority is used (see Table 9-3 on page 9-34). Nonmaskable interrupts always have priority over maskable interrupts. 000 = Priority 0 (Highest) 001 = Priority 1 010 = Priority 2 011 = Priority 3 100 = Priority 4 101 = Priority 5 110 = Priority 6 111 = Priority 7 (Lowest) The PRIMSK register (see page 9-39) can be used to mask interrupts according to their PR bit field settings.

Programming Notes

In addition to clearing the MSK bit in the CH5CON register, software must also set the appropriate bit in the source peripheral's register set to enable an event as an interrupt source. For HDLC channel B, interrupts are controlled by the HDLC Channel B Interrupt Mask 0 (HBIMSK0) and HDLC Channel B Interrupt Mask 1 (HBIMSK1) registers (see page 2-18 and page 2-22). For SmartDMA channel pair 1, interrupts are controlled by the SmartDMA Channel Pair 1 Control (SD1CON) register (see page 3-14).

All maskable interrupts are disabled if the IF bit is cleared in the Processor Status Flags (FLAGS) register. The IF bit is cleared automatically when a maskable interrupt is generated, and it is restored to its previous state (set or cleared) when the IRET processor instruction is executed. Software can also set or clear the IF bit by using the STI and CLI processor instructions, respectively. The FLAGS register is an internal processor register, not a peripheral control block register, so it is described in the *Am186 and Am188 Family Instruction Set Manual*, order #21267.

Channel 6 Control (CH6CON) Offset 30Ch

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	*Res*												MSK	PR		
Software Read/Write	R												R/W	R/W		
Hardware Set/Clear	—												—	—		
Chip Reset Default	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1

Microcontroller:	
Am186CC	✔
Am186CH	
Am186CU	!

! *= See description.*

Register Description

This is the Channel 6 Interrupt Control register. Channel 6 serves the HDLC channel C and SDMA pair 2 sources. Interrupt channel 6 behaves as a level-triggered channel.

Am186CU Microcontroller: *HDLC is not supported on the Am186CU USB microcontroller, but SmartDMA channel pair 2 is available for use with USB endpoints A and B.*

Bit Definitions

Bit	Name	Function
15–4	Res	**Reserved** For compatibility with future devices, always write this bit field with its chip reset default value.
3	MSK	**Mask** 0 = Enables interrupt. 1 = Masks (disables) interrupt.
2–0	PR	**Priority Level** Software can use the PR bit field to program the interrupt channel's priority level relative to other maskable interrupts. If two or more interrupts are requested at the same time, the interrupt priority determines which interrupt is serviced first. If the PR bit fields are the same, the interrupts' overall priority is used (see Table 9-3 on page 9-34). Nonmaskable interrupts always have priority over maskable interrupts. 000 = Priority 0 (Highest) 001 = Priority 1 010 = Priority 2 011 = Priority 3 100 = Priority 4 101 = Priority 5 110 = Priority 6 111 = Priority 7 (Lowest) The PRIMSK register (see page 9-39) can be used to mask interrupts according to their PR bit field settings.

Programming Notes

In addition to clearing the MSK bit in the CH6CON register, software must also set the appropriate bit in the source peripheral’s register set to enable an event as an interrupt source. For HDLC channel C, interrupts are controlled by the HDLC Channel C Interrupt Mask 0 (HCIMSK0) and HDLC Channel C Interrupt Mask 1 (HCIMSK1) registers (see page 2-18 and page 2-22). For SmartDMA channel pair 2, interrupts are controlled by the SmartDMA Channel Pair 2 Control (SD2CON) register (see page 3-17).

All maskable interrupts are disabled if the IF bit is cleared in the Processor Status Flags (FLAGS) register. The IF bit is cleared automatically when a maskable interrupt is generated, and it is restored to its previous state (set or cleared) when the IRET processor instruction is executed. Software can also set or clear the IF bit by using the STI and CLI processor instructions, respectively. The FLAGS register is an internal processor register, not a peripheral control block register, so it is described in the *Am186 and Am188 Family Instruction Set Manual*, order #21267.

Channel 7 Control (CH7CON) Offset 30Eh

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	*Res*												MSK	PR		
Software Read/Write	R												R/W	R/W		
Hardware Set/Clear	—												—	—		
Chip Reset Default	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1

Microcontroller:	
Am186CC	✔
Am186CH	
Am186CU	!

! = *See description.*

Register Description

This is the Channel 7 Interrupt Control register. Channel 7 serves the HDLC channel D and SDMA pair 3 sources. Interrupt channel 7 behaves as a level-triggered channel.

Am186CU Microcontroller: *HDLC is not supported on the Am186CU USB microcontroller, but SmartDMA channel pair 3 is available for use with USB endpoints C and D.*

Bit Definitions

Bit	Name	Function
15–4	Res	**Reserved** For compatibility with future devices, always write this bit field with its chip reset default value.
3	MSK	**Mask** 0 = Enables interrupt. 1 = Masks (disables) interrupt.
2–0	PR	**Priority Level** Software can use the PR bit field to program the interrupt channel's priority level relative to other maskable interrupts. If two or more interrupts are requested at the same time, the interrupt priority determines which interrupt is serviced first. If the PR bit fields are the same, the interrupts' overall priority is used (see Table 9-3 on page 9-34). Nonmaskable interrupts always have priority over maskable interrupts. 000 = Priority 0 (Highest) 001 = Priority 1 010 = Priority 2 011 = Priority 3 100 = Priority 4 101 = Priority 5 110 = Priority 6 111 = Priority 7 (Lowest) The PRIMSK register (see page 9-39) can be used to mask interrupts according to their PR bit field settings.

Programming Notes

In addition to clearing the MSK bit in the CH7CON register, software must also set the appropriate bit in the source peripheral’s register set to enable an event as an interrupt source. For HDLC channel D, interrupts are controlled by the HDLC Channel D Interrupt Mask 0 (HDIMSK0) and HDLC Channel D Interrupt Mask 1 (HDIMSK1) registers (see page 2-18 and page 2-22). For SmartDMA channel pair 3, interrupts are controlled by the SmartDMA Channel Pair 3 Control (SD3CON) register (see page 3-17).

All maskable interrupts are disabled if the IF bit is cleared in the Processor Status Flags (FLAGS) register. The IF bit is cleared automatically when a maskable interrupt is generated, and it is restored to its previous state (set or cleared) when the IRET processor instruction is executed. Software can also set or clear the IF bit by using the STI and CLI processor instructions, respectively. The FLAGS register is an internal processor register, not a peripheral control block register, so it is described in the *Am186 and Am188 Family Instruction Set Manual*, order #21267.

Channel 8 Control (CH8CON) Offset 310h

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	*Res*										SRC	LTM	MSK	PR		
Software Read/Write	R										R/W	R/W	R/W	R/W		
Hardware Set/Clear	—										—	—	—	—		
Chip Reset Default	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1

Microcontroller:	
Am186CC	✔
Am186CH	!
Am186CU	!

! *= See bit definitions.*

Register Description

This is the Channel 8 Interrupt Control register. Channel 8 serves the GCI (internal) or the INT3 pin (external) sources. The polarity, active High versus active Low, of the external interrupt on this channel can be set via a bit in the INTPOL register (see page 9-46).

Bit Definitions

Bit	Name	Function
15–6	Res	**Reserved** For compatibility with future devices, always write this bit field with its chip reset default value.
5	SRC	**Source** 0 = External source 1 = Internal source ***Am186CH and Am186CU Microcontrollers:*** *The internal source (GCI) is not supported on the Am186CH and Am186CU microcontrollers.*
4	LTM	**Level-Triggered Mode** 0 = Edge-triggered 1 = Level-triggered; held at 1 if the SRC bit is 1
3	MSK	**Mask** 0 = Enables interrupt. 1 = Masks (disables) interrupt.
2–0	PR	**Priority Level** Software can use the PR bit field to program the interrupt channel's priority level relative to other maskable interrupts. If two or more interrupts are requested at the same time, the interrupt priority determines which interrupt is serviced first. If the PR bit fields are the same, the interrupts' overall priority is used (see Table 9-3 on page 9-34). Nonmaskable interrupts always have priority over maskable interrupts. 000 = Priority 0 (Highest) 001 = Priority 1 010 = Priority 2 011 = Priority 3 100 = Priority 4 101 = Priority 5 110 = Priority 6 111 = Priority 7 (Lowest) The PRIMSK register (see page 9-39) can be used to mask interrupts according to their PR bit field settings.

Programming Notes

In addition to setting the SRC bit and clearing the MSK bit in the CH8CON register, software must also set the appropriate bit in the GCI Interrupt Mask (GIMSK) register to enable a GCI event as an interrupt source (see page 6-6).

All maskable interrupts are disabled if the IF bit is cleared in the Processor Status Flags (FLAGS) register. The IF bit is cleared automatically when a maskable interrupt is generated, and it is restored to its previous state (set or cleared) when the IRET processor instruction is executed. Software can also set or clear the IF bit by using the STI and CLI processor instructions, respectively. The FLAGS register is an internal processor register, not a peripheral control block register, so it is described in the *Am186 and Am188 Family Instruction Set Manual*, order #21267.

Channel 9 Control (CH9CON) Offset 312h

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	*Res*										SRC	LTM	MSK	PR		
Software Read/Write	R										R/W	R/W	R/W	R/W		
Hardware Set/Clear	—										—	—	—	—		
Chip Reset Default	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1

Microcontroller:	
Am186CC	✓
Am186CH	✓
Am186CU	✓

Register Description

This is the Channel 9 Interrupt Control register. Channel 9 serves the general-purpose DMA 0 and 1 (internal) or the INT4 pin (external) sources. The polarity, active High versus active Low, of the external interrupt on this channel can be set via a bit in the INTPOL register (see page 9-46).

Bit Definitions

Bit	Name	Function
15–6	Res	**Reserved** For compatibility with future devices, always write this bit field with its chip reset default value.
5	SRC	**Source** 0 = External source 1 = Internal source
4	LTM	**Level-Triggered Mode** 0 = Edge-triggered 1 = Level-triggered; held at 1 if the SRC bit is 1
3	MSK	**Mask** 0 = Enables interrupt. 1 = Masks (disables) interrupt.
2–0	PR	**Priority Level** Software can use the PR bit field to program the interrupt channel's priority level relative to other maskable interrupts. If two or more interrupts are requested at the same time, the interrupt priority determines which interrupt is serviced first. If the PR bit fields are the same, the interrupts' overall priority is used (see Table 9-3 on page 9-34). Nonmaskable interrupts always have priority over maskable interrupts. 000 = Priority 0 (Highest) 001 = Priority 1 010 = Priority 2 011 = Priority 3 100 = Priority 4 101 = Priority 5 110 = Priority 6 111 = Priority 7 (Lowest) The PRIMSK register (see page 9-39) can be used to mask interrupts according to their PR bit field settings.

Programming Notes

In addition to setting the SRC bit and clearing the MSK bit in the CH9CON register, software must also set the INT bit in the appropriate General-Purpose DMA Control 0 (GDxCON0) register to enable general-purpose DMA channel 0 or 1 as an interrupt source (see page 3-4).

All maskable interrupts are disabled if the IF bit is cleared in the Processor Status Flags (FLAGS) register. The IF bit is cleared automatically when a maskable interrupt is generated, and it is restored to its previous state (set or cleared) when the IRET processor instruction is executed. Software can also set or clear the IF bit by using the STI and CLI processor instructions, respectively. The FLAGS register is an internal processor register, not a peripheral control block register, so it is described in the *Am186 and Am188 Family Instruction Set Manual*, order #21267.

Channel 10 Control (CH10CON) — Offset 314h

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	*Res*										SRC	LTM	MSK	PR		
Software Read/Write	R										R/W	R/W	R/W	R/W		
Hardware Set/Clear	—										—	—	—	—		
Chip Reset Default	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1

Microcontroller:	
Am186CC	✔
Am186CH	✔
Am186CU	✔

Register Description

This is the Channel 10 Interrupt Control register. Channel 10 serves the general-purpose DMA 2 and 3 (internal) or the INT5 pin (external) sources. The polarity, active High versus active Low, of the external interrupt on this channel can be set via a bit in the INTPOL register (see page 9-46).

Bit Definitions

Bit	Name	Function
15–6	Res	**Reserved** For compatibility with future devices, always write this bit field with its chip reset default value.
5	SRC	**Source** 0 = External source 1 = Internal source
4	LTM	**Level-Triggered Mode** 0 = Edge-triggered 1 = Level-triggered; held at 1 if the SRC bit is 1
3	MSK	**Mask** 0 = Enables interrupt. 1 = Masks (disables) interrupt.
2–0	PR	**Priority Level** Software can use the PR bit field to program the interrupt channel's priority level relative to other maskable interrupts. If two or more interrupts are requested at the same time, the interrupt priority determines which interrupt is serviced first. If the PR bit fields are the same, the interrupts' overall priority is used (see Table 9-3 on page 9-34). Nonmaskable interrupts always have priority over maskable interrupts. 000 = Priority 0 (Highest) 001 = Priority 1 010 = Priority 2 011 = Priority 3 100 = Priority 4 101 = Priority 5 110 = Priority 6 111 = Priority 7 (Lowest) The PRIMSK register (see page 9-39) can be used to mask interrupts according to their PR bit field settings.

Programming Notes

In addition to setting the SRC bit and clearing the MSK bit in the CH10CON register, software must also set the INT bit in the appropriate General-Purpose DMA Control 0 (GDxCON0) register to enable general-purpose DMA channel 2 or 3 as an interrupt source (see page 3-4).

All maskable interrupts are disabled if the IF bit is cleared in the Processor Status Flags (FLAGS) register. The IF bit is cleared automatically when a maskable interrupt is generated, and it is restored to its previous state (set or cleared) when the IRET processor instruction is executed. Software can also set or clear the IF bit by using the STI and CLI processor instructions, respectively. The FLAGS register is an internal processor register, not a peripheral control block register, so it is described in the *Am186 and Am188 Family Instruction Set Manual*, order #21267.

Channel 11 Control (CH11CON) Offset 316h

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	*Res*										SRC	LTM	MSK	PR		
Software Read/Write	R										R/W	R/W	R/W	R/W		
Hardware Set/Clear	—										—	—	—	—		
Chip Reset Default	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1

Microcontroller:
Am186CC ✔
Am186CH ✔
Am186CU ✔

Register Description

This is the Channel 11 Interrupt Control register. Channel 11 serves the UART (internal) or the INT6 pin (external) sources. The polarity, active High versus active Low, of the external interrupt on this channel can be set via a bit in the INTPOL register (see page 9-46).

Bit Definitions

Bit	Name	Function
15–6	Res	**Reserved** For compatibility with future devices, always write this bit field with its chip reset default value.
5	SRC	**Source** 0 = External source 1 = Internal source
4	LTM	**Level-Triggered Mode** 0 = Edge-triggered 1 = Level-triggered; held at 1 if the SRC bit is 1
3	MSK	**Mask** 0 = Enables interrupt. 1 = Masks (disables) interrupt.
2–0	PR	**Priority Level** Software can use the PR bit field to program the interrupt channel's priority level relative to other maskable interrupts. If two or more interrupts are requested at the same time, the interrupt priority determines which interrupt is serviced first. If the PR bit fields are the same, the interrupts' overall priority is used (see Table 9-3 on page 9-34). Nonmaskable interrupts always have priority over maskable interrupts. 000 = Priority 0 (Highest) 001 = Priority 1 010 = Priority 2 011 = Priority 3 100 = Priority 4 101 = Priority 5 110 = Priority 6 111 = Priority 7 (Lowest) The PRIMSK register (see page 9-39) can be used to mask interrupts according to their PR bit field settings.

Programming Notes

In addition to setting the SRC bit and clearing the MSK bit in the CH11CON register, software must also set the appropriate bits in the Serial Port Control 0 (SPCON0) and Serial Port Interrupt Mask (SPIMSK) registers to enable a UART event as an interrupt source (see page 5-28 and page 5-33).

The INT6 signal is multiplexed on the same pin with the PIO19 signal. To use INT6 as an interrupt source, software must clear the PIO19 signal's PIO mode and direction bits in the PIOMODE1 and PIODIR1 registers (see page 13-9 and page 13-10). This enables the interrupt input and disables the corresponding PIO output so it cannot affect the interrupt state.

All maskable interrupts are disabled if the IF bit is cleared in the Processor Status Flags (FLAGS) register. The IF bit is cleared automatically when a maskable interrupt is generated, and it is restored to its previous state (set or cleared) when the IRET processor instruction is executed. Software can also set or clear the IF bit by using the STI and CLI processor instructions, respectively. The FLAGS register is an internal processor register, not a peripheral control block register, so it is described in the *Am186 and Am188 Family Instruction Set Manual*, order #21267.

Channel 12 Control (CH12CON) Offset 318h

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	*Res*											LTM	MSK	PR		
Software Read/Write	R											R/W	R/W	R/W		
Hardware Set/Clear	—											—	—	—		
Chip Reset Default	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1

Microcontroller:
Am186CC ✔
Am186CH ✔
Am186CU ✔

Register Description

This is the Channel 12 Interrupt Control register. Channel 12 serves the INT7 pin source. The polarity, active High versus active Low, of the external interrupt on this channel can be set via a bit in the INTPOL register (see page 9-46).

Bit Definitions

Bit	Name	Function
15–5	Res	**Reserved** For compatibility with future devices, always write this bit field with its chip reset default value.
4	LTM	**Level-Triggered Mode** 0 = Edge-triggered 1 = Level-triggered Must be programmed to 0 for Pulse-Width Demodulation (PWD) mode.
3	MSK	**Mask** 0 = Enables interrupt. 1 = Masks (disables) interrupt.
2–0	PR	**Priority Level** Software can use the PR bit field to program the interrupt channel's priority level relative to other maskable interrupts. If two or more interrupts are requested at the same time, the interrupt priority determines which interrupt is serviced first. If the PR bit fields are the same, the interrupts' overall priority is used (see Table 9-3 on page 9-34). Nonmaskable interrupts always have priority over maskable interrupts. 000 = Priority 0 (Highest) 001 = Priority 1 010 = Priority 2 011 = Priority 3 100 = Priority 4 101 = Priority 5 110 = Priority 6 111 = Priority 7 (Lowest) The PRIMSK register (see page 9-39) can be used to mask interrupts according to their PR bit field settings.

Programming Notes

Interrupt channels 12 and 13 are used by the Am186CC/CH/CU microcontrollers's pulse-width demodulation (PWD) feature when the PWD bit is set in the System Configuration (SYSCON) register (see page 16-2).

The INT7 signal is multiplexed on the same pin with the PIO7 signal. To use INT7 as an interrupt source, software must clear the PIO7 signal's PIO mode and direction bits in the PIOMODE0 and PIODIR0 registers (see page 13-4 and page 13-5). This enables the interrupt input and disables the corresponding PIO output so it cannot affect the interrupt state.

All maskable interrupts are disabled if the IF bit is cleared in the Processor Status Flags (FLAGS) register. The IF bit is cleared automatically when a maskable interrupt is generated, and it is restored to its previous state (set or cleared) when the IRET processor instruction is executed. Software can also set or clear the IF bit by using the STI and CLI processor instructions, respectively. The FLAGS register is an internal processor register, not a peripheral control block register, so it is described in the *Am186 and Am188 Family Instruction Set Manual*, order #21267.

Channel 13 Control (CH13CON) Offset 31Ah

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	*Res*											LTM	MSK	PR		
Software Read/Write	R											R/W	R/W	R/W		
Hardware Set/Clear	—											—	—	—		
Chip Reset Default	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1

Microcontroller:
Am186CC ✔
Am186CH ✔
Am186CU ✔

Register Description

This is the Channel 13 Interrupt Control register. Channel 13 serves the INT8 pin source. The polarity, active High versus active Low, of the external interrupt on this channel can be set via a bit in the INTPOL register (see page 9-46).

Bit Definitions

Bit	Name	Function
15–5	Res	**Reserved** For compatibility with future devices, always write this bit field with its chip reset default value.
4	LTM	**Level-Triggered Mode** 0 = Edge-triggered 1 = Level-triggered Must be programmed to 0 for Pulse-Width Demodulation (PWD) mode.
3	MSK	**Mask** 0 = Enables interrupt. 1 = Masks (disables) interrupt.
2–0	PR	**Priority Level** Software can use the PR bit field to program the interrupt channel's priority level relative to other maskable interrupts. If two or more interrupts are requested at the same time, the interrupt priority determines which interrupt is serviced first. If the PR bit fields are the same, the interrupts' overall priority is used (see Table 9-3 on page 9-34). Nonmaskable interrupts always have priority over maskable interrupts. 000 = Priority 0 (Highest) 001 = Priority 1 010 = Priority 2 011 = Priority 3 100 = Priority 4 101 = Priority 5 110 = Priority 6 111 = Priority 7 (Lowest) The PRIMSK register (see page 9-39) can be used to mask interrupts according to their PR bit field settings.

Programming Notes

Interrupt channels 12 and 13 are used by the Am186CC/CH/CU microcontrollers's pulse-width demodulation (PWD) feature when the PWD bit is set in the System Configuration (SYSCON) register (see page 16-2).

The INT8 signal is multiplexed on the same pin with the PIO6 signal. To use INT8 as an interrupt source, software must clear the PIO6 signal's PIO mode and direction bits in the PIOMODE0 and PIODIR0 registers (see page 13-4 and page 13-5). This enables the interrupt input and disables the corresponding PIO output so it cannot affect the interrupt state.

All maskable interrupts are disabled if the IF bit is cleared in the Processor Status Flags (FLAGS) register. The IF bit is cleared automatically when a maskable interrupt is generated, and it is restored to its previous state (set or cleared) when the IRET processor instruction is executed. Software can also set or clear the IF bit by using the STI and CLI processor instructions, respectively. The FLAGS register is an internal processor register, not a peripheral control block register, so it is described in the *Am186 and Am188 Family Instruction Set Manual*, order #21267.

Channel 14 Control (CH14CON) Offset 31Ch

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	*Res*												MSK	PR		
Software Read/Write	R												R/W	R/W		
Hardware Set/Clear	—												—	—		
Chip Reset Default	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1

Microcontroller:	
Am186CC	✔
Am186CH	✔
Am186CU	✔

Register Description

This is the Channel 14 Interrupt Control register. Channel 14 serves the Shared Request (SHREQ) source register (see page 9-44). It behaves as a level-triggered channel.

The polarity, active High versus active Low, of the sources on this channel can be set via bits in the INTPOL and PIOPOL registers (see page 9-46 and page 9-47).

Bit Definitions

Bit	Name	Function
15–4	Res	**Reserved** For compatibility with future devices, always write this bit field with its chip reset default value.
3	MSK	**Mask** 0 = Enables interrupt. 1 = Masks (disables) interrupt.
2–0	PR	**Priority Level** Software can use the PR bit field to program the interrupt channel's priority level relative to other maskable interrupts. If two or more interrupts are requested at the same time, the interrupt priority determines which interrupt is serviced first. If the PR bit fields are the same, the interrupts' overall priority is used (see Table 9-3 on page 9-34). Nonmaskable interrupts always have priority over maskable interrupts. 000 = Priority 0 (Highest) 001 = Priority 1 010 = Priority 2 011 = Priority 3 100 = Priority 4 101 = Priority 5 110 = Priority 6 111 = Priority 7 (Lowest) The PRIMSK register (see page 9-39) can be used to mask interrupts according to their PR bit field settings.

Programming Notes

All maskable interrupts are disabled if the IF bit is cleared in the Processor Status Flags (FLAGS) register. The IF bit is cleared automatically when a maskable interrupt is generated, and it is restored to its previous state (set or cleared) when the IRET processor instruction is executed. Software can also set or clear the IF bit by using the STI and CLI processor instructions, respectively. The FLAGS register is an internal processor register, not a peripheral control block register, so it is described in the *Am186 and Am188 Family Instruction Set Manual*, order #21267.

End-of-Interrupt (EOI) Offset 320h

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	NSPEC	*Res*										S				
Software Read/Write	W	R										W				
Hardware Set/Clear	C	—										C				
Chip Reset Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Microcontroller:	
Am186CC	✔
Am186CH	✔
Am186CU	✔

Register Description

This register is written by software to perform either a specific or non-specific EOI.

Bit Definitions

Bit	Name	Function
15	NSPEC	**Non-Specific EOI** 0 = When the NSPEC bit is cleared, a specific EOI is performed on the interrupt type specified in the S bit field. 1 = When this bit is set, a non-specific EOI is executed. All channels that match the highest priority in service have their In-Service bits cleared.
14–5	Res	**Reserved** For compatibility with future devices, always write this bit field with its chip reset default value.
4–0	S	**Source Interrupt Type** The S bit field specifies the type of interrupt that is currently in service. The channel that serves this type has its In-Service bit cleared during a specific EOI.

Programming Notes

Software must write to the End-of-Interrupt (EOI) register to reset the CHx bit in the INSERV register (see page 9-40) when an interrupt service routine is completed. There are two types of writes to the EOI register—specific EOI and non-specific EOI.

In a specific EOI, software must specify the interrupt type in the S bit field to indicate which CHx bit is to be reset. Specific EOI is applicable when interrupt nesting is possible or when the highest priority CHx bit that was set does not belong to the service routine in progress.

In a non-specific EOI, software does not specify which CHx bit is to be reset. Instead, the interrupt controller clears the CHx bits for all interrupt channels whose priorities match that of the highest priority interrupt in service.

The following table lists interrupt types. For more information, see the *Am186CC/CH/CU Microcontrollers User's Manual*, order #21914.

Table 9-3 Interrupt Types

Interrupt Source	Interrupt/ EOI Type	Vector Table Address	Related Instruction or Channel[1]	Overall Priority
Nonmaskable Interrupts				
Divide Error Exception	00h	00h	DIV, IDIV	1C[2]
Trace Interrupt	01h	04h	All	1A
NMI / Watchdog	02h	08h		1B
Breakpoint Interrupt	03h	0Ch	INT3	1C[2]
INTO Detected Overflow Exception	04h	10h	INTO	1C[2]
Array Bounds Exception	05h	14h	BOUND	1C[2]
Unused Opcode Exception	06h	18h	Undefined Opcodes	1C[2]
ESC Opcode Exception	07h	1Ch	ESC Opcodes	1C[2]
Maskable Interrupts				
Timer 0	08h	20h	Channel 0	2A
Timer 1	09h	24h	Channel 0	2B
Timer 2	0Ah	28h	Channel 0	2C
INT0	0Bh	2Ch	Channel 1	3
INT1[3] / USB[4]	0Ch	30h	Channel 2	4
INT2[3] / High-Speed UART	0Dh	34h	Channel 3	5
HDLC A[5]	0Eh	38h	Channel 4	6A
SDMA0	0Fh	3Ch	Channel 4	6B
HDLC B[5]	10h	40h	Channel 5	7A
SDMA1	11h	44h	Channel 5	7B
HDLC C[6]	12h	48h	Channel 6	8A
SDMA2	13h	4Ch	Channel 6	8B
HDLC D[6]	14h	50h	Channel 7	9A
SDMA3	15h	54h	Channel 7	9B
INT3[3] / GCI[6]	16h	58h	Channel 8	10A
INT4[3] / GP DMA0	17h	5Ch	Channel 9	10B
GP DMA1	18h	60h	Channel 9	10A
INT5[3] / GP DMA2	19h	64h	Channel 10	11A
GP DMA3	1Ah	68h	Channel 10	11B
INT6[3] / UART	1Bh	6Ch	Channel 11	12
INT7[3] / 2nd PWD[7]	1Ch	70h	Channel 12	13
INT8 / PWD[7]	1Dh	74h	Channel 13	14
PIO5, PIO15, PIO27, PIO29, PIO30, PIO33–PIO35 / INT 7–1 (Channel 14)[8]	1Eh	78h	Channel 14	15

Notes:

[1] See the Am186 and Am188 Family Instruction Set Manual, order #21267, for more information about the instructions. See Table 9-2 on page 9-3 for more information about the channels.

[2] These software exceptions can only occur one at a time, so there is no further priority breakdown.

[3] The type and overall priority for the INT1–INT7 pins in this table assume that these pins are being serviced by a dedicated channel; that is they are not being serviced by channel 14. When the INT1–INT7 pins are being serviced by Channel 14, they share type 1Eh, overall priority 15, as indicated by the last row in Table 9-3.

[4] USB is supported on the Am186CC and Am186CU microcontrollers only.

[5] HDLC channels A and B are supported on the Am186CC and Am186CH microcontrollers only.

[6] GCI and HDLC channels C and D are supported on the Am186CC microcontroller only.

[7] PWD is generated on the Low-to-High transition of the PWD input; the second PWD is generated on the High-to-Low transition.

[8] See the SHREQ register description on page 9-44 for information on shared Channel 14.

Poll (POLL) Offset 322h

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	IREQ	*Res*										S				
Software Read/Write	R	R										R				
Hardware Set/Clear	S/C	—										S/C				
Chip Reset Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Microcontroller:

Am186CC	✓
Am186CH	✓
Am186CU	✓

Register Description

Reading this register in a polled environment places the current interrupt in service and advances the next interrupt into the Poll register. The appropriate INSERV register bit is set (see page 9-40), but software must execute the appropriate interrupt service routine.

Bit Definitions

Bit	Name	Function
15	IREQ	**Interrupt Request** 0 = No interrupts are pending. 1 = An interrupt is pending, and the data in the S bit field is valid.
14–5	Res	**Reserved** For compatibility with future devices, always write this bit field with its chip reset default value.
4–0	S	**Poll Status** The S bit field indicates the interrupt type of the highest priority pending interrupt. Reading the POLL register acknowledges the highest priority pending interrupt by setting the INSERV register bit (see page 9-40) for that particular channel. Reading the POLL register also allows the next interrupt to advance into the POLL register. The S bit field does not change unless the IREQ bit is set. That is, if the last pending interrupt is taken and the IREQ bit clears, the S bit field contains the type number of the last interrupt taken.

Programming Notes

The interrupt controller can be used in polled mode if interrupts are not desired. In polled mode, software disables interrupts and polls the interrupt controller as required.

The interrupt controller is polled by reading the POLLST register (see page 9-36). The IREQ bit in the POLLST register indicates to the processor that an interrupt is pending. The S bit field indicates 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 CHx bit in the INSERV register (see page 9-40) for the interrupt indicated in the type field to be set, and allows the next highest priority interrupt to move into the POLL register.

Poll Status (POLLST) Offset 324h

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	IREQ	*Res*										S				
Software Read/Write	R	R										R				
Hardware Set/Clear	S/C	—										S/C				
Chip Reset Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Microcontroller:
Am186CC ☑
Am186CH ☑
Am186CU ☑

Register Description

This register mirrors the current state of the POLL register. In a polled environment, software reads this register to test the IREQ bit without affecting the status of the current interrupt request.

Bit Definitions

Bit	Name	Function
15	IREQ	**Interrupt Request** 0 = No interrupts are pending. 1 = An interrupt is pending, and the data in the S bit field is valid.
14–5	Res	**Reserved** For compatibility with future devices, always write this bit field with its chip reset default value.
4–0	S	**Poll Status** The S bit field indicates the interrupt type of the highest priority pending interrupt. Reading the POLLST register has no effect on the rest of the system. The S bit field does not change unless the IREQ bit is set. That is, if the last pending interrupt is taken and the IREQ bit clears, the S bit field contains the type number of the last interrupt taken.

Programming Notes

The interrupt controller can be used in polled mode if interrupts are not desired. In polled mode, software disables interrupts and polls the interrupt controller as required.

The interrupt controller is polled by reading the POLLST register. The IREQ bit in the POLLST register indicates to the processor that an interrupt is pending. The S bit field indicates 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 CHx bit in the INSERV register (see page 9-40) for the interrupt indicated in the type field to be set, and allows the next highest priority interrupt to move into the POLL register (see page 9-35).

Interrupt Mask (IMASK) Offset 326h

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	*Res*	CH14	CH13	CH12	CH11	CH10	CH9	CH8	CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0
Software Read/Write	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Hardware Set/Clear	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Chip Reset Default	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Microcontroller:

Am186CC	✔
Am186CH	✔
Am186CU	✔

Register Description

This register contains the mask bits for interrupt channels 0–14. These are the same physical mask bits that exist in all of the Channel Control registers, but here all channels are accessible at one time. Changing a bit in this register changes the MSK bit in the corresponding interrupt channel's control register, and vice-versa.

Bit Definitions

Bit	Name	Function
15	Res	**Reserved** For compatibility with future devices, always write this bit field with its chip reset default value.
14–0	CH14–CH0	**Channel Mask** Each of the CH14–CH0 bits corresponds to an interrupt channel. For example, the CH14 bit is the mask bit for Channel 14. 0 = Enables interrupt. 1 = Masks (disables) interrupt.

Programming Notes

All maskable interrupts are disabled if the IF bit is cleared in the Processor Status Flags (FLAGS) register. The IF bit is cleared automatically when a maskable interrupt is generated, and it is restored to its previous state (set or cleared) when the IRET processor instruction is executed. Software can also set or clear the IF bit by using the STI and CLI processor instructions, respectively. The FLAGS register is an internal processor register, not a peripheral control block register, so it is described in the *Am186 and Am188 Family Instruction Set Manual*, order #21267.

Do not write to the IMASK register while interrupts are enabled (the IF bit in the FLAGS register is set). In this case, spurious interrupt requests may be generated, including requests from devices whose interrupts were disabled both before and after the write to the IMASK register. It is safe to write the MSK bits in the CHxCON registers while interrupts are enabled.

Table 9-4 on page 9-38 lists interrupt channel sources.

Table 9-4 Interrupt Channel Sources (Same as Table 9-2)

Interrupt Channel	Default Source	Optional Source
0	Timer 0,Timer 1, and Timer 2	—
1	INT0	—
2	INT1	USB[1]
3	INT2	High-Speed UART
4	HDLC Channel A[2] and SmartDMA Channel Pair 0[2]	—
5	HDLC Channel B[2] and SmartDMA Channel Pair 1[2]	—
6	HDLC Channel C[3] and SmartDMA Channel Pair 2[1]	—
7	HDLC Channel D[3] and SmartDMA Channel Pair 3[1]	—
8	INT3	GCI[3]
9	INT4	General-Purpose DMA 0 and 1
10	INT5	General-Purpose DMA 2 and 3
11	INT6	UART
12	INT7	PWD[4]
13	INT8	PWD[4]
14	Shared Request[5]	—

Notes:

1. Am186CC and Am186CU microcontrollers only.

2. Am186CC and Am186CH microcontrollers only.

3. Am186CC microcontroller only.

4. The PWD source is selected by setting the PWD bit in the SYSCON register. See page 16-2.

5. The Shared Request source is controlled by the SHREQ and SHMASK registers. The following sources can be enabled to use the Shared Request channel: PIO pins 35, 34, 33, 30, 29, 27, 15, and 5; and INT pins 7–1.

Priority Mask (PRIMSK)   Offset 328h

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	*Res*													PRM		
Software Read/Write	R													R/W		
Hardware Set/Clear	—													—		
Chip Reset Default	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1

Microcontroller:	
Am186CC	✔
Am186CH	✔
Am186CU	✔

Register Description

This register determines the minimum priority required for a maskable interrupt source to be requested.

Bit Definitions

Bit	Name	Function
15–3	Res	**Reserved** For compatibility with future devices, always write this bit field with its chip reset default value.
2–0	PRM	**Priority Field Mask** 000 = Priority 0 (Highest) 001 = Priority 1 010 = Priority 2 011 = Priority 3 100 = Priority 4 101 = Priority 5 110 = Priority 6 111 = Priority 7 (Lowest)

Programming Notes

Programming the PRIMSK register masks any interrupt requests with a lower programmable priority than the priority mask. For example, if PRM reads 011b (3) then only those interrupts with a programmable priority of 000b, 001b, 010b, and 011b (levels 0-3) are allowed. The programmable priority for a channel is set with the PR bit in the appropriate CHxCON register (see the descriptions starting on page 9-4).

In-Service (INSERV) Offset 32Ah

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	*Res*	CH14	CH13	CH12	CH11	CH10	CH9	CH8	CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0
Software Read/Write	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Hardware Set/Clear	—	S/C	S/C	S/C	S/C	S/C	S/C	S/C	S/C	S/C	S/C	S/C	S/C	S/C	S/C	S/C
Chip Reset Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Microcontroller:

- **Am186CC** ☑
- **Am186CH** ☑
- **Am186CU** ☑

Register Description

This register contains the in-service bits for interrupt channels 0–14.

Bit Definitions

Bit	Name	Function
15	Res	**Reserved** For compatibility with future devices, always write this bit field with its chip reset default value.
14–0	CH14–CH0	**Channel In Service** Each of the CH14–CH0 bits corresponds to an interrupt channel. For example, CH14 is the in-service bit for Channel 14. 0 = The channel's interrupt service routine is not active. 1 = The channel's interrupt service routine is active. In-service bits are cleared by writing the interrupt type of the in-service routine to the EOI register, or by setting the non-specific EOI bit (see page 9-33).

Programming Notes

Interrupt Request (REQST) Offset 32Ch

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	*Res*	CH14	CH13	CH12	CH11	CH10	CH9	CH8	CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0
Software Read/Write	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Hardware Set/Clear	—	S/C	S/C	S/C	S/C	S/C	S/C	S/C	S/C	S/C	S/C	S/C	S/C	S/C	S/C	S/C
Chip Reset Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Microcontroller:
Am186CC ✔
Am186CH ✔
Am186CU ✔

Register Description

This register contains the request bits for interrupt channels 0–14.

Bit Definitions

Bit	Name	Function
15	Res	**Reserved** For compatibility with future devices, always write this bit field with its chip reset default value.
14–0	CH14–CH0	**Channel Request** Each of the CH14–CH0 bits corresponds to an interrupt channel. For example, CH14 is the request bit for Channel 14. 0 = If the channel is edge-triggered, the interrupt has not occurred or the channel's in-service bit has been set. If the channel is level-triggered, the interrupt source is not active. 1 = The channel has a pending request.

Programming Notes

If the channel is level-triggered, its request bit reflects the value of the requesting source if active high or the inverted value if active low. If the channel is edge-triggered, its request bit is set when the active edge is seen on the request line, and the request bit is cleared when the corresponding in-service bit is set in the INSERV register (see page 9-40).

Request bits corresponding to internal sources are cleared when the channel's in-service bit is set in the INSERV register.

The request bit for the three timers in channel 0, CH0, is the logical OR of the three TIM bits in the INTSTS register (see page 9-42).

The request bit for channel 9, CH9, is dependent on which sources it is servicing. If channel 9 is servicing the INT4 pin, the CH9 request bit reflects the value of the INT4 pin. If channel 9 is servicing General-Purpose DMA0 and General-Purpose DMA1, then the CH9 request bit is the logical OR of the DMA0 and DMA1 request bits in the Interrupt Status (INTSTS) register. Channel 10 is implemented the same as channel 9, except that channel 10 services either the INT5 pin, or the General-Purpose DMA2 and DMA3 sources.

Interrupt Status (INTSTS) Offset 32Eh

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	*Res*									DMA3	DMA2	DMA1	DMA0	TIM2	TIM1	TIM0
Software Read/Write	R									R/W0	R/W0	R/W0	R/W0	R/W0	R/W0	R/W0
Hardware Set/Clear	—									S/C	S/C	S/C	S/C	S/C	S/C	S/C
Chip Reset Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Microcontroller:	
Am186CC	✔
Am186CH	✔
Am186CU	✔

Register Description

This register indicates the status of the general-purpose DMA and timer interrupt channels.

Bit Definitions

Bit	Name	Function
15–7	Res	**Reserved** For compatibility with future devices, always write this bit field with its chip reset default value.
6–3	DMA3–DMA0	**GP DMA 3–0 Interrupt Request** Each of the DMA3–DMA0 bits corresponds to a general-purpose DMA channel. For example, DMA3 is the status bit for general-purpose DMA 3. 0 = No outstanding interrupt request is pending for this device, or software has cleared this bit. 1 = The device has a pending request. Software can only clear the DMA3–DMA0 bits. Writing a 1 has no effect.
2–0	TIM2–TIM0	**Timer 2–0 Interrupt Request** Each of the TIM2–TIM0 bits corresponds to a timer channel. For example, TIM2 is the request bit for Timer 2. 0 = No outstanding interrupt request is pending for this device, or software has cleared this bit. 1 = The device has a pending request. Software can only clear the TIM2–TIM0 bits. Writing a 1 has no effect.

Programming Notes

A device’s bit in the INTSTS register is set when the interrupt controller latches a request from that device. The bit is cleared when the corresponding interrupt vector is fetched. The interrupt controller clears only the INTSTS bit for the *device* whose vector is fetched (e.g., the TIM2 bit for Timer 2, not the TIM1 or TIM0 bit). When a device’s INTSTS register bit is cleared, the INSERV register bit for the corresponding *channel* is set (e.g., the CH0 bit in the INSERV register is set when any Timer interrupt vector is fetched, see page 9-40).

Bit 15 of the INTSTS register is the legacy location of the DHLT bit, which has been relocated to the DMA Halt register (see page 9-43).

DMA Halt (DMAHLT) Offset 330h

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	DHLT	*Res*														
Software Read/Write	R/W	R														
Hardware Set/Clear	S/C	—														
Chip Reset Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Microcontroller:	
Am186CC	✓
Am186CH	✓
Am186CU	✓

Register Description

This register controls DMA activity when nonmaskable interrupts occur.

Bit Definitions

Bit	Name	Function
15	DHLT	**DMA Halt** 0 = DMA activity is allowed. 1 = All DMA activity is halted. The DHLT bit is set by an NMI and cleared by any IRET instruction. Software can also read and write this bit.
14–0	Res	**Reserved** For compatibility with future devices, always write this bit field with its chip reset default value.

Programming Notes

See Chapter 3, “DMA Registers”, for DMA register information.

Shared Request (SHREQ) Offset 332h

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	PIO35	PIO34	PIO33	PIO30	PIO29	PIO27	PIO15	PIO5	INT7	INT6	INT5	INT4	INT3	INT2	INT1	*Res*
Software Read/Write	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Hardware Set/Clear	S/C	S/C	S/C	S/C	S/C	S/C	S/C	S/C	S/C	S/C	S/C	S/C	S/C	S/C	S/C	—
Chip Reset Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Microcontroller:
Am186CC ✔
Am186CH ✔
Am186CU ✔

Register Description

This register contains interrupt request status bits for the interrupt sources that can share Channel 14.

Bit Definitions

Bit **Name** **Function**

15–8 PIO35 PIO34 PIO33 PIO30 PIO29 PIO27 PIO15 PIO5 **PIO Shared Request**

Each bit in this field corresponds to the named PIO pin. For example, bit PIO35 corresponds to the PIO35 pin.

0 = The corresponding PIO pin is not enabled for shared interrupts or is not requesting an interrupt.

1 = The corresponding PIO pin is enabled for shared interrupts and is currently requesting an interrupt on the shared channel, Channel 14.

Software must configure a PIO pin as a PIO input or output before using it as an interrupt source. See Chapter 13, “Programmable I/O (PIO) Registers”.

7–1 INT7–INT1 **INT Shared Request**

Each of the INT7–INT1 bits corresponds to an INT pin. For example, the INT7 bit corresponds to the INT7 pin.

0 = The corresponding INT pin is not enabled for shared interrupts or is not requesting an interrupt.

1 = The corresponding INT pin is enabled for shared interrupts and is currently requesting an interrupt on the shared channel, Channel 14.

0 Res **Reserved**

For compatibility with future devices, always write this bit field with its chip reset default value.

Programming Notes

All Channel 14 shared interrupt sources are level-triggered only. The requesting device must hold the request pin active until the interrupt is serviced.

Software can mask individual Channel 14 interrupt sources in the SHMASK register (see page 9-45).

The polarity, active High or active Low, of the external sources on Channel 14 can be set via bits in the INTPOL and PIOPOL registers (see page 9-46 and page 9-47).

Shared Mask (SHMASK) Offset 334h

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	PIO35	PIO34	PIO33	PIO30	PIO29	PIO27	PIO15	PIO5	INT7	INT6	INT5	INT4	INT3	INT2	INT1	*Res*
Software Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R
Hardware Set/Clear	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Chip Reset Default	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Microcontroller:

Am186CC ✓
Am186CH ✓
Am186CU ✓

Register Description

This is the mask register for the interrupt sources that can share Channel 14.

Bit Definitions

Bit	Name	Function
15–8	PIO35 PIO34 PIO33 PIO30 PIO29 PIO27 PIO15 PIO5	**PIO Shared Request** Each bit in this field corresponds to the named PIO pin. For example, bit PIO35 corresponds to the PIO35 pin. 0 = The corresponding PIO pin is enabled as a source for Channel 14. 1 = The corresponding PIO pin is masked (disabled) as a source for Channel 14.
7–1	INT7–INT1	**INT Shared Request** Each of the INT7–INT1 bits corresponds to an INT pin. For example, the INT7 bit corresponds to the INT7 pin. 0 = The corresponding INT pin is enabled as a source for Channel 14. 1 = The corresponding INT pin is masked (disabled) as a source for Channel 14.
0	Res	**Reserved** For compatibility with future devices, always write this bit field with its chip reset default value.

Programming Notes

All Channel 14 shared interrupt sources are level-triggered only. The requesting device must hold the request pin active until the interrupt is serviced.

The individual Channel 14 interrupt request bits are in the SHREQ register (see page 9-44).

The polarity, active High or active Low, of the external sources on Channel 14 can be set via bits in the INTPOL and PIOPOL registers (see page 9-46 and page 9-47).

Interrupt Polarity (INTPOL) Offset 336h

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	*Res*							INT8	INT7	INT6	INT5	INT4	INT3	INT2	INT1	INT0
Software Read/Write	R							R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Hardware Set/Clear	—							—	—	—	—	—	—	—	—	—
Chip Reset Default	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Microcontroller:	
Am186CC	✓
Am186CH	✓
Am186CU	✓

Register Description

This register sets the polarity, active High or active Low, of the INT pins.

Bit Definitions

Bit	Name	Function
15–9	Res	**Reserved** For compatibility with future devices, always write this bit field with its chip reset default value.
8–0	INT8–INT0	**INT 8–0 Polarity** Each of the INT8–INT0 bits corresponds to an INT pin. For example, the INT8 bit corresponds to the INT8 pin. 0 = Interrupt active Low 1 = Interrupt active High

Programming Notes

PIO Polarity (PIOPOL) Offset 338h

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	PIO35	PIO34	PIO33	PIO30	PIO29	PIO27	PIO15	PIO5	*Res*							
Software Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R							
Hardware Set/Clear	—	—	—	—	—	—	—	—	—							
Chip Reset Default	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Microcontroller:

Am186CC	✔
Am186CH	✔
Am186CU	✔

Register Description

This register sets the polarity, active High or active Low, of PIO pins that are used as interrupt sources.

Bit Definitions

Bit	Name	Function
15–8	PIO35 PIO34 PIO33 PIO30 PIO29 PIO27 PIO15 PIO5	**PIO Polarity** Each bit in this field corresponds to the named PIO pin. For example, bit PIO35 corresponds to the PIO35 pin. 0 = Interrupt active Low 1 = Interrupt active High
7–0	Res	**Reserved** For compatibility with future devices, always write this bit field with its chip reset default value.

Programming Notes

CHAPTER

10 TIMER REGISTERS

10.1 OVERVIEW

This chapter describes the Timer registers on the Am186CC/CH/CU microcontrollers.

The Am186CC/CH/CU microcontrollers provide three independent 16-bit programmable timer/counters: Timer 0, Timer 1, and Timer 2. All three timers can be polled or used to generate interrupts for real-time coding or time-delay applications.

Timers 0–1 are connected to four external pins. These timers' output pins can be used to generate a variety of waveforms. Their input pins can be used to count or time external events, or as an external source clock input. Otherwise, these timers count at one quarter of the processor clock frequency. Timers 0–1 can be used together to perform pulse-width demodulation (PWD), which measures the Low-state and High-state duration of an input signal. The PWD feature is selected by setting the PWD bit in the System Configuration (SYSCON) register (see page 16-2).

Timer 2 can be used as a DMA request source or as a prescaler for Timer 0 and 1. The source clock for Timer 2 is always one quarter of the processor frequency.

For more information about using timers, see the *Am186CC/CH/CU Microcontrollers User's Manual*, order #21914.

Table 10-1 lists the Timer registers in offset order, with the corresponding description's page number.

Table 10-1 Timer Register Map

Register Name	Mnemonic	Offset	Page Number
Timer 0 Mode/Control	T0CON	0340h	page 10-2
Timer 0 Count	T0CNT	0342h	page 10-5
Timer 0 Maxcount Compare A	T0CMPA	0344h	page 10-6
Timer 0 Maxcount Compare B	T0CMPB	0346h	page 10-7
Timer 1 Mode/Control	T1CON	0348h	page 10-8
Timer 1 Count	T1CNT	034Ah	page 10-11
Timer 1 Maxcount Compare A	T1CMPA	034Ch	page 10-12
Timer 1 Maxcount Compare B	T1CMPB	034Eh	page 10-13
Timer 2 Mode/Control	T2CON	0350h	page 10-14
Timer 2 Count	T2CNT	0352h	page 10-16
Timer 2 Maxcount Compare A	T2CMPA	0354h	page 10-17

Timer 0 Mode/Control (T0CON) Offset 0340h

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	EN	$\overline{\text{INH}}$	INT	RIU	*Res*						MC	RTG	P	EXT	ALT	CONT
Software Read/Write	R/W	W	R/W	R	R						R/W0	R/W	R/W	R/W	R/W	R/W
Hardware Set/Clear	C	—	—	S/C	—						S	—	—	—	—	—
Chip Reset Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Microcontroller:

Am186CC	✔
Am186CH	✔
Am186CU	✔

Register Description

The T0CON register is used to control the functionality and the modes of operation of Timer 0.

Bit Definitions

Bit **Name** **Function**

15 EN **Enable**

0 = Timer 0 is inhibited from counting.

1 = Timer 0 is enabled.

In a scenario where the timer had been previously enabled (via setting the EN bit), was operating, and then software cleared the EN bit, Timer 0 would be inhibited from counting but would not be reset. The various timer status bits and the timer output pin (TMROUT0) also remain stable. In this scenario, setting the EN bit again causes the timer to continue from the state that it was in just prior to being disabled.

The EN bit can only be modified (set or cleared) via software if the $\overline{\text{INH}}$ bit of this register is set (i.e., written as 1) during the same write cycle access to the register.

The EN bit is automatically cleared by hardware under certain circumstances if Non-Continuous mode is selected. See the CONT bit description on page 10-4.

14 $\overline{\text{INH}}$ **Inhibit**

The $\overline{\text{INH}}$ bit allows selective software modifications of the EN bit. When the $\overline{\text{INH}}$ bit is set during a write cycle access to this register, the EN bit can be modified in that same write cycle. When the $\overline{\text{INH}}$ bit is not written as 1 during a write cycle access to this register, the EN bit cannot be modified. The $\overline{\text{INH}}$ bit is always read back as a 0.

13 INT **Interrupt**

The INT bit allows the timer to generate an interrupt when the timer counter value reaches a maximum count compare register value.

0 = The timer does not cause the TIM0 bit in the INTSTS register to be set (see page 9-42); therefore, a timer interrupt is not generated.

1 = An interrupt request is generated each time a maximum count is reached (i.e., each time the MC bit is set, see page 10-3).

In addition to setting the INT bit in this register, software must configure the interrupt Channel 0 Control (CH0CON) register to enable the interrupt channel before any timer interrupts can be generated. See Chapter 9, "Interrupt Controller Registers".

Bit	Name	Function
12	RIU	**Register-In-Use** The RIU bit can be used by software with the MC bit to determine where the timer is in its current count sequence. 0 = Hardware clears the RIU bit when the Timer 0 Maxcount Compare A register is being used for comparison to the Timer 0 count value. The RIU bit is also cleared anytime the timer has been disabled due to the EN bit being cleared under hardware control (i.e., at the end of timer sequence while in Non-Continuous mode). See the CONT bit description on page 10-4. 1 = Hardware sets the RIU bit when the Timer 0 Maxcount Compare B register is being used for comparison to the Timer 0 count value.
11–6	Res	**Reserved** For compatibility with future devices, always write this bit field with its chip reset default value.
5	MC	**Maximum Count** The MC bit can be used by software with the RIU bit to determine where the timer is in its current count sequence. 0 = A maximum count was not reached, or software has cleared the MC bit. The MC bit is never automatically cleared by hardware. Software must clear the MC bit. 1 = Hardware sets the MC bit anytime the timer count value reaches the maximum count value being used (maximum count value A or maximum count value B). The MC bit cannot be set by software. When Timer 0 is in dual maximum count mode (via setting the ALT bit), the MC bit is set whenever the timer 0 count value equals either the Timer 0 Maxcount Compare A or B register value. The MC bit is set for this condition regardless of the state of the INT bit. The MC bit can be used to monitor timer status through software polling instead of making use of interrupt generation.
4	RTG	**Retrigger** The RTG bit determines the control function provided by the Timer 0 input pin (TMRIN0) when TMRIN0 is not configured as the timer clock source (i.e., when the EXT bit is cleared). 0 = When the timer is enabled, a High level on the TMRIN0 input pin allows the timer to count, and a Low level on this pin holds the timer count value constant. The RTG bit is ignored when external clocking is selected (i.e., the EXT bit is set). 1 = When the timer is enabled, a 0 to 1 edge transition on the TMRIN0 pin resets the existing Timer 0 Count register value, then counting continues. The RTG bit is ignored when external clocking is enabled (i.e., the EXT bit is set). The TMRIN0 pin is multiplexed with a Programmable Input Output (PIO) pin. If this pin is configured to be a PIO, the TMRIN0 signal is driven High internal to the chip, and therefore the RTG bit has no effect.
3	P	**Prescaler** The P bit selects the Timer 0 clock source when the TMRIN0 input pin is not configured as the timer clock source (i.e., when the EXT bit is cleared). 0 = The Timer 0 clock source is 1/4 of the CPU clock speed. 1 = Timer 0 is prescaled by Timer 2 (i.e., the Timer 0 count increments each time Timer 2 reaches its maximum count). The P bit is ignored when external clocking is enabled (i.e., the EXT bit is set).
2	EXT	**External Clock** 0 = An internal Timer 0 clock source is used as configured via the P bit. 1 = An external Timer 0 clock source is used (i.e., the TMRIN0 pin). Timer 0 advances on every positive edge driven on the TMRIN0 input pin. In this mode, the maximum timer input clock frequency is 1/4 of the CPU clock speed. The TMRIN0 pin is multiplexed with a Programmable Input Output (PIO) pin. If this pin is configured to be a PIO, the TMRIN0 signal is driven High internal to the chip, and the timer count will never increment if the EXT bit is 1.

Bit	Name	Function
1	ALT	**Alternate Compare** The ALT bit is used to configure Timer 0 for dual or single maximum count mode. 0 = When the timer is enabled, the timer counts to the Timer 0 Maxcount Compare A register value, then resets the Timer 0 Count register value to 0. In this case, the Timer 0 Maxcount Compare B register is not used. In this mode, the TMROUT0 pin is High while the counter is counting and being compared to the Maxcount Compare A register. The TMROUT0 pin is pulsed Low for a single CPU clock cycle after the maximum value is reached. 1 = When the timer is enabled, the timer counts to the Timer 0 Maxcount Compare A register value, then resets the Timer 0 Count register value to 0. The timer then counts to the Timer 0 Maxcount Compare B register value and resets the Timer 0 Count register value to 0. This is referred to as “dual maximum count mode”. In this mode, the TMROUT0 pin is High while the counter is counting and being compared to the Maxcount Compare A register. The TMROUT0 pin is Low while the counter is counting and being compared to the Maxcount Compare B register. In other words, the TMROUT0 pin is the inverse of the RIU bit. If external clocking is used and the clock is stopped during a count sequence, the timer output remains in its previous state (i.e., the state that it was in prior to the clock stopping). The remaining timer status also remains the same, and normal operation commences upon the external clock being driven again. See the CONT bit description for a more detailed description of how the comparison registers are used in Continuous and Non-Continuous modes.
0	CONT	**Continuous Mode** The CONT bit is used to configure Timer 0 for Continuous or Non-Continuous mode. 0 = Timer 0 runs in Non-Continuous mode. The Timer 0 Count register is cleared and the timer halts whenever the count value reaches the maximum count value. The EN bit is also cleared by hardware after every counter sequence. If the CONT bit is cleared and the ALT bit is set, Timer 0 counts to the Timer 0 Maxcount Compare A register value, then resets the count value. After the timer count has been reset, the timer continues operation by counting to the Timer 0 Maxcount Compare B register value. When the timer count value reaches the Maxcount Compare B register value, it resets its count value, clears the EN bit and then halts. 1 = Timer 0 runs in Continuous mode. Continuous mode causes the Timer Count register to be reset to 0 after it reaches the Maxcount Compare A or B register value. The timer immediately begins timing out again.

Programming Notes

Timers 0 and 1 are used by the Am186CC/CH/CU microcontrollers’s pulse-width demodulation (PWD) feature when the PWD bit is set in the System Configuration (SYSCON) register (see page 16-2).

Timer 0 Count (T0CNT) Offset 0342h

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	TC															
Software Read/Write	R/W															
Hardware Set/Clear	S/C															
Chip Reset Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Microcontroller:	
Am186CC	✓
Am186CH	✓
Am186CU	✓

Register Description

The T0CNT register is incremented by 1 each time Timer 0 advances its count. This count register is compared with the Timer 0 Maxcount Compare A or B registers, and various actions are taken when the maximum count is reached.

Bit Definitions

Bit	Name	Function
15–0	TC	**Timer 0 Count** The TC bit field contains the current count of Timer 0. The count is incremented every fourth processor clock cycle when the timer is in internal clocked mode, or each time the Timer 2 maxcount is reached if Timer 0 is configured to be prescaled by Timer 2. Timer 0 can also be configured for external clocking control based on the TMRIN0 signal. The TC bit field can be read at any time to determine the remaining count duration until a maximum count value is reached. The TC bit field can be written at any time. If the TC bit field is written while the counter is enabled, the new value is latched into the Timer 0 counter and counting proceeds from the new value.

Programming Notes

Timer 0 Maxcount Compare A (T0CMPA) Offset 0344h

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	TC															
Software Read/Write	R/W															
Hardware Set/Clear	—															
Chip Reset Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Microcontroller:	
Am186CC	✔
Am186CH	✔
Am186CU	✔

Register Description

The T0CMPA register contains one of two compare values for the Timer 0 Count register. (Timer 0 and Timer 1 each have two Maxcount Compare registers.) If a Maxcount Compare register is written with the value 0000h and the timer is enabled, the timer counts to FFFFh, at which point the appropriate action occurs based on the timer configuration options that are set.

Bit Definitions

Bit	Name	Function
15–0	TC	**Timer 0 Maxcount Compare A** The TC bit field contains one of the maximum values that Timer 0 can count to before resetting its count register to 0.

Programming Notes

Timer 0 Maxcount Compare B (T0CMPB) Offset 0346h

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	TC															
Software Read/Write	R/W															
Hardware Set/Clear	—															
Chip Reset Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Microcontroller:	
Am186CC	✔
Am186CH	✔
Am186CU	✔

Register Description

The T0CMPB register contains one of two compare values for the Timer 0 Count register. (Timer 0 and Timer 1 each have two Maxcount Compare registers.) If a Maxcount Compare register is written with the value 0000h and the timer is enabled, the timer counts to FFFFh, at which point the appropriate action occurs based on the timer configuration options that are set.

Bit Definitions

Bit	Name	Function
15–0	TC	**Timer 0 Maxcount Compare B** The TC bit field contains one of the maximum values that Timer 0 can count to before resetting its count register to 0.

Programming Notes

Timer 1 Mode/Control (T1CON) Offset 0348h

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	EN	$\overline{\text{INH}}$	INT	RIU	Res						MC	RTG	P	EXT	ALT	CONT
Software Read/Write	R/W	W	R/W	R	R						R/W0	R/W	R/W	R/W	R/W	R/W
Hardware Set/Clear	C	—	S	S/C	—						S	—	—	—	—	—
Chip Reset Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Microcontroller:
Am186CC ✔
Am186CH ✔
Am186CU ✔

Register Description

The T1CON register is used to control the functionality and the modes of operation of Timer 1.

Bit Definitions

Bit **Name** **Function**

15 EN **Enable**

0 = Timer 1 is inhibited from counting.

1 = Timer 1 is enabled.

In a scenario where the timer had been previously enabled (via setting the EN bit), was operating, and then software cleared the EN bit, Timer 1 would be inhibited from counting but would not be reset. The various timer status bits and the timer output pin (TMROUT1) also remain stable. In this scenario, setting the EN bit again causes the timer to continue from the state that it was in just prior to being disabled.

The EN bit can only be modified (set or cleared) via software if the $\overline{\text{INH}}$ bit of this register is set (i.e., written as 1) during the same write cycle access to this register.

The EN bit is automatically cleared by hardware under certain circumstances if Non-Continuous mode is selected. See the CONT bit description on page 10-10.

14 $\overline{\text{INH}}$ **Inhibit**

The $\overline{\text{INH}}$ bit allows selective software modifications of the EN bit. When the $\overline{\text{INH}}$ bit is set during a write cycle access to this register, the EN bit can be modified in that same write cycle. When the $\overline{\text{INH}}$ bit is not written as 1 during a write cycle access to this register, the EN bit cannot be modified. The $\overline{\text{INH}}$ bit is always read back as a 0.

13 INT **Interrupt**

The INT bit allows the timer to generate an interrupt when the timer counter value reaches a maximum count compare register value.

0 = The timer does not cause the TIM1 bit in the INTSTS register to be set (see page 9-42); therefore, a timer interrupt is not generated.

1 = An interrupt request is generated each time a maximum count is reached (i.e., each time the MC bit is set, see page 10-9).

In addition to setting the INT bit in this register, software must configure the interrupt Channel 0 Control (CH0CON) register to enable the interrupt channel before any timer interrupts can be generated. See Chapter 9, “Interrupt Controller Registers”.

Bit	Name	Function
12	RIU	**Register-In-Use** The RIU bit can be used by software with the MC bit to determine where the timer is in its current count sequence. 0 = Hardware clears the RIU bit when the Timer 1 Maxcount Compare A register is being used for comparison to the Timer 1 count value. The RIU bit is also cleared anytime the timer has been disabled due to the EN bit being cleared under hardware control (i.e., at the end of timer sequence while in Non-Continuous mode). See the CONT bit description on page 10-10. 1 = Hardware sets the RIU bit when the Timer 1 Maxcount Compare B register is being used for comparison to the Timer 1 count value.
11–6	Res	**Reserved** For compatibility with future devices, always write this bit field with its chip reset default value.
5	MC	**Maximum Count** The MC bit can be used by software with the RIU bit to determine where the timer is in its current count sequence. 0 = A maximum count was not reached, or software has cleared the MC bit. The MC bit is never automatically cleared by hardware. Software must clear the MC bit. 1 = Hardware sets the MC bit anytime the timer count value reaches the maximum count value being used (maximum count value A or maximum count value B). The MC bit cannot be set by software. When Timer 1 is in dual maximum count mode (via setting the ALT bit), the MC bit is set whenever the timer 1 count value equals either the Timer 1 Maxcount Compare A or B register value. The MC bit is set for this condition regardless of the state of the INT bit. The MC bit can be used to monitor timer status through software polling instead of making use of interrupt generation.
4	RTG	**Retrigger** The RTG bit determines the control function provided by the Timer 1 input pin (TMRIN1) when TMRIN1 is not configured as the timer clock source (i.e., when the EXT bit is cleared). 0 = When the timer is enabled, a High level on the TMRIN1 input pin allows the timer to count, and a Low level on this pin holds the timer count value constant. The RTG bit is ignored when external clocking is selected (i.e., the EXT bit is set). 1 = When the timer is enabled, a 0 to 1 edge transition on the TMRIN1 pin resets the existing Timer 1 Count register value, then counting continues. The RTG bit is ignored when external clocking is enabled (i.e., the EXT bit is set). The TMRIN1 pin is multiplexed with a Programmable Input Output (PIO) pin. If this pin is configured to be a PIO, the TMRIN1 signal is driven High internal to the chip, and therefore the RTG bit has no effect.
3	P	**Prescaler** The P bit selects the Timer 1 clock source when the TMRIN1 input pin is not configured as the timer clock source (i.e., when the EXT bit is cleared). 0 = The Timer 1 clock source is 1/4 of the CPU clock speed. 1 = Timer 1 is prescaled by Timer 2 (i.e., the Timer 1 count increments each time Timer 2 reaches its maximum count). The P bit is ignored when external clocking is enabled (i.e., the EXT bit is set).
2	EXT	**External Clock** 0 = An internal Timer 1 clock source is used as configured via the P bit. 1 = An external Timer 1 clock source is used (i.e., the TMRIN1 pin). Timer 1 advances on every positive edge driven on the TMRIN1 input pin. In this mode, the maximum timer input clock frequency is 1/4 of the CPU clock speed. The TMRIN1 pin is multiplexed with a Programmable Input Output (PIO) pin. If this pin is configured to be a PIO, the TMRIN1 signal is driven High internal to the chip, and the timer count will never increment if the EXT bit is 1.

Bit	Name	Function
1	ALT	**Alternate Compare** The ALT bit is used to configure Timer 1 for dual or single maximum count mode. 0 = When the timer is enabled, the timer counts to the Timer 1 Maxcount Compare A register value, then resets the Timer 1 Count register value to 0. In this case, the Timer 1 Maxcount Compare B register is not used. In this mode, the TMROUT1 pin is High while the counter is counting and being compared to the Maxcount Compare A register. The TMROUT1 pin is pulsed Low for a single CPU clock cycle after the maximum value is reached. 1 = When the timer is enabled, the timer counts to the Timer 1 Maxcount Compare A register value, then resets the Timer 1 Count register value to 0. The timer then counts to the Timer 1 Maxcount Compare B register value and resets the Timer 1 Count register value to 0. This is referred to as “dual maximum count mode”. In this mode, the TMROUT1 pin is High while the counter is counting and being compared to the Maxcount Compare A register. The TMROUT1 pin is Low while the counter is counting and being compared to the Maxcount Compare B register. In other words, the TMROUT1 pin is the inverse of the RIU bit. If external clocking is used and the clock is stopped during a count sequence, the timer output remains in its previous state (i.e., the state that it was in prior to the clock stopping). The remaining timer status also remains the same, and normal operation commences upon the external clock being driven again. See the CONT bit description for a more detailed description of how the comparison registers are used in Continuous and Non-Continuous modes.
0	CONT	**Continuous Mode** The CONT bit is used to configure Timer 1 for Continuous or Non-Continuous mode. 0 = Timer 1 runs in Non-Continuous mode. The Timer 1 Count register is cleared and the timer halts whenever the count value reaches the maximum count value. The EN bit is also cleared by hardware after every counter sequence. If the CONT bit is cleared and the ALT bit is set, Timer 1 counts to the Timer 1 Maxcount Compare A register value, then resets the count value. After the timer count has been reset, the timer continues operation by counting to the Timer 1 Maxcount Compare B register value. When the timer count value reaches the Maxcount Compare B register value, it resets its count value, clears the EN bit, and then halts. 1 = Timer 1 runs in Continuous mode. Continuous mode causes the Timer Count register to be reset to 0, after it reaches the Maxcount Compare A or B register value. The timer immediately begins timing out again.

Programming Notes

Timers 0 and 1 are used by the Am186CC/CH/CU microcontrollers’s pulse-width demodulation (PWD) feature when the PWD bit is set in the System Configuration (SYSCON) register (see page 16-2).

Timer 1 Count (T1CNT) Offset 034Ah

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	TC															
Software Read/Write	R/W															
Hardware Set/Clear	S/C															
Chip Reset Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Microcontroller:

Am186CC ✓
Am186CH ✓
Am186CU ✓

Register Description

The T1CNT register is incremented by 1 each time Timer 1 advances its count. This count register is compared with the Timer 1 Maxcount Compare A or B registers, and various actions are taken when the maximum count is reached.

Bit Definitions

Bit	Name	Function
15–0	TC	**Timer 1 Count** The TC bit field contains the current count of Timer 1. The count is incremented every fourth processor clock cycle when the timer is in internal clocked mode, or each time the Timer 2 maxcount is reached if Timer 1 is configured to be prescaled by Timer 2. Timer 1 can also be configured for external clocking control based on the TMRIN1 signal. The TC bit field can be read at any time to determine the remaining count duration until a maximum count value is reached. The TC bit field can be written at any time. If the TC bit field is written while the counter is enabled, the new value is latched into the Timer 1 counter and counting proceeds from the new value.

Programming Notes

Timer 1 Maxcount Compare A (T1CMPA) Offset 034Ch

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	TC															
Software Read/Write	R/W															
Hardware Set/Clear	—															
Chip Reset Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Microcontroller:	
Am186CC	✔
Am186CH	✔
Am186CU	✔

Register Description

The T1CMPA register contains one of two compare values for the Timer 1 Count register. (Timer 1 and Timer 0 each have two Maxcount Compare registers.) If a Maxcount Compare register is written with the value 0000h and the timer is enabled, the timer counts to FFFFh, at which point the appropriate action occurs based on the timer configuration options that are set.

Bit Definitions

Bit	Name	Function
15–0	TC	**Timer 1 Maxcount Compare A** The TC bit field contains one of the maximum values that Timer 1 can count to before resetting its count register to 0.

Programming Notes

Timer 1 Maxcount Compare B (T1CMPB) Offset 034Eh

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	TC															
Software Read/Write	R/W															
Hardware Set/Clear	—															
Chip Reset Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Microcontroller:	
Am186CC	✔
Am186CH	✔
Am186CU	✔

Register Description

The T1CMPB register contains one of two compare values for the Timer 1 Count register. (Timer 1 and Timer 0 each have two Maxcount Compare registers.) If a Maxcount Compare register is written with the value 0000h and the timer is enabled, the timer counts to FFFFh, at which point the appropriate action occurs based on the timer configuration options that are set.

Bit Definitions

Bit	Name	Function
15–0	TC	**Timer 1 Maxcount Compare B** The TC bit field contains one of the maximum values that Timer 1 can count to before resetting its count register to 0.

Programming Notes

Timer 2 Mode/Control (T2CON) Offset 0350h

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	EN	$\overline{\text{INH}}$	INT	*Res*							MC	*Res*				CONT
Software Read/Write	R/W	W	R/W	R							R/W0	R				R/W
Hardware Set/Clear	C	—	—	—							S	—				—
Chip Reset Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Microcontroller:	
Am186CC	✔
Am186CH	✔
Am186CU	✔

Register Description

The T2CON register is used to control the functionality and the modes of operation of Timer 2.

Bit Definitions

Bit	Name	Function
15	EN	**Enable** 0 = Timer 2 is inhibited from counting. 1 = Timer 2 is enabled. The EN bit can only be modified (set or cleared) via software if the $\overline{\text{INH}}$ bit of this register is set (i.e., written as 1) during the same write cycle access to this register. The EN bit is automatically cleared by hardware at the end of a count sequence if Non-Continuous mode is selected. See the CONT bit description on page 10-15.
14	$\overline{\text{INH}}$	**Inhibit** The $\overline{\text{INH}}$ bit allows selective software modifications of the EN bit. When the $\overline{\text{INH}}$ bit is set during a write cycle access to this register, the EN bit can be modified in that same write cycle. When the INH bit is not written as 1 during a write cycle access to this register, the EN bit cannot be modified. The INH bit is always read back as a 0.
13	INT	**Interrupt** The INT bit allows the timer to generate an interrupt when the timer count value reaches the maximum count compare register value. 0 = The timer does not cause the TIM2 bit in the INTSTS register to be set (see page 9-42); therefore, a timer interrupt is not generated. 1 = An interrupt is generated (i.e., the TIM2 bit in the INTSTS register is set and latched) when the Timer 2 Count register equals the value of the Timer 2 Maxcount Compare A register. In addition to setting the INT bit in this register, software must configure the interrupt Channel 0 Control (CH0CON) register to enable the interrupt channel before any timer interrupts can be generated. See Chapter 9, “Interrupt Controller Registers”.
12–6	Res	**Reserved** For compatibility with future devices, always write this bit field with its chip reset default value.
5	MC	**Maximum Count** The MC bit can be used by software to determine when the timer reaches its maximum count. 0 = The maximum count was not reached, or software has cleared the MC bit. The MC bit is never automatically cleared by hardware. Software must clear the MC bit. 1 = Hardware sets the MC bit anytime the timer count value reaches its maximum count value. The MC bit is set for this condition regardless of the state of the INT bit. The MC bit can be used to monitor timer status through software polling instead of making use of interrupt generation.

Bit	Name	Function
4–1	Res	**Reserved** For compatibility with future devices, always write this bit field with its chip reset default value.
0	CONT	**Continuous Mode** The CONT bit is used to configure Timer 2 for Continuous or Non-Continuous mode. 0 = Timer 2 runs in Non-Continuous mode. The Timer 2 Count register is cleared and the timer halts whenever the count value reaches the maximum count value. The EN bit is also cleared by hardware after every counter sequence. 1 = Timer 2 runs in Continuous mode. Continuous mode causes the Timer count register to be reset to 0 after it reaches the Maxcount Compare A register value. The timer immediately begins timing out again.

Programming Notes

Timer 2 Count (T2CNT) Offset 0352h

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	TC															
Software Read/Write	R/W															
Hardware Set/Clear	S/C															
Chip Reset Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Microcontroller:	
Am186CC	✓
Am186CH	✓
Am186CU	✓

Register Description

The T2CNT register is incremented by 1 each time Timer 2 advances its count. This count register is compared with the Timer 2 Maxcount Compare A register, and various actions are taken when the maximum count is reached.

Bit Definitions

Bit	Name	Function
15–0	TC	**Timer 2 Count** The TC bit field contains the current count of Timer 2. The count is incremented every fourth processor clock cycle. The TC bit field can be read at any time to determine the remaining count duration until a maximum count value is reached. The TC bit field can be written at any time. If the TC bit field is written while the counter is enabled, the new value is latched into the Timer 2 counter and counting proceeds from the new value.

Programming Notes

Timer 2 Maxcount Compare A (T2CMPA) Offset 0354h

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	TC															
Software Read/Write	R/W															
Hardware Set/Clear	—															
Chip Reset Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Microcontroller:	
Am186CC	✔
Am186CH	✔
Am186CU	✔

Register Description

The T2CMPA register contains the single compare value for the Timer 2 Count register. If this Maxcount Compare register is written with the value 0000h and the timer is enabled, the timer counts to FFFFh, at which point the appropriate action occurs based on the timer configuration options that are set.

Bit Definitions

Bit	Name	Function
15–0	TC	**Timer 2 Maxcount Compare A** The TC bit field contains the maximum value that Timer 2 can count to before resetting its count register to 0.

Programming Notes

CHAPTER

11 CHIP SELECT REGISTERS

11.1 OVERVIEW

This chapter describes the Chip Select registers on the Am186CC/CH/CU microcontrollers.

The Am186CC/CH/CU microcontrollers provide programmable chip select generation for both memory devices and peripherals. In addition, the logic can be programmed to provide ready or wait-state generation.

The Upper Memory Chip Select (UMCS) and Lower Memory Chip Select (LMCS) registers can be configured to gluelessly support DRAM devices. If DRAM is enabled for the upper or lower memory spaces, the DRAM controller refresh registers must also be configured (see Chapter 12, “DRAM Controller Registers”).

Except for the $\overline{\text{UCS}}$ signal, which is active on external or watchdog timer reset, chip selects are not activated until the associated registers have been accessed by a write operation. For details, see the *Am186CC/CH/CU Microcontrollers User’s Manual*, order #21914.

Table 11-1 lists the Chip Select registers in offset order, with the corresponding description’s page number.

Table 11-1 Chip Select Register Map

Register Name	Mnemonic	Offset	Page Number
Upper Memory Chip Select	UMCS	3A0h	page 11-2
Lower Memory Chip Select	LMCS	3A2h	page 11-5
Peripheral Chip Select	PACS	3A4h	page 11-9
Midrange Memory Chip Select	MMCS	3A6h	page 11-11
PCS and MCS Auxiliary	MPCS	3A8h	page 11-13

Upper Memory Chip Select (UMCS) Offset 3A0h

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	*Res*	LB			*Res*				DA	UDEN	USIZ	*Res*		R2	R1	R0
Software Read/Write	R	R/W			R				R/W	R/W	R/W	R		R/W	R/W	R/W
Hardware Set/Clear	—	—			—				—	—	—	—		—	—	—
Chip Reset Default	1	1	1	1	0	0	0	0	0	0	?	1	1	0	1	1

Microcontroller:
Am186CC ✓
Am186CH ✓
Am186CU ✓

Register Description

The UMCS register is used to configure the memory chip select that is associated with the upper regions of the processor’s 1-Mbyte linear address space (i.e., upper memory). The $\overline{\text{UCS}}$ (Upper Memory Chip Select) output pin is active for memory accesses to this memory region. This chip select is enabled immediately after an external or watchdog timer reset. The UMCS control register is used to configure the size (i.e., memory address range), AD bus options, DRAM option, data bus size option, and wait state control for this chip select.

Bit Definitions

Bit **Name** **Function**

15 Res **Reserved**
For compatibility with future devices, always write this bit field with its chip reset default value.

14–12 LB **Lower Boundary**
The LB bit field defines the lowest memory address that can cause the $\overline{\text{UCS}}$ pin (or the $\overline{\text{RAS1}}$ pin if UDEN is set) to be asserted. $\overline{\text{UCS}}$ memory block sizes of 64K, 128K, 256K, or 512K can be selected. The highest memory address that can cause the $\overline{\text{UCS}}$ pin to be asserted is always FFFFFh. The table below shows the LB bit encoding for the $\overline{\text{UCS}}$ memory space.

Memory Block Size	UCS Starting Address	LB
64K	F0000h	111b (Default)
128K	E0000h	110b
256K	C0000h	100b
512K	80000h	000b

11–8 Res **Reserved**
For compatibility with future devices, always write this bit field with its chip reset default value.

Bit **Name** **Function**

7 DA **Disable Address**

The DA bit enables or disables the AD15–AD0 bus during the address phase of a bus cycle when the UCS signal (or RAS1 if UDEN is set) is asserted.

0 = The AD bus is driven during the address phase of a UCS bus cycle unless memory addresses are disabled globally (DISMEM = 1 in the SYSCON register, see page 16-2).

1 = The AD bus is not driven during the address phase of a UCS bus cycle.

The DA bit has no effect if the ADEN input pin is sampled in its Low state during an external reset (i.e., when the RES input pin transitions from Low to High). Otherwise, the DA bit functions as described above for the UCS memory space.

The DA bit functions as described above regardless of the state of the UDEN bit. In addition, if the DA bit is set and UCS space has been configured for DRAM, an access to a valid PCS memory region that overlaps the UCS space does not cause the address phase of the AD bus to be disabled unless memory addresses are disabled globally via the DISMEM bit in the SYSCON register (see page 16-2).

6 UDEN **UCS DRAM Enable**

The UDEN bit selects the UCS space as a DRAM bank.

0 = UCS memory functions as a non-DRAM bank.

1 = UCS DRAM is enabled: the MCS3 pin becomes RAS1, and the MCS1 and MCS2 pins become CAS1 and CAS0, respectively. The UCS pin is also forced to its High state.

Immediately following an external or watchdog timer reset, the UCS memory region functions as a non-DRAM bank, allowing a system to boot from a non-volatile memory device prior to software switching the upper memory region to a DRAM bank via setting the UDEN bit.

The UCS block size, DA bit, and the R1–R0 bits all function the same regardless of the state of the UDEN bit. The R2 bit is a “don’t care” when DRAM is enabled because the external readies are forced to be ignored for UCS memory cycles when the UCS space is configured for DRAM. In addition, the USIZ bit is a “don’t care” when DRAM is enabled.

The UCS bus size is forced to be x16 when the UDEN bit is set (i.e., x8 DRAM interface is not supported). If a PCS memory space is overlapped with a (DRAM-mapped) UCS space, the PCS space is accessed as x16 memory.

5 USIZ **Upper Memory Data Bus Size**

The USIZ bit determines the width of the data bus (i.e., x8 or x16) for memory accesses to the upper memory region.

0 = The data bus width for these accesses is configured to be x16.

1 = The data bus width for these accesses is configured to be x8.

Immediately after an external reset (i.e., the RES input pin transitions from Low to High), the USIZ bit reflects the inverted state of the UCSX8 pin that was latched upon exiting reset. This relationship is shown in the following table:

UCSX8 Pin Reset State	USIZ State	USIZ R/W	Description
0	1	R/W	UCS is configured to be x8 at boot time and can be overridden by clearing this bit.
1	0	R	UCS is configured to be x16 at boot time and cannot be overridden.

The UCSX8 pin is not sampled again during an internal (watchdog timer) reset, so the watchdog timer has no effect on the USIZ bit state.

Software must not modify the USIZ bit while executing from the UCS space.

If the UDEN bit is 1, the USIZ bit becomes a “don’t care” and the UCS bus size is forced to be x16 (i.e., x8 DRAM interface is not supported).

4–3 Res **Reserved**

For compatibility with future devices, always write this bit field with its chip reset default value.

Bit **Name** **Function**

2 R2 **Ready Mode**

The R2 bit is used to configure the ready mode for the $\overline{\text{UCS}}$ chip select.

0 = The external ready inputs, in addition to the R1 and R0 bits, are used to control the number of wait states for each bus cycle that is directed to the $\overline{\text{UCS}}$ memory region. In this configuration, the R1–R0 bits define the minimum number of wait states that are added to the UMCS bus cycle. Additional wait states may or may not be added via the external ready inputs, depending on their assertion or deassertion relative to the bus cycle phase. (See the SRDY and ARDY timing diagrams in the Am186CC/CH/CU microcontroller data sheets for further information regarding the external ready inputs.)

1 = The states of the external ready input pins (SRDY and ARDY) are ignored during a bus cycle in which the $\overline{\text{UCS}}$ signal is active. In this case, the R1 and R0 bits alone are used to configure the number of wait states for memory accesses to the $\overline{\text{UCS}}$ region.

The R2 bit is a “don’t care” if the UDEN bit is 1 (i.e., external ready signals are ignored when the UDEN bit is set). When DRAM is enabled for the upper memory region, the R1–R0 wait state control bits can be used to control the bus cycle access time during a DRAM access.

1–0 R1–R0 **Wait State Value**

The R1 and R0 bits are used to control the minimum number of wait states for a memory cycle directed to the $\overline{\text{UCS}}$ memory region. Additional wait states can be added via the SRDY and ARDY signals if the R2 bit is 0.

The value of this two-bit field determines the minimum number of wait states added (i.e., zero to three wait states can be added). The number of wait states added to a bus cycle that are selected via configuring the R1–R0 bits are present regardless of the state of the R2 bit.

The table below shows how bits R1–R0 are decoded.

R1	R0	Added Wait States
0	0	0
0	1	1
1	0	2
1	1	3 (Default)

Programming Notes

Lower Memory Chip Select (LMCS) Offset 3A2h

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	*Res*	UB			*Res*				DA	LDEN	LSIZ	*Res*		R2	R1	R0
Software Read/Write	R	R/W			R				R/W	R/W	R/W	R		R/W	R/W	R/W
Hardware Set/Clear	—	—			—				—	—	—	—		—	—	—
Chip Reset Default	0	0	0	0	1	1	1	1	0	0	0	1	1	0	1	1

Microcontroller:
Am186CC ✔
Am186CH ✔
Am186CU ✔

Register Description

The LMCS register is used to configure the memory chip select that is associated with the lower region of the processor's 1-Mbyte linear address space (i.e., lower memory). The $\overline{\text{LCS}}$ (Lower Memory Chip Select) output pin is active for memory accesses to this memory region. This chip select is not enabled immediately after an external or watchdog timer reset. This chip select can only be activated by system software performing a write operation to the LMCS register. This control register is used to configure the size (i.e., memory address range), AD bus options, DRAM option, data bus size option, and wait state control for this chip select.

Bit Definitions

Bit	Name	Function
15	Res	**Reserved** For compatibility with future devices, always write this bit field with its chip reset default value.
14–12	UB	**Upper Boundary** The UB bits define the highest memory address that can cause the $\overline{\text{LCS}}$ pin (or the $\overline{\text{RAS0}}$ pin if LDEN is set) to be asserted. $\overline{\text{LCS}}$ memory block sizes of 64K, 128K, 256K, or 512K can be selected. The lowest memory address that can cause the $\overline{\text{LCS}}$ pin to be asserted is always 00000h. The table below shows the UB bit encoding for the $\overline{\text{LCS}}$ memory space.

Memory Block Size	LCS Ending Address	UB
64K	0FFFFh	000b (Default)
128K	1FFFFh	001b
256K	3FFFFh	011b
512K	7FFFFh	111b

Bit	Name	Function
11–8	Res	**Reserved** For compatibility with future devices, always write this bit field with its chip reset default value.

Bit	Name	Function
7	DA	**Disable Address** The DA bit enables or disables the AD15–AD0 bus during the address phase of a bus cycle when the LCS signal (or RAS0 if LDEN is set) is asserted. 0 = The AD bus is driven during the address phase of a LCS bus cycle unless memory addresses are disabled globally (DISMEM = 1 in the SYSCON register, see page 16-2). 1 = The AD bus is not driven during the address phase of a LCS bus cycle. The DA bit has no effect if the ADEN input pin is sampled in its Low state during an external reset (i.e., when the RES input pin transitions from Low to High). Otherwise, the DA bit functions as described above for the LCS memory space. The DA bit functions as described above regardless of the state of the LDEN bit. In addition, if the DA bit is set and LCS space has been configured for DRAM, an access to a valid PCS memory region that overlaps the LCS space does not cause the address phase of the AD bus to be disabled unless memory addresses are disabled globally via the DISMEM bit in the SYSCON register (see page 16-2).
6	LDEN	**LCS DRAM Enable** The LDEN bit selects the LCS space as a DRAM bank. 0 = LCS memory functions as a non-DRAM bank. 1 = LCS DRAM is enabled: the LCS pin becomes RAS0, and the MCS1 and MCS2 pins become CAS1 and CAS0, respectively. The LCS block size, DA bit, and the R1–R0 bits all function the same regardless of the state of the LDEN bit. The R2 bit is a “don’t care” when DRAM is enabled because the external readies are forced to be ignored for LCS memory cycles when the LCS space is configured for DRAM. In addition, the LSIZ bit is a “don’t care” when DRAM is enabled. The LCS bus size is forced to be x16 when the LDEN bit is set (i.e., x8 DRAM interface is not supported). If a PCS memory space is overlapped with a (DRAM-mapped) LCS space, the PCS space is accessed as x16 memory.
5	LSIZ	**Lower Memory Data Bus Size** The LSIZ bit determines the width of the data bus (i.e., x8 or x16) for memory accesses to the lower memory region. 0 = The data bus width for these accesses is configured to be x16. 1 = The data bus width for these accesses is configured to be x8. Software must not modify this bit while executing from the LCS space. If the LDEN bit is 1, the LSIZ bit becomes a “don’t care” and the LCS bus size is forced to be x16 (i.e., x8 DRAM interface is not supported).
4–3	Res	**Reserved** For compatibility with future devices, always write this bit field with its chip reset default value.

Bit **Name** **Function**

2 R2 **Ready Mode**

The R2 bit is used to configure the ready mode for the $\overline{\text{LCS}}$ chip select.

0 = The external ready inputs, in addition to the R1 and R0 bits, are used to control the number of wait states for each bus cycle that is directed to the $\overline{\text{LCS}}$ memory region. In this configuration, the R1–R0 bits define the minimum number of wait states that are added to the LMCS bus cycle. Additional wait states may or may not be added via the external ready inputs, depending on their assertion or deassertion relative to the bus cycle phase. (See the SRDY and ARDY timing diagrams in the Am186CC/CH/CU microcontroller data sheets for further information regarding the external ready inputs.)

1 = The states of the external ready input pins (SRDY and ARDY) are ignored during a bus cycle in which the $\overline{\text{LCS}}$ signal is active. In this case, the R1 and R0 bits alone are used to configure the number of wait states for memory accesses to the $\overline{\text{LCS}}$ region.

This bit is a “don’t care” if LDEN = 1 (i.e., external ready’s are ignored when the LDEN bit is set). When DRAM is enabled for this memory region, the R1–R0 wait state control bits can be used to control the bus cycle access time during a DRAM access.

1–0 R1–R0 **Wait State Value**

The R1–R0 bits are used to control the minimum number of wait states for a memory cycle directed to the $\overline{\text{LCS}}$ memory region. Additional wait states can be added via the SRDY and ARDY signals if the R2 bit is 0.

The value of this two-bit field determines the minimum number of wait states added (i.e., zero to three wait states can be added). The number of wait states added to a bus cycle that are selected via configuring the R1–R0 bits are present regardless of the state of the R2 bit.

The table below shows how bits R1–R0 are decoded.

R1	R0	Added Wait States
0	0	0
0	1	1
1	0	2
1	1	3 (Default)

Programming Notes

Peripheral Chip Select (PACS) Offset 3A4h

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	BA[19–11]									*Res*			R3	R2	R1	R0
Software Read/Write	R/W									R			R/W	R/W	R/W	R/W
Hardware Set/Clear	C				—					—			—	—	—	—
Chip Reset Default	0	0	0	0	0	0	0	0	0	1	1	1	0	0	1	1

Microcontroller:	
Am186CC	✓
Am186CH	✓
Am186CU	✓

Register Description

The PACS register is used to partially configure the peripheral chip selects (i.e., $\overline{\text{PCS7}}$–$\overline{\text{PCS0}}$). The PCS and MPCS Auxiliary (MPCS) register is also used to configure these chip selects (see page 11-13). The $\overline{\text{PCS}}$ chip selects are not enabled immediately after an external or watchdog timer reset. The $\overline{\text{PCS}}$ chip selects can only be activated by system software performing a write operation to both the PACS register and the MPCS register.

Bit Definitions

Bit	Name	Function

15–7 BA[19–11]

Base Address

The BA[19–11] bits define the base address of the memory or I/O region that is associated with the peripheral chip selects ($\overline{\text{PCS7}}$–$\overline{\text{PCS0}}$). These bits correspond to the bits A19 through A11 of the CPU's 20-bit address. If the peripheral chip selects are programmed to reside in the CPU's I/O space, bits BA[19–16] are forced to be 0 by hardware, as the upper bound of the CPU's I/O space is 64K.

When the peripheral chip selects are enabled, each of the individual chip selects is associated with a 256-byte address region. These eight 256-byte address regions are contiguous in memory. The table below shows the $\overline{\text{PCS}}$ mapping.

PCS Signal	Low Address Range	High Address Range
$\overline{\text{PCS0}}$	Base Address	Base Address + 255
$\overline{\text{PCS1}}$	Base Address + 256	Base Address + 511
$\overline{\text{PCS2}}$	Base Address + 512	Base Address + 767
$\overline{\text{PCS3}}$	Base Address + 768	Base Address + 1023
$\overline{\text{PCS4}}$	Base Address + 1024	Base Address + 1279
$\overline{\text{PCS5}}$	Base Address + 1280	Base Address + 1535
$\overline{\text{PCS6}}$	Base Address + 1536	Base Address + 1791
$\overline{\text{PCS7}}$	Base Address + 1792	Base Address + 2047

The PCS base address can be located anywhere in the 1-Mbyte memory address space as long as it is a multiple of 2 Kbyte (0 is a valid multiple) and the memory space is not already mapped to by the $\overline{\text{LCS}}$, $\overline{\text{UCS}}$, (if $\overline{\text{LCS}}$ and $\overline{\text{UCS}}$ are not configured for DRAM), or $\overline{\text{MCS}}$ chip selects.

When the $\overline{\text{LCS}}$ or $\overline{\text{UCS}}$ memory regions are configured for DRAM, the $\overline{\text{PCS}}$ region (assuming that PCS is enabled and configured for memory) can overlap with the $\overline{\text{LCS}}$ or $\overline{\text{UCS}}$ memory region. In this case, the $\overline{\text{PCS}}$ bus width is x16, and the access takes precedence over the $\overline{\text{LCS}}$ or $\overline{\text{UCS}}$ access.

6–4 Res

Reserved

For compatibility with future devices, always write this bit field with its chip reset default value.

Bit **Name** **Function**

3 R3 **Wait State Value**

The R3 bit is used with bits R1–R0 to configure the minimum number of wait states for the PCS3–PCS0 peripheral chip selects. (PCS7–PCS4 wait states are configured in the MPCS register, see page 11-14.)

0 = The number of wait states added (zero to three wait states) is the two-bit value contained in the R1 and R0 bits of the PACS register.

1 = The number of wait states added (5 to 15 wait states) is encoded in the R1 and R0 bits of the PACS register.

See the R1–R0 bit description for a table listing the different wait states that can be added to bus cycles when the PCS3–PCS0 signals are active.

2 R2 **Ready Mode**

The R2 bit is used to configure the ready mode for the PCS3–PCS0 peripheral chip selects. (PCS7–PCS4 ready mode is configured in the MPCS register, see page 11-15.)

0 = The external ready inputs, in addition to the R3, R1, and R0 bits, are used to control the number of wait states for each bus cycle that is directed to the PCS3–PCS0 memory region. In this configuration, the R3, R1, and R0 bits define the minimum number of wait states that are added to the PCS3–PCS0 bus cycle. Additional wait states may or may not be added via the external ready inputs, depending on their assertion or deassertion relative to the bus cycle phase. (See the SRDY and ARDY timing diagrams in the Am186CC/CH/CU microcontroller data sheets for further information regarding the external ready inputs.)

1 = The states of the external ready input pins (SRDY and ARDY) are ignored during a bus cycle in which the PCS3–PCS0 signals are active. In this case, the R3, R1, and R0 bits alone are used to configure the number of wait states for memory accesses to the PCS3–PCS0 memory region.

1–0 R1–R0 **Wait State Value**

The R1–R0 bits are used with the R3 bit to configure the minimum number of wait states for the PCS3–PCS0 peripheral chip selects. Additional wait states can be added via the SRDY and ARDY signals if the R2 bit is 0.

The value of this two-bit field, along with the value of the R3 bit determine the number of wait states added (i.e., 0 to 15 wait states can be added). The number of wait states added to a bus cycle that are selected via configuring these three bits are present regardless of the state of the R2 bit.

The table below shows how bits R3 and R1–R0 are decoded.

R3	R1	R0	Added Wait States
0	0	0	0
0	0	1	1
0	1	0	2
0	1	1	3 (Default)
1	0	0	5
1	0	1	7
1	1	0	9
1	1	1	15

Note: *If a PCS chip select is programmed to overlap a DRAM-mapped UCS or LCS memory space, it is illegal to program the PCS space with fewer wait states than the DRAM that is overlapped.*

Programming Notes

Midrange Memory Chip Select (MMCS) Offset 3A6h

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	BA[19–13]							*Res*			MCS0_ONLY	*Res*		R2	R1	R0
Software Read/Write	R/W							R			R/W	R		R/W	R/W	R/W
Hardware Set/Clear	—							—			—	—		—	—	—
Chip Reset Default	0	1	1	1	1	1	1	1	1	1	0	1	1	0	1	1

Microcontroller:

Am186CC	✔
Am186CH	✔
Am186CU	✔

Register Description

The MMCS register is used to partially configure the midrange memory chip selects (i.e., MCS3–MCS0). The PCS and MCS Auxiliary (MPCS) register is also used to configure these chip selects (see page 11-13). The MMCS register is used to configure the wait state control and the base address for the starting location, in the 1-Mbyte linear address range, of the CPU memory region. This base address can be located at any multiple of the block size of the MCS memory region, but the user must ensure that the MCS address range does not overlap any other memory regions such as the UCS, LCS, or PCS memory spaces. Note that PCB memory space can overlap with any other memory space as long as that other memory space is programmed to ignore external ready and is set up for zero wait states internally generated.

The midrange memory chip selects are not enabled immediately after an external or watchdog timer reset. These chip selects can only be activated by system software performing a write operation to both the MMCS register and the MPCS register.

If the midrange memory chip selects are enabled, and the MCS0_ONLY bit is cleared, each individual MCS chip select is active for one fourth of the total block size of the MCS memory region. If the MCS0ONLY bit is set, the MCS0 pin is active for the entire block size of the MCS memory region. The MCS block size is configured in the MPCS register. The MCS base address and wait state configuration are common to all four of the MCS pins.

Bit Definitions

Bit	Name	Function
15–9	BA[19–13]	**Base Address** The BA[19–13] bits define the base address of the memory region that is associated with the midrange memory chip selects (MCS3–MCS0). These bits correspond to the bits A19 through A13 of the CPU's 20-bit memory address. The base address can be configured to be any multiple of the block size (including 0) of the MCS memory region as configured in the PCS and MCS Auxiliary register. For example, if the midrange block size is 32 Kbytes, the MCS memory region base address could be located at 00000h, 10000h, or 18000h, or C8000h, but not at 14000h. The BA[19–13] bit field default value is 3Fh (i.e., the default base address is 3F000h). The MCS memory region should not overlap with any other memory region in the system. One exception is the PCB region. If the PCB region is configured to reside in memory, it can be overlapped with MCS only if MCS is configured as not using external ready and all internal wait states are disabled. PCB accesses always take precedence over the other memory region that they overlap.
8–6	Res	**Reserved** For compatibility with future devices, always write this bit field with its chip reset default value.

Bit	Name	Function
5	MCS0_ONLY	**MCS0 Only Mode** The MCS0_ONLY bit controls the MCS0 Only mode. 0 = When this bit is cleared and the MCS chip selects are enabled, the MCS3–MCS0 signals are each asserted over one fourth of the total MCS block size. 1 = When this bit is set and the MCS chip selects are enabled, MCS0 is asserted over the entire MCS address range (MCS3 to MCS0 inclusive). MCS3–MCS1 are still asserted over their individual address ranges. This mode is useful if DRAM is enabled in the UMCS or LMCS register (see page 11-3 and page 11-6). The MCS3–MCS1 pins are multiplexed and can be configured at any time for alternative functions such as the UCS/LCS DRAM interface or as programmable I/O pins. If these signals are configured for their alternate functions, they no longer operate as Midrange Memory Chip Selects.
4–3	Res	**Reserved** For compatibility with future devices, always write this bit field with its chip reset default value.
2	R2	**Ready Mode** The R2 bit is used to configure the ready mode for the MCS3–MCS0 chip selects. 0 = The external ready inputs, in addition to the R1 and R0 bits, are used to control the number of wait states for each bus cycle that is directed to the MCS3–MCS0 memory region. In this configuration, the R1–R0 bits define the minimum number of wait states that are added to the MCS bus cycle. Additional wait states may or may not be added via the external ready inputs, depending on their assertion or deassertion relative to the bus cycle phase. (See the SRDY and ARDY timing diagrams in the Am186CC/CH/CU microcontroller data sheets for further information regarding the external ready inputs.) 1 = The states of the external ready input pins (SRDY and ARDY) are ignored during a bus cycle in which the MCS signals are active. In this case, the R1 and the R0 bits are used to control the number of wait states for memory accesses to the MCS region.
1–0	R1–R0	**Wait State Value** The R1–R0 bits are used to control the minimum number of wait states for a memory cycle directed to the MCS memory region. Additional wait states can be added via the SRDY and ARDY signals if the R2 bit is 0. The binary value of this two-bit field determines the number of wait states added (i.e., zero to three wait states can be added). The number of wait states added to a bus cycle that are selected via configuring the R1–R0 bits are present regardless of the state of the R2 bit. The table below shows how bits R1–R0 are decoded.

R1	R0	Added Wait States
0	0	0
0	1	1
1	0	2
1	1	3 (Default)

Programming Notes

PCS and MCS Auxiliary (MPCS) Offset 3A8h

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	*Res*	M[6–0]							*Res*	MS	OMSIZ	IOSIZ	R3	R2	R1	R0
Software Read/Write	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Hardware Set/Clear	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Chip Reset Default	1	0	0	0	0	0	0	1	1	0	0	0	0	0	1	1

Microcontroller:
Am186CC ✔
Am186CH ✔
Am186CU ✔

Register Description

This register is used to partially configure both the chip selects associated with the midrange memory region (i.e., MCS3–MCS0) and the peripheral chip selects (i.e., PCS7–PCS0). The MCS and PCS chip selects are not enabled immediately after an external or watchdog timer reset. The MCS chip selects can only be activated by system software performing a write operation to both the MPCS register and the MMCS register (see page 11-11). The PCS chip selects can only be activated by system software performing a write operation to both the MPCS register and the PACS register (see page 11-9).

Bit Definitions

Bit **Name** **Function**

15 Res **Reserved**
For compatibility with future devices, always write this bit field with its chip reset default value.

14–8 M[6–0] **MCS Block Size**
The M[6–0] bit field determines the total block size for the MCS3–MCS0 chip selects. Each individual MCS I/O pin is active for one fourth of the total block size when the MCS0_ONLY bit in the MMCS register is cleared.

Only one of the bits in the M[6–0] bit field can be set at a time. If this is not followed, unpredictable behavior of the MCS signals occurs.

The encoding of this bit field is shown in the table below.

Total Block Size	Individual Chip Select Size	M[6–0] Bit Field
8K	2K	0000001b (default)
16K	4K	0000010b
32K	8K	0000100b
64K	16K	0001000b
128K	32K	0010000b
256K	64K	0100000b
512K	128K	1000000b

7 Res **Reserved**
For compatibility with future devices, always write this bit field with its chip reset default value.

6 MS **Memory or I/O Space Selector**
The MS bit determines whether the PCS chip selects are mapped to I/O or memory space.

0 = The PCS output signals are active for I/O bus cycles.

1 = The PCS output signals are active for memory bus cycles.

Bit	Name	Function
5	OMSIZ	**Non-UCS and Non-LCS Memory (Other Memory) Data Bus Size** The OMSIZ bit determines the width of the data bus (i.e., x8 or x16) for memory accesses between the LCS and UCS memory regions (i.e., accesses above the LCS region and below the UCS region). An MCS space can only be configured to lie in the space affected by the OMSIZ bit. A PCS space is only affected by the OMSIZ bit if the PCS space is configured to reside in memory space and does not overlap a DRAM-mapped LCS or UCS region. (A PCS space can overlap only a DRAM-mapped LCS or UCS region.) If a PCS space is configured to overlap an LCS or UCS region, the PCS space is accessed as x16 memory. 0 =The data bus width for these accesses is configured to be x16. 1 =The data bus width for these accesses is configured to be x8. Software must not modify the OMSIZ bit while executing from the associated address regions. If the OMSIZ bit is set to 1 and the PCB space is mapped to memory (the M/IO bit is 1 in the RELOC register; see page 16-5), each PCB register access generates two external bus cycles, but all 16 register bits are accessed internally on the first cycle.
4	IOSIZ	**I/O Space Data Bus Size** The IOSIZ bit determines the width of the data bus (i.e., x8 or x16) for all I/O accesses. 0 =The data bus width for these accesses is configured to be x16. 1 =The data bus width for these accesses is configured to be x8. If the IOSIZ bit is set to 1 and the PCB space is mapped to I/O (the M/IO bit is 0 in the RELOC register; see page 16-5), each PCB register access generates two external bus cycles, but all 16 register bits are accessed internally on the first cycle.
3	R3	**Wait State Value** The R3 bit is used with bits R1–R0 to configure the minimum number of wait states for the PCS7–PCS4 peripheral chip selects only. (PCS3–PCS0 wait states are configured in the PACS register, see page 11-10.) 0 = The number of wait states added (zero to three wait states) is the two-bit value contained in the R1 and R0 bits of this register. 1 = The number of wait states added (5 to 15 wait states) is encoded in the R1 and R0 bits of this register. See the R1–R0 bit description for a table listing the different wait states that can be added to bus cycles when the PCS7–PCS4 signals are active.

Bit **Name** **Function**

2 R2 **Ready Mode**

The R2 bit is used to configure the ready mode for the PCS7–PCS4 peripheral chip selects only. (PCS3–PCS0 ready mode is configured in the PACS register, see page 11-10.)

0 = The external ready inputs, in addition to the R3, R1, and R0 bits, are used to control the number of wait states for each bus cycle that is directed to the PCS7–PCS4 memory region. In this configuration, the R3, R1, and R0 bits define the minimum number of wait states that are added to the PCS7–PCS4 bus cycle. Additional wait states may or may not be added via the external ready inputs, depending on their assertion or deassertion relative to the bus cycle phase. (See the SRDY and ARDY timing diagrams in the Am186CC/CH/CU microcontroller data sheets for further information regarding the external ready inputs.)

1 = The states of the external ready input pins (SRDY and ARDY) are ignored during a bus cycle in which the PCS7–PCS4 signals are active. In this case, the R3, R1, and R0 bits alone are used to configure the number of wait states for memory accesses to the PCS7–PCS4 memory region.

1–0 R1–R0 **Wait State Value**

The R1–R0 bits are used with the R3 bit to configure the minimum number of wait states for the PCS7–PCS4 peripheral chip selects. Additional wait states can be added via the SRDY and ARDY signals if the R2 bit is 0.

The value of this two-bit field along with the value of the R3 bit determine the number of wait states added (i.e., 0 to 15 wait states can be added). The number of wait states added to a bus cycle that are selected via configuring these three bits are present regardless of the state of the R2 bit.

The table below shows how bits R3 and R1–R0 are decoded.

R3	R1	R0	Added Wait States
0	0	0	0
0	0	1	1
0	1	0	2
0	1	1	3 (Default)
1	0	0	5
1	0	1	7
1	1	0	9
1	1	1	15

Note: *If a PCS chip select is programmed to overlap a DRAM-mapped UCS or LCS memory space, it is illegal to program the PCS space with fewer wait states than the DRAM that is overlapped.*

Programming Notes

CHAPTER

12 DRAM CONTROLLER REGISTERS

12.1 OVERVIEW

This chapter describes the Dynamic Random-Access Memory (DRAM) controller refresh registers on the Am186CC/CH/CU microcontrollers.

The Am186CC/CH/CU microcontrollers have a fully integrated DRAM controller. The Upper Memory Chip Select (UMCS) and Lower Memory Chip Select (LMCS) registers can be enabled to control DRAM devices (see page 11-3 and page 11-6). If DRAM is enabled for these memory spaces, the registers in this chapter must also be configured.

For more information about using the DRAM controller, see the *Am186CC/CH/CU Microcontrollers User's Manual*, order #21914.

Table 12-1 lists the DRAM registers in offset order, with the corresponding description's page number.

Table 12-1 DRAM Controller Register Map

Register Name	Mnemonic	Offset	Page Number
Refresh Clock Prescaler	CDRAM	3AAh	page 12-2
Enable Refresh Control	EDRAM	3ACh	page 12-3

Refresh Clock Prescaler (CDRAM) Offset 3AAh

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	*Res*			RC												
Software Read/Write	R			R/W												
Hardware Set/Clear	—			—												
Chip Reset Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Microcontroller:

Am186CC	✔
Am186CH	✔
Am186CU	✔

Register Description

This register is used to configure the DRAM refresh rate. The value programmed into the RC bit field determines the number of CPU clocks between refresh cycles when a DRAM interface is enabled in the $\overline{\text{UCS}}$ or $\overline{\text{LCS}}$ memory space.

Bit Definitions

Bit	Name	Function
15–13	Res	**Reserved** For compatibility with future devices, always write this bit field with its chip reset default value.
12–0	RC	**Refresh Controller Reload Value** The RC bit field is used to configure the DRAM refresh period (i.e., the time interval between DRAM refreshes). The value programmed into this bit field determines the number of CPU clocks between refresh cycles given by the formula below: $CPU_{clkper} \cdot (RC_{value}) = \text{Refresh Period}$ Where: CPU_{clkper} = CPU clock period RC_{value} = Decimal value of the RC bit field If RC_{value} is configured to be less than 18d, the CPU is not allowed enough bus cycles to adequately execute code. Changing the value of the RC bit field after refresh has been enabled (by setting the E bit of the EDRAM register) does not cause the new value to be loaded into the refresh counter until the current counter value has reached 0. At a CPU clock rate of 50MHz, CPU_{clkper} = 20ns. If the RC bit field is programmed to be 1FFFh, the Refresh period = 163.9µs (Maximum Refresh Period when the CPU clock rate = 50MHz).

Programming Notes

Enable Refresh Control (EDRAM) Offset 3ACh

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	EN	*Res*		T												
Software Read/Write	R/W	R		R												
Hardware Set/Clear	—	—		S/C												
Chip Reset Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Microcontroller:	
Am186CC	✓
Am186CH	✓
Am186CU	✓

Register Description

This register is used to enable the refresh counter. It can also be used to sample the present value of the refresh down counter.

Bit Definitions

Bit	Name	Function
15	EN	**Enable Refresh** The E bit is used to enable DRAM refresh. 0 = The refresh counter is cleared and disabled. 1 = The refresh counter is enabled.
14–13	Res	**Reserved** For compatibility with future devices, always write this bit field with its chip reset default value.
12–0	T	**Refresh Count** The T bit field contains the present value of the refresh counter. When this counter reaches 0, a refresh request is generated.

Programming Notes

CHAPTER

13 PROGRAMMABLE I/O (PIO) REGISTERS

13.1 OVERVIEW

This chapter describes the Programmable Input/Output (PIO) registers on the Am186CC/CH/CU microcontrollers.

The Am186CC/CH/CU microcontrollers provide 48 programmable input/output (PIO) pins. Each PIO pin is controlled by three register bits; one bit in each of three separate register types, Mode, Direction, and Data. Because each register contains only 16 bits, there are three of each type, for a total of nine PIO Mode, Direction, and Data registers. In addition, separate Set and Clear offsets are provided so software can easily set or clear individual PIO Data bits without affecting the other PIOs.

For more information about using Programmable I/O, see the *Am186CC/CH/CU Microcontrollers User's Manual*, order #21914, and either the *Am186™CC Communications Controller Data Sheet*, order #21915, the *Am186™CH HDLC Microcontroller Data Sheet*, order #22024, or the *Am186™CU USB Microcontroller Data Sheet*, order #22025.

Table 13-1 lists the PIO registers in offset order, with the corresponding description's page number. Table 13-2 on page 13-2 lists each PIO pin with alternate signals that are multiplexed on the pin, the pin's default signal function, and the PIOMODE register and bit that controls the PIO. (The same register and bit relationship applies to other PIO registers.)

Table 13-1 PIO Register Map

Register Name	Mnemonic	Offset	Page Number
PIO Mode 0	PIOMODE0	3C0h	page 13-4
PIO Direction 0	PIODIR0	3C2h	page 13-5
PIO Data 0	PIODATA0	3C4h	page 13-6
PIO Set 0	PIOSET0	3C6h	page 13-7
PIO Clear 0	PIOCLR0	3C8h	page 13-8
PIO Mode 1	PIOMODE1	3CAh	page 13-9
PIO Direction 1	PIODIR1	3CCh	page 13-10
PIO Data 1	PIODATA1	3CEh	page 13-11
PIO Set 1	PIOSET1	3D0h	page 13-12
PIO Clear 1	PIOCLR1	3D2h	page 13-13
PIO Mode 2	PIOMODE2	3D4h	page 13-14
PIO Direction 2	PIODIR2	3D6h	page 13-15
PIO Data 2	PIODATA2	3D8h	page 13-16
PIO Set 2	PIOSET2	3DAh	page 13-17
PIO Clear 2	PIOCLR2	3DCh	page 13-18

Table 13-2 PIO Multiplexed Signals

Signal	Multiplexed Signal(s)	Default Signal	Register[Bit No.]
PIO0	TMRIN1	PIO0: input with pullup	PIOMODE0[0]
PIO1	TMROUT1	PIO1: input with pulldown	PIOMODE0[1]
PIO2	$\overline{\mathrm{PCS5}}$	PIO2: input with pullup	PIOMODE0[2]
PIO3	$\overline{\mathrm{PCS4}}$	PIO3: input with pullup	PIOMODE0[3]
PIO4	$\overline{\mathrm{MCS0}}$	PIO4: input with pullup	PIOMODE0[4]
PIO5	$\overline{\mathrm{MCS3}}$ / $\overline{\mathrm{RAS1}}$	PIO5: input with pullup	PIOMODE0[5]
PIO6	INT8 / PWD	PIO6: input with pullup	PIOMODE0[6]
PIO7	INT7	PIO7: input with pullup	PIOMODE0[7]
PIO8	ARDY	ARDY: input with pullup	PIOMODE0[8]
PIO9	DRQ0	PIO9: input with pulldown	PIOMODE0[9]
PIO10	SDEN	PIO10: input with pulldown	PIOMODE0[10]
PIO11	SCLK	PIO11: input with pullup	PIOMODE0[11]
PIO12	SDATA	PIO12: input with pullup	PIOMODE0[12]
PIO13	$\overline{\mathrm{PCS0}}$	$\overline{\mathrm{PCS0}}$: input with pullup	PIOMODE0[13]
PIO14	$\overline{\mathrm{PCS1}}$	$\overline{\mathrm{PCS1}}$: input with pullup	PIOMODE0[14]
PIO15	$\overline{\mathrm{WR}}$	$\overline{\mathrm{WR}}$: input with pullup	PIOMODE0[15]
PIO16	RXD_HU	PIO16: input with pullup	PIOMODE1[0]
PIO17	$\overline{\text{DCE\_CTS\_A}}$[1] / $\overline{\text{PCM\_TSC\_A}}$[1]	PIO17: input with pullup	PIOMODE1[1]
PIO18	$\overline{\text{DCE\_RTR\_A}}$[1]	PIO18: input with pullup	PIOMODE1[2]
PIO19	INT6	PIO19: input with pullup	PIOMODE1[3]
PIO20	TXD_U/ DCE_TXD_D[2] / PCM_TXD_D[2]	PIO20: input with pullup	PIOMODE1[4]
PIO21	UCLK / USBSOF[3] / USBSCI[3]	PIO21: input with pullup	PIOMODE1[5]
PIO22	DCE_RCLK_C[2] / PCM_CLK_C[2]	PIO22: input with pulldown	PIOMODE1[6]
PIO23	DCE_TCLK_C[2] / PCM_FSC_C[2]	PIO23: input with pulldown	PIOMODE1[7]
PIO24	$\overline{\text{CTS\_U}}$ / DCE_TCLK_D[2] / PCM_FSC_D[2]	PIO24: input with pullup	PIOMODE1[8]
PIO25	$\overline{\text{RTR\_U}}$ / DCE_RCLK_D[2] / PCM_CLK_D[2]	PIO25: input with pullup	PIOMODE1[9]
PIO26	RXD_U / DCE_RXD_D[2] / PCM_RXD_D[2]	PIO26: input with pullup	PIOMODE1[10]
PIO27	TMRIN0	PIO27: input with pullup	PIOMODE1[11]
PIO28	TMROUT0	PIO28: input with pulldown	PIOMODE1[12]
PIO29	DT/$\overline{\mathrm{R}}$	DT/$\overline{\mathrm{R}}$: three-state output with pullup	PIOMODE1[13]
PIO30	$\overline{\mathrm{DEN}}$ / $\overline{\mathrm{DS}}$	$\overline{\mathrm{DEN}}$: three-state output with pullup	PIOMODE1[14]
PIO31	$\overline{\mathrm{PCS7}}$	PIO31: input with pullup	PIOMODE1[15]
PIO32	$\overline{\mathrm{PCS6}}$	PIO32: input with pullup	PIOMODE2[0]
PIO33	ALE	ALE: three-state output with pulldown	PIOMODE2[1]
PIO34	$\overline{\mathrm{BHE}}$	$\overline{\mathrm{BHE}}$: input with pullup	PIOMODE2[2]
PIO35	SRDY	SRDY: input with pullup	PIOMODE2[3]
PIO36	DCE_RXD_B[1] / PCM_RXD_B[1]	PIO36: input with pullup	PIOMODE2[4]
PIO37	DCE_TXD_B[1] / PCM_TXD_B[1]	PIO37: input with pullup	PIOMODE2[5]
PIO38	$\overline{\text{DCE\_CTS\_B}}$[1] / $\overline{\text{PCM\_TSC\_B}}$[1]	PIO38: input with pullup	PIOMODE2[6]
PIO39	$\overline{\text{DCE\_RTR\_B}}$[1]	PIO39: input with pullup	PIOMODE2[7]

Table 13-2 PIO Multiplexed Signals (Continued)

Signal	Multiplexed Signal(s)	Default Signal	Register[Bit No.]
PIO40	DCE_RCLK_B[1] / PCM_CLK_B[1]	PIO40: input with pullup	PIOMODE2[8]
PIO41	DCE_TCLK_B[1] / PCM_FSC_B[1]	PIO41: input with pullup	PIOMODE2[9]
PIO42	DCE_RXD_C[2] / PCM_RXD_C[2]	PIO42: input with pulldown	PIOMODE2[10]
PIO43	DCE_TXD_C[2] / PCM_TXD_C[2]	PIO43: input with pulldown	PIOMODE2[11]
PIO44	$\overline{\text{DCE\_CTS\_C}}$[2] / $\overline{\text{PCM\_TSC\_C}}$[2]	PIO44: input with pullup	PIOMODE2[12]
PIO45	$\overline{\text{DCE\_RTR\_C}}$[2]	PIO45: input with pullup	PIOMODE2[13]
PIO46	$\overline{\text{CTS\_HU}}$ / $\overline{\text{DCE\_CTS\_D}}$[2] / $\overline{\text{PCM\_TSC\_D}}$[2]	PIO46: input with pullup	PIOMODE2[14]
PIO47	$\overline{\text{RTR\_HU}}$ / $\overline{\text{DCE\_RTR\_D}}$[2]	PIO47: input with pullup	PIOMODE2[15]

Notes:

1. HDLC channel A and B pin functions are supported on the Am186CC and Am186CH microcontrollers only.

2. HDLC channel C and D pin functions are supported on the Am186CC microcontroller only.

3. USB pin functions are supported on the Am186CC and Am186CU microcontrollers only.

PIO Mode 0 (PIOMODE0) — Offset 3C0h

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	PMODE 15	PMODE 14	PMODE 13	PMODE 12	PMODE 11	PMODE 10	PMODE 9	PMODE 8	PMODE 7	PMODE 6	PMODE 5	PMODE 4	PMODE 3	PMODE 2	PMODE 1	PMODE 0
Software Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Hardware Set/Clear	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Chip Reset Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Microcontroller:
Am186CC ✔
Am186CH ✔
Am186CU ✔

Register Description

This is the mode register for Programmable Input/Output (PIO) pins 15–0. The PIOMODE0 register bits, along with the corresponding PIODIR0 register bits (see page 13-5), determine whether each PIO-capable pin performs its preassigned function or is configured as a custom PIO signal.

Register definitions for PIO pins 31–16 begin on page 13-9.

Register definitions for PIO pins 47–32 begin on page 13-14.

Bit Definitions

Bit	Name	Function
15–0	PMODE15–PMODE0	**PIO Mode Bits 15–0**

Each PMODE15–PMODE0 bit determines whether a PIO-capable pin is enabled as a PIO (if an output) or terminated (if an input). Pins are programmed as input or output in the corresponding PIO Direction register (see page 13-5). Each PMODE bit corresponds to a PIO pin; for example, PMODE15 sets the mode for the PIO15 pin.

The PIOMODE0 and PIODIR0 register bits combine to configure one of the following classes of behavior for each PIO-capable pin:

Mode	Direction	Pin Function
0	0	Alternate operation
0	1	Terminated PIO input
1	0	PIO output
1	1	Unterminated PIO input

Terminated operation applies an internal 100-KΩ pullup or pulldown resistor, depending on the pin. For output, termination is not relevant because the pin would overdrive the internal termination, so this case is used to share the pin with an alternate preassigned function.

Programming Notes

The following PIO signals are multiplexed with signals that may be used by emulators: PIO8, PIO15, PIO33, PIO34, and PIO35. Consider emulator requirements for the alternate signals before using these pins as PIOs.

The following PIO signals can be configured as interrupt sources in the interrupt controller's Shared Mask (SHMASK) register (see page 9-45): PIO5, PIO15, PIO27, PIO29, PIO30, PIO33, PIO34, and PIO35. Typically, these signals should be configured as inputs when used as an interrupt source. If any of these signals is configured as both a PIO output and as an interrupt source, the PIO output signal generates interrupts.

Unlike the PIO interrupts controlled by the SHMASK register, the signals PIO6, PIO7, and PIO19 are multiplexed with interrupt pins INT8, INT7, and INT6, respectively. To use these pins as interrupt sources, software must clear the corresponding PIO mode and direction bits. This disables the PIO outputs so they cannot affect the interrupt state.

PIO Direction 0 (PIODIR0) Offset 3C2h

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	PDIR15	PDIR14	PDIR13	PDIR12	PDIR11	PDIR10	PDIR9	PDIR8	PDIR7	PDIR6	PDIR5	PDIR4	PDIR3	PDIR2	PDIR1	PDIR0
Software Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Hardware Set/Clear	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Chip Reset Default	0	0	0	1	1	1	1	0	1	1	1	1	1	1	1	1

Microcontroller:
Am186CC ✔
Am186CH ✔
Am186CU ✔

Register Description

This is the direction register for Programmable Input/Output (PIO) pins 15–0. The default direction of each PIO pin is indicated by the Chip Reset Default values in the register diagram above.

Bit Definitions

Bit	Name	Function
15–0	PDIR15–PDIR0	**PIO Direction Bits 15–0**

Each PDIR15–PDIR0 bit individually programs a PIO-capable pin as an input or output. Each bit corresponds to a PIO pin; for example, PDIR15 sets the direction for the PIO15 pin.

The PIODIR0 register bits combine with the corresponding PIOMODE0 register bits (see page 13-4) to configure one of the following classes of behavior for each PIO-capable pin:

Mode	Direction	Pin Function
0	0	Alternate operation
0	1	Terminated PIO input
1	0	PIO output
1	1	Unterminated PIO input

Terminated operation applies an internal 100-KΩ pullup or pulldown resistor, depending on the pin. For output, termination is not relevant because the pin would overdrive the internal termination, so this case is used to share the pin with an alternate preassigned function.

Programming Notes

PIO Data 0 (PIODATA0) Offset 3C4h

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	PDATA 15	PDATA 14	PDATA 13	PDATA 12	PDATA 11	PDATA 10	PDATA 9	PDATA 8	PDATA 7	PDATA 6	PDATA 5	PDATA 4	PDATA 3	PDATA 2	PDATA 1	PDATA 0
Software Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Hardware Set/Clear	S/C	S/C	S/C	S/C	S/C	S/C	S/C	S/C	S/C	S/C	S/C	S/C	S/C	S/C	S/C	S/C
Chip Reset Default	x	x	x	?	?	?	?	x	?	?	?	?	?	?	?	?

Microcontroller:
Am186CC ✓
Am186CH ✓
Am186CU ✓

Register Description

This is the data read and write register for Programmable Input/Output (PIO) pins 15–0.

Bit Definitions

Bit	Name	Function
15–0	PDATA15– PDATA0	**PIO Data Bits 15–0** Each PDATA15–PDATA0 bit determines the level driven on each PIO pin or reflects the external level of the pin, depending on whether the pin is configured as an output or an input in the corresponding PIODIR0 register bit (see page 13-5). Each bit in this field corresponds to a PIO-capable pin; for example, PDATA15 (bit 15 of this register) corresponds to the PIO15 pin.

Programming Notes

If a PIO pin is enabled as an output, the corresponding bit value in the PIODATA0 register is driven on the pin with no inversion.

The current state of a PIO pin is reflected with no inversion in the corresponding bit value read from the PIODATA0 register. The pin state can be read regardless of the pin's configuration (input, output, or alternate function).

A bit value written to the PIODATA0 register (or modified by writing to the PIOSET0 or PIOCLR0 register) is stored regardless of the corresponding PIO direction, but a bit value read from PIODATA0 does not necessarily match the stored value unless the corresponding PIO is enabled as an output. The current stored value can always be read from the PIOSET0 or PIOCLR0 register. Enabling a PIO output drives the pin with the stored bit value.

To use a PIO pin as an open-drain output (output drives Low or is disabled), set the PIO pin's mode and data bits, then write the pin's data value into the associated bit position in the PIODIR0 register.

PIO Set 0 (PIOSET0) — Offset 3C6h

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	PSET15	PSET14	PSET13	PSET12	PSET11	PSET10	PSET9	PSET8	PSET7	PSET6	PSET5	PSET4	PSET3	PSET2	PSET1	PSET0
Software Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Hardware Set/Clear	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Chip Reset Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Microcontroller:	
Am186CC	✓
Am186CH	✓
Am186CU	✓

Register Description

This register provides an efficient way to set individual bits in the corresponding PIODATA0 register without affecting other bits in the register.

Bit Definitions

Bit	Name	Function
15–0	PSET15–PSET0	**PIO Set Bits 15–0** The PSET15–PSET0 bit field determines the contents of the corresponding PIODATA0 register or reflects the contents of the PIODATA0 register. The PIOSET0 register bits correspond to PIODATA0 register bits. Writing a 1 to a bit in this field causes the corresponding PIODATA0 register bit to be forced to a 1. Writing a 0 to a bit in this field does not affect the corresponding PIODATA0 register bit. Reading the PSET15–PSET0 bit field returns the last value written by software in the corresponding PIODATA0 register (including changes made via the Set and Clear registers). This allows software to read back the value that would be driven if a PIO pin is changed from input to output.

Programming Notes

The PIOSET0 register is only active during execution of an instruction that writes to it.

PIO Clear 0 (PIOCLR0) Offset 3C8h

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	PCLR15	PCLR14	PCLR13	PCLR12	PCLR11	PCLR10	PCLR9	PCLR8	PCLR7	PCLR6	PCLR5	PCLR4	PCLR3	PCLR2	PCLR1	PCLR0
Software Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Hardware Set/Clear	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Chip Reset Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Microcontroller:	
Am186CC	✓
Am186CH	✓
Am186CU	✓

Register Description

This register provides an efficient way to clear individual bits in the corresponding PIODATA0 register without affecting other bits in the register.

Bit Definitions

Bit	Name	Function
15–0	PCLR15–PCLR0	**PIO Clear Bits 15–0** The PCLR15–PCLR0 bit field determines the contents of the corresponding PIODATA0 register or reflects the contents of the PIODATA0 register. The PIOCLR0 register bits correspond to PIODATA0 register bits. Writing a 1 to a bit in this field causes the corresponding PIODATA0 register bit to be forced to a 0. Writing a 0 to a bit in this field does not affect the corresponding PIODATA0 register bit. Reading the PCLR15–PCLR0 bit field returns the last value written by software in the corresponding PIODATA0 register (including changes made via the Set and Clear registers). This allows software to read back the value that would be driven if a PIO pin is changed from input to output.

Programming Notes

The PIOCLR0 register is only active during execution of an instruction that writes to it.

PIO Mode 1 (PIOMODE1) Offset 3CAh

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	PMODE 31	PMODE 30	PMODE 29	PMODE 28	PMODE 27	PMODE 26	PMODE 25	PMODE 24	PMODE 23	PMODE 22	PMODE 21	PMODE 20	PMODE 19	PMODE 18	PMODE 17	PMODE 16
Software Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Hardware Set/Clear	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Chip Reset Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Microcontroller:	
Am186CC	✔
Am186CH	✔
Am186CU	✔

Register Description

This is the mode register for Programmable Input/Output (PIO) pins 31–16. The PIOMODE1 register bits, along with the corresponding PIODIR1 register bits (see page 13-10), determine whether each PIO-capable pin performs its preassigned function or is configured as a custom PIO signal.

Register definitions for PIO pins 15–0 begin on page 13-4.

Register definitions for PIO pins 47–32 begin on page 13-14.

Bit Definitions

Bit	Name	Function
15–0	PMODE31– PMODE16	**PIO Mode Bits 31–16**

Each PMODE31–PMODE16 bit determines whether a PIO-capable pin is enabled as a PIO (if an output) or terminated (if an input). Pins are programmed as input or output in the corresponding PIO Direction register (see page 13-10). Each PMODE bit corresponds to a PIO pin; for example, PMODE31 sets the mode for the PIO31 pin.

The PIOMODE1 and PIODIR1 register bits combine to configure one of the following classes of behavior for each PIO-capable pin:

Mode	Direction	Pin Function
0	0	Alternate operation
0	1	Terminated PIO input
1	0	PIO output
1	1	Unterminated PIO input

Terminated operation applies an internal 100-KΩ pullup or pulldown resistor, depending on the pin. For output, termination is not relevant because the pin would overdrive the internal termination, so this case is used to share the pin with an alternate preassigned function.

Programming Notes

The following PIO signals are multiplexed with signals that may be used by emulators: PIO8, PIO15, PIO33, PIO34, and PIO35. Consider emulator requirements for the alternate signals before using these pins as PIOs.

The following PIO signals can be configured as interrupt sources in the interrupt controller's Shared Mask (SHMASK) register (see page 9-45): PIO5, PIO15, PIO27, PIO29, PIO30, PIO33, PIO34, and PIO35. Typically, these signals should be configured as inputs when used as an interrupt source. If any of these signals is configured as both a PIO output and as an interrupt source, the PIO output signal generates interrupts.

Unlike the PIO interrupts controlled by the SHMASK register, the signals PIO6, PIO7, and PIO19 are multiplexed with interrupt pins INT8, INT7, and INT6, respectively. To use these pins as interrupt sources, software must clear the corresponding PIO mode and direction bits. This disables the PIO outputs so they cannot affect the interrupt state.

PIO Direction 1 (PIODIR1) Offset 3CCh

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	PDIR31	PDIR30	PDIR29	PDIR28	PDIR27	PDIR26	PDIR25	PDIR24	PDIR23	PDIR22	PDIR21	PDIR20	PDIR19	PDIR18	PDIR17	PDIR16
Software Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Hardware Set/Clear	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Chip Reset Default	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1

Microcontroller:

Am186CC	✔
Am186CH	✔
Am186CU	✔

Register Description

This is the direction register for Programmable Input/Output (PIO) pins 31–16. The default direction of each PIO pin is indicated by the Chip Reset Default values in the register diagram above.

Bit Definitions

Bit	Name	Function
15–0	PDIR31– PDIR16	**PIO Direction Bits 31–16** Each PDIR31–PDIR16 bit individually programs a PIO-capable pin as an input or output. Each bit corresponds to a PIO pin; for example, PDIR31 sets the direction for the PIO31 pin. The PIODIR1 register bits combine with the corresponding PIOMODE1 register bits (see page 13-9) to configure one of the following classes of behavior for each PIO-capable pin:

Mode	Direction	Pin Function
0	0	Alternate operation
0	1	Terminated PIO input
1	0	PIO output
1	1	Unterminated PIO input

Terminated operation applies an internal 100-KΩ pullup or pulldown resistor, depending on the pin. For output, termination is not relevant because the pin would overdrive the internal termination, so this case is used to share the pin with an alternate preassigned function.

Programming Notes

PIO Data 1 (PIODATA1) Offset 3CEh

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	PDATA 31	PDATA 30	PDATA 29	PDATA 28	PDATA 27	PDATA 26	PDATA 25	PDATA 24	PDATA 23	PDATA 22	PDATA 21	PDATA 20	PDATA 19	PDATA 18	PDATA 17	PDATA 16
Software Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Hardware Set/Clear	S/C	S/C	S/C	S/C	S/C	S/C	S/C	S/C	S/C	S/C	S/C	S/C	S/C	S/C	S/C	S/C
Chip Reset Default	?	x	x	?	?	?	?	?	?	?	?	?	?	?	?	?

Microcontroller:
Am186CC ✔
Am186CH ✔
Am186CU ✔

Register Description

This is the data read and write register for Programmable Input/Output (PIO) pins 31–16.

Bit Definitions

Bit	Name	Function
15–0	PDATA31– PDATA16	**PIO Data Bits 31–16** Each PDATA31–PDATA16 bit determines the level driven on each PIO pin or reflects the external level of the pin, depending on whether the pin is configured as an output or an input in the corresponding PIODIR1 register bit (see page 13-10). Each bit in this field corresponds to a PIO-capable pin; for example, PDATA31 (bit 15 of this register) corresponds to the PIO31 pin.

Programming Notes

If a PIO pin is enabled as an output, the corresponding bit value in the PIODATA1 register is driven on the pin with no inversion.

The current state of a PIO pin is reflected with no inversion in the corresponding bit value read from the PIODATA1 register. The pin state can be read regardless of the pin's configuration (input, output, or alternate function).

A bit value written to the PIODATA1 register (or modified by writing to the PIOSET1 or PIOCLR1 register) is stored regardless of the corresponding PIO direction, but a bit value read from PIODATA1 does not necessarily match the stored value unless the corresponding PIO is enabled as an output. The current stored value can always be read from the PIOSET1 or PIOCLR1 register. Enabling a PIO output drives the pin with the stored bit value.

To use a PIO pin as an open-drain output (output drives Low or is disabled), set the PIO pin's mode and data bits, then write the pin's data value into the associated bit position in the PIODIR1 register.

PIO Set 1 (PIOSET1) Offset 3D0h

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	PSET31	PSET30	PSET29	PSET28	PSET27	PSET26	PSET25	PSET24	PSET23	PSET22	PSET21	PSET20	PSET19	PSET18	PSET17	PSET16
Software Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Hardware Set/Clear	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Chip Reset Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Microcontroller:	
Am186CC	✓
Am186CH	✓
Am186CU	✓

Register Description

This register provides an efficient way to set individual bits in the corresponding PIODATA1 register without affecting other bits in the register.

Bit Definitions

Bit	Name	Function
15–0	PSET31– PSET16	**PIO Set Bits 31–16** The PSET31–PSET16 bit field determines the contents of the corresponding PIODATA1 register or reflects the contents of the PIODATA1 register. The PIOSET1 register bits correspond to PIODATA1 register bits. Writing a 1 to a bit in this field causes the corresponding PIODATA1 register bit to be forced to a 1. Writing a 0 to a bit in this field does not affect the corresponding PIODATA1 register bit. Reading the PSET31–PSET16 bit field returns the last value written by software in the corresponding PIODATA1 register (including changes made via the Set and Clear registers). This allows software to read back the value that would be driven if a PIO pin is changed from input to output.

Programming Notes

The PIOSET1 register is only active during execution of an instruction that writes to it.

PIO Clear 1 (PIOCLR1) Offset 3D2h

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	PCLR31	PCLR30	PCLR29	PCLR28	PCLR27	PCLR26	PCLR25	PCLR24	PCLR23	PCLR22	PCLR21	PCLR20	PCLR19	PCLR18	PCLR17	PCLR16
Software Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Hardware Set/Clear	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Chip Reset Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Microcontroller:
Am186CC ☑
Am186CH ☑
Am186CU ☑

Register Description

This register provides an efficient way to clear individual bits in the corresponding PIODATA1 register without affecting other bits in the register.

Bit Definitions

Bit	Name	Function
15–0	PCLR31– PCLR16	**PIO Clear Bits 31–16** The PCLR31–PCLR16 bit field determines the contents of the corresponding PIODATA1 register or reflects the contents of the PIODATA1 register. The PIOCLR1 register bits correspond to PIODATA1 register bits. Writing a 1 to a bit in this field causes the corresponding PIODATA1 register bit to be forced to a 0. Writing a 0 to a bit in this field does not affect the corresponding PIODATA1 register bit. Reading the PCLR31–PCLR16 bit field returns the last value written by software in the corresponding PIODATA1 register (including changes made via the Set and Clear registers). This allows software to read back the value that would be driven if a PIO pin is changed from input to output.

Programming Notes

The PIOCLR1 register is only active during execution of an instruction that writes to it.

PIO Mode 2 (PIOMODE2) — Offset 3D4h

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	PMODE 47	PMODE 46	PMODE 45	PMODE 44	PMODE 43	PMODE 42	PMODE 41	PMODE 40	PMODE 39	PMODE 38	PMODE 37	PMODE 36	PMODE 35	PMODE 34	PMODE 33	PMODE 32
Software Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Hardware Set/Clear	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Chip Reset Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Microcontroller:

Am186CC	✔
Am186CH	✔
Am186CU	✔

Register Description

This is the mode register for Programmable Input/Output (PIO) pins 47–32. The PIOMODE2 register bits, along with the corresponding PIODIR2 register bits (see page 13-15), determine whether each PIO-capable pin performs its preassigned function or is configured as a custom PIO signal.

Register definitions for PIO pins 15–0 begin on page 13-4.

Register definitions for PIO pins 31–16 begin on page 13-9.

Bit Definitions

Bit	Name	Function
15–0	PMODE47–PMODE32	**PIO Mode Bits 47–32**

Each PMODE47–PMODE32 bit determines whether a PIO-capable pin is enabled as a PIO (if an output) or terminated (if an input). Pins are programmed as input or output in the corresponding PIO Direction register (see page 13-15). Each PMODE bit corresponds to a PIO pin; for example, PMODE47 sets the mode for the PIO47 pin.

The PIOMODE2 and PIODIR2 register bits combine to configure one of the following classes of behavior for each PIO-capable pin:

Mode	Direction	Pin Function
0	0	Alternate operation
0	1	Terminated PIO input
1	0	PIO output
1	1	Unterminated PIO input

Terminated operation applies an internal 100-KΩ pullup or pulldown resistor, depending on the pin. For output, termination is not relevant because the pin would overdrive the internal termination, so this case is used to share the pin with an alternate preassigned function.

Programming Notes

The following PIO signals are multiplexed with signals that may be used by emulators: PIO8, PIO15, PIO33, PIO34, and PIO35. Consider emulator requirements for the alternate signals before using these pins as PIOs.

The following PIO signals can be configured as interrupt sources in the interrupt controller’s Shared Mask (SHMASK) register (see page 9-45): PIO5, PIO15, PIO27, PIO29, PIO30, PIO33, PIO34, and PIO35. Typically, these signals should be configured as inputs when used as an interrupt source. If any of these signals is configured as both a PIO output and as an interrupt source, the PIO output signal generates interrupts.

Unlike the PIO interrupts controlled by the SHMASK register, the signals PIO6, PIO7, and PIO19 are multiplexed with interrupt pins INT8, INT7, and INT6, respectively. To use these pins as interrupt sources, software must clear the corresponding PIO mode and direction bits. This disables the PIO outputs so they cannot affect the interrupt state.

PIO Direction 2 (PIODIR2) Offset 3D6h

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	PDIR47	PDIR46	PDIR45	PDIR44	PDIR43	PDIR42	PDIR41	PDIR40	PDIR39	PDIR38	PDIR37	PDIR36	PDIR35	PDIR34	PDIR33	PDIR32
Software Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Hardware Set/Clear	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Chip Reset Default	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	1

Microcontroller:	
Am186CC	✔
Am186CH	✔
Am186CU	✔

Register Description

This is the direction register for Programmable Input/Output (PIO) pins 47–32. The default direction of each PIO pin is indicated by the Chip Reset Default values in the register diagram above.

Bit Definitions

Bit **Name** **Function**

15–0 PDIR47–PDIR32

PIO Direction Bits 47–32
Each PDIR47–PDIR32 bit individually programs a PIO-capable pin as an input or output. Each bit corresponds to a PIO pin; for example, PDIR47 sets the direction for the PIO47 pin.

The PIODIR2 register bits combine with the corresponding PIOMODE2 register bits (see page 13-14) to configure one of the following classes of behavior for each PIO-capable pin:

Mode	Direction	Pin Function
0	0	Alternate operation
0	1	Terminated PIO input
1	0	PIO output
1	1	Unterminated PIO input

Terminated operation applies an internal 100-KΩ pullup or pulldown resistor, depending on the pin. For output, termination is not relevant because the pin would overdrive the internal termination, so this case is used to share the pin with an alternate preassigned function.

Programming Notes

PIO Data 2 (PIODATA2) Offset 3D8h

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	PDATA 47	PDATA 46	PDATA 45	PDATA 44	PDATA 43	PDATA 42	PDATA 41	PDATA 40	PDATA 39	PDATA 38	PDATA 37	PDATA 36	PDATA 35	PDATA 34	PDATA 33	PDATA 32
Software Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Hardware Set/Clear	S/C	S/C	S/C	S/C	S/C	S/C	S/C	S/C	S/C	S/C	S/C	S/C	S/C	S/C	S/C	S/C
Chip Reset Default	?	?	?	?	?	?	?	?	?	?	?	?	x	x	x	?

Microcontroller:
Am186CC ✔
Am186CH ✔
Am186CU ✔

Register Description

This is the data read and write register for Programmable Input/Output (PIO) pins 47–32.

Bit Definitions

Bit	Name	Function
15–0	PDATA47– PDATA32	**PIO Data Bits 47–32** Each PDATA47–PDATA32 bit determines the level driven on each PIO pin or reflects the external level of the pin, depending on whether the pin is configured as an output or an input in the corresponding PIODIR2 register bit (see page 13-15). Each bit in this field corresponds to a PIO-capable pin; for example, PDATA47 (bit 15 of this register) corresponds to the PIO47 pin.

Programming Notes

If a PIO pin is enabled as an output, the corresponding bit value in the PIODATA2 register is driven on the pin with no inversion.

The current state of a PIO pin is reflected with no inversion in the corresponding bit value read from the PIODATA2 register. The pin state can be read regardless of the pin's configuration (input, output, or alternate function).

A bit value written to the PIODATA2 register (or modified by writing to the PIOSET2 or PIOCLR2 register) is stored regardless of the corresponding PIO direction, but a bit value read from PIODATA2 does not necessarily match the stored value unless the corresponding PIO is enabled as an output. The current stored value can always be read from the PIOSET2 or PIOCLR2 register. Enabling a PIO output drives the pin with the stored bit value.

To use a PIO pin as an open-drain output (output drives Low or is disabled), set the PIO pin's mode and data bits, then write the pin's data value into the associated bit position in the PIODIR2 register.

PIO Set 2 (PIOSET2) Offset 3DAh

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	PSET47	PSET46	PSET45	PSET44	PSET43	PSET42	PSET41	PSET40	PSET39	PSET38	PSET37	PSET36	PSET35	PSET34	PSET33	PSET32
Software Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Hardware Set/Clear	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Chip Reset Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Microcontroller:

Am186CC	✓
Am186CH	✓
Am186CU	✓

Register Description

This register provides an efficient way to set individual bits in the corresponding PIODATA2 register without affecting other bits in the register.

Bit Definitions

Bit	Name	Function
15–0	PSET47– PSET32	**PIO Set Bits 47–32** The PSET47–PSET32 bit field determines the contents of the corresponding PIODATA2 register or reflects the contents of the PIODATA2 register. The PIOSET2 register bits correspond to PIODATA2 register bits. Writing a 1 to a bit in this field causes the corresponding PIODATA2 register bit to be forced to a 1. Writing a 0 to a bit in this field does not affect the corresponding PIODATA2 register bit. Reading the PSET47–PSET32 bit field returns the last value written by software in the corresponding PIODATA2 register (including changes made via the Set and Clear registers). This allows software to read back the value that would be driven if a PIO pin is changed from input to output.

Programming Notes

The PIOSET2 register is only active during execution of an instruction that writes to it.

PIO Clear 2 (PIOCLR2) Offset 3DCh

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	PCLR47	PCLR46	PCLR45	PCLR44	PCLR43	PCLR42	PCLR41	PCLR40	PCLR39	PCLR38	PCLR37	PCLR36	PCLR35	PCLR34	PCLR33	PCLR32
Software Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Hardware Set/Clear	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Chip Reset Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Microcontroller:	
Am186CC	✓
Am186CH	✓
Am186CU	✓

Register Description

This register provides an efficient way to clear individual bits in the corresponding PIODATA2 register without affecting other bits in the register.

Bit Definitions

Bit	Name	Function
15–0	PCLR47– PCLR32	**PIO Clear Bits 47–32** The PCLR47–PCLR32 bit field determines the contents of the corresponding PIODATA2 register or reflects the contents of the PIODATA2 register. The PIOCLR2 register bits correspond to PIODATA2 register bits. Writing a 1 to a bit in this field causes the corresponding PIODATA2 register bit to be forced to a 0. Writing a 0 to a bit in this field does not affect the corresponding PIODATA2 register bit. Reading the PCLR47–PCLR32 bit field returns the last value written by software in the corresponding PIODATA2 register (including changes made via the Set and Clear registers). This allows software to read back the value that would be driven if a PIO pin is changed from input to output.

Programming Notes

The PIOCLR2 register is only active during execution of an instruction that writes to it.

CHAPTER

14 RESET CONFIGURATION REGISTER

14.1 OVERVIEW

This chapter describes the Reset Configuration register on the Am186CC/CH/CU microcontrollers.

The Reset Configuration register has no effect on the microcontroller's operation. It simply reflects the signal states that were present on the AD15–AD0 pins during an external reset. A design can use weak pullup or pulldown resistors on these pins to convey configuration information (such as jumper or switch settings) to software.

Table 14-1 lists the Reset Configuration register's mnemonic, offset, and page number.

Table 14-1 Reset Configuration Register Map

Register Name	Mnemonic	Offset	Page Number
Reset Configuration	RESCON	3DEh	page 14-2

Reset Configuration (RESCON) Offset 3DEh

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	RCD15	RCD14	RCD13	RCD12	RCD11	RCD10	RCD9	RCD8	RCD7	RCD6	RCD5	RCD4	RCD3	RCD2	RCD1	RCD0
Software Read/Write	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Hardware Set/Clear	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Chip Reset Default	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?

Microcontroller:
Am186CC ✔
Am186CH ✔
Am186CU ✔

Register Description

This register provides the means to make design-specific hardware configuration information available to software.

Bit Definitions

Bit	Name	Function
15–0	RCD15– RCD0	**Reset Configuration** On an external reset, the RCD15–RCD0 bits are set to the values found on the AD15–AD0 pins. There is a one-to-one correspondence between address/data bus signals and the RESCON register's bits during an external reset. On the Am186CC/CH/CU microcontrollers, the AD15 pin corresponds to bit 15 of the RESCON register, and so on. After $\overline{\text{RES}}$ is deasserted, the RESCON register holds its value. This value can be read by software to determine the configuration information. The RESCON register is not affected by a watchdog timer reset.

Programming Notes

The RESCON register latches system-configuration information that is presented to the processor on the address/data bus (AD15–AD0) at the rising edge of $\overline{\text{RES}}$. The interpretation of this information is system-specific. The processor does not impose any predetermined interpretation, but simply provides a means for communicating this information to software.

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

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

CHAPTER

15 WATCHDOG TIMER REGISTER

15.1 OVERVIEW

This chapter describes the Watchdog Timer register on the Am186CC/CH/CU microcontrollers.

The watchdog timer can be used to regain control of the system when software fails to behave as expected (i.e., when the system hangs). Unless the watchdog timer is disabled by software, a repetitive function in the device software must reset the timer periodically at a rate faster than the programmed timeout value.

The watchdog timer can be programmed to generate either a nonmaskable interrupt (NMI) or an internal reset when the timeout is reached. The timer can also be programmed to generate a system reset signal when a watchdog timer reset occurs.

For more information about using the watchdog timer, see the *Am186CC/CH/CU Microcontrollers User's Manual*, order #21914.

Table 15-1 lists the Reset Configuration register's mnemonic, offset, and page number.

Table 15-1 Watchdog Timer Register Map

Register Name	Mnemonic	Offset	Page Number
Watchdog Timer Control	WDTCON	3E0h	page 15-2

Watchdog Timer Control (WDTCON) Offset 03E0h

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	ENA	WRST	RSTFLAG	NMIFLAG	*Res*			EXRST	ES							
Software Read/Write	R/W	R/W	R/W0	R/W0	R			R/W	R/W							
Hardware Set/Clear	—	—	—	—	—			—	—							
Chip Reset Default	1	1	0	0	0	0	0	1	1	0	0	0	0	0	0	0

Microcontroller:	
Am186CC	✔
Am186CH	✔
Am186CU	✔

Register Description

This is the combined status and control register for the watchdog timer. The watchdog timer can generate either an internal reset or NMI if its count register is not cleared periodically by software.

Following an external or watchdog timer reset, the watchdog timer is enabled and generates an internal reset if its count is not cleared within every 2^{26} processor clock cycles. To clear the count, software must write the clear key sequence of AAAAh followed by 5555h to the WDTCON register. It is not possible to read the current count of the watchdog timer.

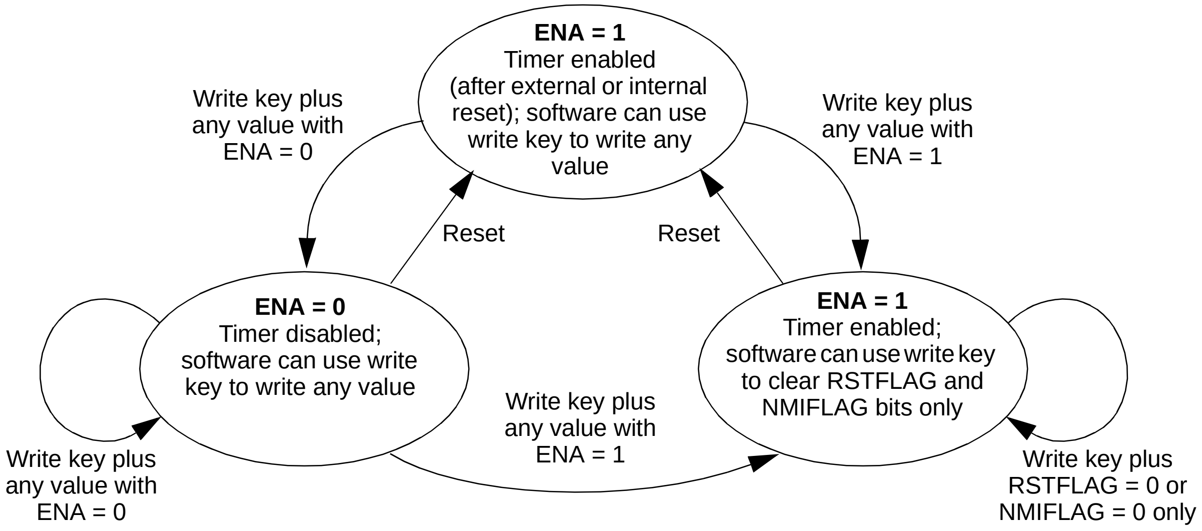
Software cannot change the WDTCON register's contents unless it first writes the write key sequence of 3333h followed by CCCCh to the WDTCON register. This sequence opens the register for a single write.

Following an external or watchdog timer reset, software can use the write key sequence to make as many changes to WDTCON as needed, so long as the ENA bit is always cleared. Clearing the ENA bit disables the watchdog timer, allowing debug monitor code to leave the watchdog timer alone until the system software is ready to support it.

When software writes a 1 to the ENA bit, the entire WDTCON register becomes read-only except for the RSTFLAG and NMIFLAG bits, which software can still clear by using the write key sequence. The ENA bit is also read-only after it is set, and it cannot be cleared again until after an external or watchdog timer reset. This ensures that runaway software cannot modify the watchdog timer programming. Figure 15-1 illustrates the rules for accessing the WDTCON register.

Any number of processor cycles, including memory and I/O reads and writes, can be inserted between the two halves of a key sequence or between the write key and the writing of data as long as the inserted cycles do not access the WDTCON register.

Figure 15-1 Access to the WDTCON Register


ENA = 1
Timer enabled
(after external or internal reset); software can use write key to write any value
Write key plus any value with ENA = 0
Write key plus any value with ENA = 1
Reset
Reset
ENA = 0
Timer disabled; software can use write key to write any value
ENA = 1
Timer enabled; software can use write key to clear RSTFLAG and NMIFLAG bits only
Write key plus any value with ENA = 1
Write key plus any value with ENA = 0
Write key plus RSTFLAG = 0 or NMIFLAG = 0 only

Notes:

Only one write is allowed to the WDTCON register after each write key sequence of 3333h followed by CCCCh.

Bit Definitions

Bit	Name	Function
15	ENA	**Enable** 0 = The watchdog timer is disabled. 1 = The watchdog timer is enabled. When the ENA bit is written with a 1, the current count is automatically reset to 0 and the ENA, WRST, EXRST, and ES bit fields become read only.
14	WRST	**Watchdog Reset** 0= If the NMIFLAG bit is 0, the watchdog timer generates an NMI when a watchdog timer timeout occurs. When the NMIFLAG bit is 1, an internal reset is always generated on a WDT timeout regardless of the setting of the WRST bit. 1 = The watchdog timer generates an internal reset when a watchdog timer timeout occurs. If the EXRST bit is also set, a system reset is also generated.
13	RSTFLAG	**Reset Flag** 0 = No watchdog timer reset event has occurred. 1 = A watchdog timer reset event has occurred. Software can clear the RSTFLAG bit after writing the write key sequence of 3333h followed by CCCCh.
12	NMIFLAG	**NMI Flag** 0 = No watchdog timer NMI event has occurred. 1 = A watchdog timer NMI event has occurred. If a watchdog timer timeout occurs while the NMIFLAG bit is set, the watchdog timer generates a reset regardless of the setting of the WRST bit. Software can clear the NMIFLAG bit after writing the write key sequence of 3333h followed by CCCCh.
11–9	Res	**Reserved** For compatibility with future devices, always write this bit field with its chip reset default value.
8	EXRST	**External Reset Enable** 0 = The RESOUT signal is not asserted during watchdog timer resets. 1 = The watchdog timer generates a system reset signal on the RESOUT pin whenever a watchdog timer reset event occurs. The EXTRST bit is set by an external reset and is unchanged by a watchdog timer reset.
7–0	ES	**Exponent Select** The ES bit field determines the duration of the watchdog timer timeout interval. The duration is determined by setting one bit in the ES field to select a predefined exponent to use in the following equation: Timeout Duration = 2^{exponent} / CPU Frequency Where Timeout Duration is the timeout period in seconds, exponent is the value selected from the following table, and CPU Frequency is the operating speed of the CPU in Hz. Setting more than one bit in the ES bit field results in the shorter timeout value.

Bit								
7	**6**	**5**	**4**	**3**	**2**	**1**	**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

Programming Notes

The value of the RSTFLAG and EXRST bits after any reset is dependent on whether the reset was generated externally (by asserting $\overline{\text{RES}}$) or caused by a timeout on the watchdog timer. If the reset was generated externally, the RSTFLAG bit is 0 and the EXRST bit is 1. If the reset was generated by the watchdog timer, the RSTFLAG bit is 1 and the EXRST bit is not changed (retains previously programmed value).

Pins that are latched on external reset are only sampled on external reset, not during a watchdog timer reset. These pins include pinstraps and the RESCON register inputs (see page 14-2).

All other Am186CC/CH/CU microcontrollers functions are affected the same by a watchdog timer reset as by the assertion of $\overline{\text{RES}}$.

If the $\overline{\text{RES}}$ signal is asserted while the watchdog timer is performing a watchdog timer reset, the external reset takes precedence.

CHAPTER

16 MISCELLANEOUS CONFIGURATION REGISTERS

16.1 OVERVIEW

This chapter describes the following miscellaneous configuration registers on the Am186CC/CH/CU microcontrollers.

- The System Control register contains system-wide configuration bits.
- The Processor Revision register contains the specific release level of the processor.
- The Relocation register allows software to relocate the peripheral control block to start at any even 1024-byte location in either memory or I/O space.

For more information about system configuration, see the *Am186CC/CH/CU Microcontrollers User's Manual*, order #21914.

Table 16-1 lists these registers in offset order, with the corresponding description's page number.

Table 16-1 Miscellaneous Configuration Register Map

Register Name	Mnemonic	Offset	Page Number
System Configuration	SYSCON	03F0h	page 16-2
Processor Revision Level	PRL	03F4h	page 16-4
Relocation	RELOC	03FEh	page 16-5

System Configuration (SYSCON) Offset 03F0h

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	*Res*		DSDEN	PWD	DISMEM	DISIO	ITF4		EXSYNC	*Res*			DISCLK	*Res*		
Software Read/Write	R		R/W	R/W	R/W	R/W	R/W		R/W	R			R/W	R		
Hardware Set/Clear	—	—	—	—	—	—	—		—	—			—	—	—	—
Chip Reset Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Microcontroller:	
Am186CC	✔
Am186CH	!
Am186CU	!

! = *See bit definitions.*

Register Description

This register contains system-wide configuration bits that affect the operation on a global basis. The SYSCON register is typically written once to establish the proper mode(s) of operation based on the system in which the part is operating.

Bit Definitions

Bit	Name	Function
15–14	Res	**Reserved** For compatibility with future devices, always write this bit field with its chip reset default value.
13	DSDEN	**DS/DEN Select** The DSDEN bit enables the data strobe timings on the DEN pin. 0 = The DEN timing for both reads and writes is normal. 1 = Data Strobe Bus mode is enabled, and the DS timing for reads and writes is identical to the normal read cycle DEN timing. The DEN pin is renamed DS in Data Strobe Bus mode.
12	PWD	**Pulse Width Demodulation** 0 = Pulse Width Demodulation (PWD) mode is disabled. 1 = Pulse Width Demodulation mode is enabled. In PWD mode, Timer 0 and Timer 1 are used to measure the High and Low time of a signal connected to the INT8 pin. Software is notified of the input signal transitions by interrupt 8 (channel 13, type 1Dh) and interrupt 7 (channel 12, type 1Ch). For more information, see the *Am186CC/CH/CU Microcontrollers User's Manual*, order #21914.
11	DISMEM	**Disable Memory Addresses** 0 = Memory addresses are not disabled on the AD15–AD0 bus. 1 = Memory addresses are disabled on the AD15–AD0 bus. Setting the DISMEM bit overrides the clearing of the DA bits in the UMCS and LMCS registers (see page 11-3 and page 11-6). The DISMEM bit has no effect if the ADEN input pin is sampled in its Low state during an external reset (i.e., when the RES input pin transitions from Low to High).
10	DISIO	**Disable I/O Addresses** 0 = I/O addresses are not disabled on the AD15–AD0 bus. 1 = I/O addresses are disabled on the AD15–AD0 bus. The DISIO bit has no effect if the ADEN input pin is sampled in its Low state during an external reset (i.e., when the RES input pin transitions from Low to High).

Bit	Name	Function
9–8	ITF4	**Interface 4 Select** The ITF4 bits configure the interface for external interface D as follows: 00 = Full HDLC with flow control; High-Speed UART without flow control; no UART 01 = HDLC without flow control; High-Speed UART with flow control; no UART 10 = No HDLC; High-Speed UART with flow control; UART with flow control 11 = Reserved ***Am186CH and Am186CU Microcontrollers:*** *HDLC interface D is not supported on the Am186CH and Am186CU microcontrollers. The ITF4 bit field default value is 00b, however, so software must change the value to 10b before using the UART interface or High-Speed UART with flow control.*
7	EXSYNC	**External Codec Synchronization Support** 0 = The HDLC Channel C interface is available for raw DCE or PCM highway operation. 1 = GCI clock and frame information are driven out of HDLC Channel C. If the EXSYNC bit is set, the TSCCON register MODE bit can be configured for GCI time-slotting on channel A (see page 7-3). ***Am186CH and Am186CU Microcontrollers:*** *GCI and HDLC channel C are not supported on the Am186CH and Am186CU microcontrollers.*
6–4	RES	**Reserved** For compatibility with future devices, always write this bit field with its chip reset default value.
3	DISCLK	**Disable Clock** 0 = The CLKOUT pin operates normally. 1 = The CLKOUT pin is disabled and drives a zero externally.
2–0	RES	**Reserved** For compatibility with future devices, always write this bit field with its chip reset default value.

Programming Notes

Processor Revision Level (PRL) Offset 03F4h

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	PRL															
Software Read/Write	R															
Hardware Set/Clear	—															
Chip Reset Default	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Note: *This register's chip reset default value depends on the processor revision level. The PRL default shown is for Revision B0.*

Microcontroller:	
Am186CC	✓
Am186CH	✓
Am186CU	✓

Register Description

The PRL register contains the processor revision level for the device.

Processor revisions are synchronized between the Am186CC, Am186CH, and Am186CU microcontrollers. At a particular revision, the PRL value is the same for all three.

Bit Definitions

Bit	Name	Function
15–0	PRL	**Processor Revision Level** The PRL bit field describes the processor revision level. A larger value indicates a later revision. Contact your AMD representative for current processor level information. The PRL for Revision A0 is 4000h. The PRL for Revision B0 is 4001h.

Programming Notes

Relocation (RELOC) Offset 03FEh

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Name	DUAL	*Res*		M/$\overline{IO}$	R[19–10]										*Res*	
Software Read/Write	R/W	R		R/W	R/W										R	
Hardware Set/Clear	—	—		—	—										—	
Chip Reset Default	0	0	1	0	0	0	0	0	1	1	1	1	1	1	0	0

Microcontroller:

Am186CC ✔

Am186CH ✔

Am186CU ✔

Register Description

This register determines the base address for the internal peripheral control block (PCB, see Chapter 1). The base address must be on an even 1024-byte boundary (i.e., the lower 10 bits of the base address are all 0's). The control block can be mapped into either the memory or I/O space. If mapped into I/O space, the upper four bits of the base address must be programmed as 0 (because I/O addresses are only 16-bits wide). Other chip selects can overlap the control block only if they are programmed to zero wait states and ignore external ready.

Bit Definitions

Bit **Name** **Function**

15 DUAL **Dual Map**

0 = The PCB is decoded only at the programmed location.

1 = The PCB is decoded at the original external or watchdog timer reset location in I/O space as well as at the I/O space in which it is currently programmed to reside.

Do not set the DUAL bit when the M/$\overline{IO}$ bit is set. Setting both the DUAL and M/$\overline{IO}$ bits causes unpredictable results.

14–13 Res **Reserved**

For compatibility with future devices, always write this bit field with its chip reset default value.

12 M/$\overline{IO}$ **Memory/I/O**

0 = The PCB is located in I/O space.

1 =The PCB is located in memory.

Do not set the M/$\overline{IO}$ bit when the DUAL bit is set. Setting both the M/$\overline{IO}$ and DUAL bits causes unpredictable results.

11–2 R[19–10] **Relocation Address**

The R[19–10] bit field determines the base address of the PCB space.

1–0 Res **Reserved**

For compatibility with future devices, always write this bit field with its chip reset default value.

Programming Notes

APPENDIX

A REGISTER SUMMARY

Table A-1, starting on page A-2, provides a summary of all the Am186CC/CH/CU microcontrollers' peripheral control block (PCB) registers, listed in offset order. The table includes the following information for each register:

- Abbreviated name
- Register description page number
- Relative offset from the PCB base (set in RELOC)
- Default location in I/O space (equal to the default PCB base of FC00h plus the register's relative offset)
- Default value at reset
- Bit and field names and layout

An "x" in the default value column denotes a digit for which the default value is not defined. A "?" indicates that the digit's value depends on external inputs. If a digit contains both undefined and external input bits, a "?" is used.

If more than one default value is given for a register, it contains one or more bits with undefined defaults. In this case either value might be present.

If a group of registers is not supported on all the Am186CC/CH/CU microcontrollers, the group heading indicates the controllers that support that group of registers. An exclamation point (**!**) following a specific bit or register name indicates that additional controller-specific information can be found in the individual register or bit description.

Table A-1 Am186CC/CH/CU Microcontrollers Register Summary

NAME	SEE PAGE	OFFSET	DEFAULT LOCATION	DEFAULT VALUE	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HDLC Channel A Registers (Am186CC and Am186CH Microcontrollers Only)																				
HACON	2-5	00h	FC00h	0000h	*Res*								HRESET	*Res*	NRZI	TRANSM	LOOPR	LOOPL	CRCTYPE	
HATCON0	2-7	02h	FC02h	0000h	*Res*				TTHRSH		*Res*			TFIFOEN	FORABR	HTEN	IMSTART	CRCDIS	LBREAD	LBNOW
HATCON1	2-9	04h	FC04h	0000h	*Res*				FLAGIDL	MLTDRP	AUTOCTS	TMSBF	TXCINV	GCIDEN	ODRV		TDELAY			
HARCON0	2-11	06h	FC06h	0000h	*Res*				RTHRSH		RCPST	RMSBF	RXCINV	RREJECT	RSTOP	HREN	MINRL			
HARCON1	2-13	08h	FC08h	0000h	MAXRL															
HASTATE	2-14	0Ah	FC0Ah	0010h 0030h	*Res*										CTSS	RTRS	ABORTS	MARKIS	FLAGS	FRAMES
HAISTAT0	2-16	0Ch	FC0Ch	0000h	*Res*			REOF	RTHRES	RDATA1	TTHRES	TDATA1	*Res*			FABRST	CTSLST	TUFLO	TGOODF	TSTOP
HAIMSK0	2-18	0Eh	FC0Eh	0000h	*Res*			REOF	RTHRES	RDATA1	TTHRES	TDATA1	*Res*			FABRST	CTSLST	TUFLO	TGOODF	TSTOP
HAISTAT1	2-20	10h	FC10h	0000h	*Res*						MAMC	SFMC	SHORT	VSHORT	RTRDES	ROFLO	ABORTE	MARKIE	FLAGE	FRAMEE
HAIMSK1	2-22	12h	FC12h	0000h	*Res*						MAMC	SFMC	SHORT	VSHORT	RTRDES	ROFLO	ABORTE	MARKIE	FLAGE	FRAMEE
HATD	2-24	14h	FC14h	00xxh	*Res*								TDATA							
HARD	2-25	16h	FC16h	00xxh	STAT1A	STAT0A	STATNUM		RTHRES	RDATA1	TTHRES	TDATA1	RDATA							
HARFS1	2-29	(16h)	(FC16h)	00xxh	STAT1A	STAT0A	STATNUM		RTHRES	RDATA1	TTHRES	TDATA1	FBCNT[7–0]							
HARFS2	2-31	(16h)	(FC16h)	00xxh	STAT1A	STAT0A	STATNUM		RTHRES	RDATA1	TTHRES	TDATA1	FBCNT[15–8]							
HARFS3	2-33	(16h)	(FC16h)	00xxh	STAT1A	STAT0A	STATNUM		RTHRES	RDATA1	TTHRES	TDATA1	FRAM	FOFLO	CRCE	MTCH		FABORT	FLONG	FSHORT
HARDP	2-27	18h	FC18h	00xxh	STAT1A	STAT0A	STATNUM		RTHRES	RDATA1	TTHRES	TDATA1	RDATA							
HASFCNT	2-35	1Ah	FC1Ah	0000h	HSFCNT															
HASFCNTP	2-36	1Ch	FC1Ch	0000h	HSFCNTP															
HAMACNT	2-37	1Eh	FC1Eh	0000h	HMACNT															
HAMACNTP	2-38	20h	FC20h	0000h	HMACNTP															
HAA0	2-39	22h	FC22h	0000h	HA															
HAA0MSK	2-41	24h	FC24h	0000h	HAMSK															
HAA1	2-39	26h	FC26h	0000h	HA															
HAA1MSK	2-41	28h	FC28h	0000h	HAMSK															
HAA2	2-39	2Ah	FC2Ah	0000h	HA															
HAA2MSK	2-41	2Ch	FC2Ch	0000h	HAMSK															
HAA3	2-39	2Eh	FC2Eh	0000h	HA															
HAA3MSK	2-41	30h	FC30h	0000h	HAMSK															
NAME	SEE PAGE	OFFSET	DEFAULT LOCATION	DEFAULT VALUE	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

! *= See register or bit description for controller-specific details.*

Table A-1 Am186CC/CH/CU Microcontrollers Register Summary (Continued)

NAME	SEE PAGE	OFFSET	DEFAULT LOCATION	DEFAULT VALUE	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HDLC Channel B Registers (Am186CC and Am186CH Microcontrollers Only)																				
HBCON	2-5	40h	FC40h	0000h	*Res*								HRESET	*Res*	NRZI	TRANSM	LOOPR	LOOPL	CRCTYPE	
HBTCON0	2-7	42h	FC42h	0000h	*Res*				TTHRSH		*Res*			TFIFOEN	FORABR	HTEN	IMSTART	CRCDIS	LBREAD	LBNOW
HBTCON1	2-9	44h	FC44h	0000h	*Res*				FLAGIDL	MLTDRP	AUTOCTS	TMSBF	TXCINV	GCIDEN	ODRV		TDELAY			
HBRCON0	2-11	46h	FC46h	0000h	*Res*				RTHRSH		RCPST	RMSBF	RXCINV	RREJECT	RSTOP	HREN	MINRL			
HBRCON1	2-13	48h	FC48h	0000h	MAXRL															
HBSTATE	2-14	4Ah	FC4Ah	0010h 0030h	*Res*										CTSS	RTRS	ABORTS	MARKIS	FLAGS	FRAMES
HBISTAT0	2-16	4Ch	FC4Ch	0000h	*Res*			REOF	RTHRES	RDATA1	TTHRES	TDATA1	*Res*			FABRST	CTSLST	TUFLO	TGOODF	TSTOP
HBIMSK0	2-18	4Eh	FC4Eh	0000h	*Res*			REOF	RTHRES	RDATA1	TTHRES	TDATA1	*Res*			FABRST	CTSLST	TUFLO	TGOODF	TSTOP
HBISTAT1	2-20	50h	FC50h	0000h	*Res*						MAMC	SFMC	SHORT	VSHORT	RTRDES	ROFLO	ABORTE	MARKIE	FLAGE	FRAMEE
HBIMSK1	2-22	52h	FC52h	0000h	*Res*						MAMC	SFMC	SHORT	VSHORT	RTRDES	ROFLO	ABORTE	MARKIE	FLAGE	FRAMEE
HBTD	2-24	54h	FC54h	00xxh	*Res*								TDATA							
HBRD	2-25	56h	FC56h	00xxh	STAT1A	STAT0A	STATNUM		RTHRES	RDATA1	TTHRES	TDATA1	RDATA							
HBRFS1	2-29	(56h)	(FC56h)	00xxh	STAT1A	STAT0A	STATNUM		RTHRES	RDATA1	TTHRES	TDATA1	FBCNT[7–0]							
HBRFS2	2-31	(56h)	(FC56h)	00xxh	STAT1A	STAT0A	STATNUM		RTHRES	RDATA1	TTHRES	TDATA1	FBCNT[15–8]							
HBRFS3	2-33	(56h)	(FC56h)	00xxh	STAT1A	STAT0A	STATNUM		RTHRES	RDATA1	TTHRES	TDATA1	FRAM	FOFLO	CRCE	MTCH		FABORT	FLONG	FSHORT
HBRDP	2-27	58h	FC58h	00xxh	STAT1A	STAT0A	STATNUM		RTHRES	RDATA1	TTHRES	TDATA1	RDATA							
HBSFCNT	2-35	5Ah	FC5Ah	0000h	HSFCNT															
HBSFCNTP	2-36	5Ch	FC5Ch	0000h	HSFCNTP															
HBMACNT	2-37	5Eh	FC5Eh	0000h	HMACNT															
HBMACNTP	2-38	60h	FC60h	0000h	HMACNTP															
HBA0	2-39	62h	FC62h	0000h	HA															
HBA0MSK	2-41	64h	FC64h	0000h	HAMSK															
HBA1	2-39	66h	FC66h	0000h	HA															
HBA1MSK	2-41	68h	FC68h	0000h	HAMSK															
HBA2	2-39	6Ah	FC6Ah	0000h	HA															
HBA2MSK	2-41	6Ch	FC6Ch	0000h	HAMSK															
HBA3	2-39	6Eh	FC6Eh	0000h	HA															
HBA3MSK	2-41	70h	FC70h	0000h	HAMSK															
NAME	SEE PAGE	OFFSET	DEFAULT LOCATION	DEFAULT VALUE	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

! *= See register or bit description for controller-specific details.*

Table A-1 Am186CC/CH/CU Microcontrollers Register Summary (Continued)

NAME	SEE PAGE	OFFSET	DEFAULT LOCATION	DEFAULT VALUE	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HDLC Channel C Registers (Am186CC Microcontroller Only)																				
HCCON	2-5	80h	FC80h	0000h	*Res*								HRESET	*Res*	NRZI	TRANSM	LOOPR	LOOPL	CRCTYPE	
HCTCON0	2-7	82h	FC82h	0000h	*Res*				TTHRSH		*Res*			TFIFOEN	FORABR	HTEN	IMSTART	CRCDIS	LBREAD	LBNOW
HCTCON1	2-9	84h	FC84h	0000h	*Res*				FLAGIDL	MLTDRP	AUTOCTS	TMSBF	TXCINV	GCIDEN	ODRV		TDELAY			
HCRCON0	2-11	86h	FC86h	0000h	*Res*				RTHRSH		RCPST	RMSBF	RXCINV	RREJECT	RSTOP	HREN	MINRL			
HCRCON1	2-13	88h	FC88h	0000h	MAXRL															
HCSTATE	2-14	8Ah	FC8Ah	0010h 0030h	*Res*										CTSS	RTRS	ABORTS	MARKIS	FLAGS	FRAMES
HCISTAT0	2-16	8Ch	FC8Ch	0000h	*Res*			REOF	RTHRES	RDATA1	TTHRES	TDATA1	*Res*			FABRST	CTSLST	TUFLO	TGOODF	TSTOP
HCIMSK0	2-18	8Eh	FC8Eh	0000h	*Res*			REOF	RTHRES	RDATA1	TTHRES	TDATA1	*Res*			FABRST	CTSLST	TUFLO	TGOODF	TSTOP
HCISTAT1	2-20	90h	FC90h	0000h	*Res*						MAMC	SFMC	SHORT	VSHORT	RTRDES	ROFLO	ABORTE	MARKIE	FLAGE	FRAMEE
HCIMSK1	2-22	92h	FC92h	0000h	*Res*						MAMC	SFMC	SHORT	VSHORT	RTRDES	ROFLO	ABORTE	MARKIE	FLAGE	FRAMEE
HCTD	2-24	94h	FC94h	00xxh	*Res*								TDATA							
HCRD	2-25	96h	FC96h	00xxh	STAT1A	STAT0A	STATNUM		RTHRES	RDATA1	TTHRES	TDATA1	RDATA							
HCRFS1	2-29	(96h)	(FC96h)	00xxh	STAT1A	STAT0A	STATNUM		RTHRES	RDATA1	TTHRES	TDATA1	FBCNT[7–0]							
HCRFS2	2-31	(96h)	(FC96h)	00xxh	STAT1A	STAT0A	STATNUM		RTHRES	RDATA1	TTHRES	TDATA1	FBCNT[15–8]							
HCRFS3	2-33	(96h)	(FC96h)	00xxh	STAT1A	STAT0A	STATNUM		RTHRES	RDATA1	TTHRES	TDATA1	FRAM	FOFLO	CRCE	MTCH		FABORT	FLONG	FSHORT
HCRDP	2-27	98h	FC98h	00xxh	STAT1A	STAT0A	STATNUM		RTHRES	RDATA1	TTHRES	TDATA1	RDATA							
HCSFCNT	2-35	9Ah	FC9Ah	0000h	HSFCNT															
HCSFCNTP	2-36	9Ch	FC9Ch	0000h	HSFCNTP															
HCMACNT	2-37	9Eh	FC9Eh	0000h	HMACNT															
HCMACNTP	2-38	A0h	FCA0h	0000h	HMACNTP															
HCA0	2-39	A2h	FCA2h	0000h	HA															
HCA0MSK	2-41	A4h	FCA4h	0000h	HAMSK															
HCA1	2-39	A6h	FCA6h	0000h	HA															
HCA1MSK	2-41	A8h	FCA8h	0000h	HAMSK															
HCA2	2-39	AAh	FCAAh	0000h	HA															
HCA2MSK	2-41	ACh	FCACh	0000h	HAMSK															
HCA3	2-39	AEh	FCAEh	0000h	HA															
HCA3MSK	2-41	B0h	FCB0h	0000h	HAMSK															
NAME	SEE PAGE	OFFSET	DEFAULT LOCATION	DEFAULT VALUE	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

! = See register or bit description for controller-specific details.

Table A-1 Am186CC/CH/CU Microcontrollers Register Summary (Continued)

NAME	SEE PAGE	OFFSET	DEFAULT LOCATION	DEFAULT VALUE	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HDLC Channel D Registers (Am186CC Microcontroller Only)																				
HDCON	2-5	C0h	FCC0h	0000h	*Res*								HRESET	*Res*	NRZI	TRANSM	LOOPR	LOOPL	CRCTYPE	
HDTCON0	2-7	C2h	FCC2h	0000h	*Res*				TTHRSH		*Res*			TFIFOEN	FORABR	HTEN	IMSTART	CRCDIS	LBREAD	LBNOW
HDTCON1	2-9	C4h	FCC4h	0000h	*Res*				FLAGIDL	MLTDRP	AUTOCTS	TMSBF	TXCINV	GCIDEN	ODRV		TDELAY			
HDRCON0	2-11	C6h	FCC6h	0000h	*Res*				RTHRSH		RCPST	RMSBF	RXCINV	RREJECT	RSTOP	HREN	MINRL			
HDRCON1	2-13	C8h	FCC8h	0000h	MAXRL															
HDSTATE	2-14	CAh	FCCAh	0010h 0030h	*Res*										CTSS	RTRS	ABORTS	MARKIS	FLAGS	FRAMES
HDISTAT0	2-16	CCh	FCCCh	0000h	*Res*			REOF	RTHRES	RDATA1	TTHRES	TDATA1	*Res*			FABRST	CTSLST	TUFLO	TGOODF	TSTOP
HDIMSK0	2-18	CEh	FCCEh	0000h	*Res*			REOF	RTHRES	RDATA1	TTHRES	TDATA1	*Res*			FABRST	CTSLST	TUFLO	TGOODF	TSTOP
HDISTAT1	2-20	D0h	FCD0h	0000h	*Res*						MAMC	SFMC	SHORT	VSHORT	RTRDES	ROFLO	ABORTE	MARKIE	FLAGE	FRAMEE
HDIMSK1	2-22	D2h	FCD2h	0000h	*Res*						MAMC	SFMC	SHORT	VSHORT	RTRDES	ROFLO	ABORTE	MARKIE	FLAGE	FRAMEE
HDTD	2-24	D4h	FCD4h	00xxh	*Res*								TDATA							
HDRD	2-25	D6h	FCD6h	00xxh	STAT1A	STAT0A	STATNUM		RTHRES	RDATA1	TTHRES	TDATA1	RDATA							
HDRFS1	2-29	(D6h)	(FCD6h)	00xxh	STAT1A	STAT0A	STATNUM		RTHRES	RDATA1	TTHRES	TDATA1	FBCNT[7–0]							
HDRFS2	2-31	(D6h)	(FCD6h)	00xxh	STAT1A	STAT0A	STATNUM		RTHRES	RDATA1	TTHRES	TDATA1	FBCNT[15–8]							
HDRFS3	2-33	(D6h)	(FCD6h)	00xxh	STAT1A	STAT0A	STATNUM		RTHRES	RDATA1	TTHRES	TDATA1	FRAM	FOFLO	CRCE	MTCH		FABORT	FLONG	FSHORT
HDRDP	2-27	D8h	FCD8h	00xxh	STAT1A	STAT0A	STATNUM		RTHRES	RDATA1	TTHRES	TDATA1	RDATA							
HDSFCNT	2-35	DAh	FCDAh	0000h	HSFCNT															
HDSFCNTP	2-36	DCh	FCDCh	0000h	HSFCNTP															
HDMACNT	2-37	DEh	FCDEh	0000h	HMACNT															
HDMACNTP	2-38	E0h	FCE0h	0000h	HMACNTP															
HDA0	2-39	E2h	FCE2h	0000h	HA															
HDA0MSK	2-41	E4h	FCE4h	0000h	HAMSK															
HDA1	2-39	E6h	FCE6h	0000h	HA															
HDA1MSK	2-41	E8h	FCE8h	0000h	HAMSK															
HDA2	2-39	EAh	FCEAh	0000h	HA															
HDA2MSK	2-41	ECh	FCECh	0000h	HAMSK															
HDA3	2-39	EEh	FCEEh	0000h	HA															
HDA3MSK	2-41	F0h	FCF0h	0000h	HAMSK															
NAME	SEE PAGE	OFFSET	DEFAULT LOCATION	DEFAULT VALUE	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

! = *See register or bit description for controller-specific details.*

Table A-1 Am186CC/CH/CU Microcontrollers Register Summary (Continued)

NAME	SEE PAGE	OFFSET	DEFAULT LOCATION	DEFAULT VALUE	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
General-Purpose DMA Channel 0 Registers																				
GD0CON0	3-4	100h	FD00h	0000h	ST	AST	TC	INT	*Res*		P		*Res*	TS	*Res*	DSEL !				
GD0CON1	3-7	102h	FD02h	0000h	SM/$\overline{\text{IO}}$	SAW			SINC				DM/$\overline{\text{IO}}$	DAW			DINC			
GD0SRCL	3-9	104h	FD04h	0000h	DSA[15–0]															
GD0SRCH	3-10	106h	FD06h	0000h	*Res*												DSA[19–16]			
GD0DSTL	3-11	108h	FD08h	0000h	DDA[15–0]															
GD0DSTH	3-12	10Ah	FD0Ah	0000h	*Res*												DDA[19–16]			
GD0TC	3-13	10Ch	FD0Ch	0000h	TC															
General-Purpose DMA Channel 1 Registers																				
GD1CON0	3-4	110h	FD10h	0000h	ST	AST	TC	INT	*Res*		P		*Res*	TS	*Res*	DSEL !				
GD1CON1	3-7	112h	FD12h	0000h	SM/$\overline{\text{IO}}$	SAW			SINC				DM/$\overline{\text{IO}}$	DAW			DINC			
GD1SRCL	3-9	114h	FD14h	0000h	DSA[15–0]															
GD1SRCH	3-10	116h	FD16h	0000h	*Res*												DSA[19–16]			
GD1DSTL	3-11	118h	FD18h	0000h	DDA[15–0]															
GD1DSTH	3-12	11Ah	FD1Ah	0000h	*Res*												DDA[19–16]			
GD1TC	3-13	11Ch	FD1Ch	0000h	TC															
General-Purpose DMA Channel 2 Registers																				
GD2CON0	3-4	120h	FD20h	0000h	ST	AST	TC	INT	*Res*		P		*Res*	TS	*Res*	DSEL !				
GD2CON1	3-7	122h	FD22h	0000h	SM/$\overline{\text{IO}}$	SAW			SINC				DM/$\overline{\text{IO}}$	DAW			DINC			
GD2SRCL	3-9	124h	FD24h	0000h	DSA[15–0]															
GD2SRCH	3-10	126h	FD26h	0000h	*Res*												DSA[19–16]			
GD2DSTL	3-11	128h	FD28h	0000h	DDA[15–0]															
GD2DSTH	3-12	12Ah	FD2Ah	0000h	*Res*												DDA[19–16]			
GD2TC	3-13	12Ch	FD2Ch	0000h	TC															
General-Purpose DMA Channel 3 Registers																				
GD3CON0	3-4	130h	FD30h	0000h	ST	AST	TC	INT	*Res*		P		*Res*	TS	*Res*	DSEL !				
GD3CON1	3-7	132h	FD32h	0000h	SM/$\overline{\text{IO}}$	SAW			SINC				DM/$\overline{\text{IO}}$	DAW			DINC			
GD3SRCL	3-9	134h	FD34h	0000h	DSA[15–0]															
GD3SRCH	3-10	136h	FD36h	0000h	*Res*												DSA[19–16]			
GD3DSTL	3-11	138h	FD38h	0000h	DDA[15–0]															
GD3DSTH	3-12	13Ah	FD3Ah	0000h	*Res*												DDA[19–16]			
NAME	SEE PAGE	OFFSET	DEFAULT LOCATION	DEFAULT VALUE	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

! *= See register or bit description for controller-specific details.*

Table A-1 Am186CC/CH/CU Microcontrollers Register Summary (Continued)

NAME	SEE PAGE	OFFSET	DEFAULT LOCATION	DEFAULT VALUE	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GD3TC	3-13	13Ch	FD3Ch	0000h	TC															
SmartDMA Channel Pair 0 Registers (Am186CC and Am186CH Microcontrollers Only)																				
SD0CON	3-14	140h	FD40h	0000h	*Res*		TEPI	TBUI	TTCI	REPI	RBUI	RTCI	TXSO	RXSO	P		POLL	*Res*	TXST	RXST
SD0TRCAL	3-20	142h	FD42h	0000h	TRA[15–4]												*Res*	TRC		
SD0TRAH	3-21	144h	FD44h	0000h	*Res*												TRA[19–16]			
SD0RRCAL	3-22	146h	FD46h	0000h	RRA[15–4]												*Res*	RRC		
SD0RRAH	3-23	148h	FD48h	0000h	*Res*												RRA[19–16]			
SD0STAT	3-24	14Ah	FD4Ah	0000h	*Res*		TEP	TBU	TTC	REP	RBU	RTC	*Res*							
SD0CBD	3-26	14Ch	FD4Ch	0000h	*Res*	CRBD							*Res*	CTBD						
SD0CTAD	3-27	14Eh	FD4Eh	0000h	CTAD															
SD0CRAD	3-28	150h	FD50h	0000h	CRAD															
SmartDMA Channel Pair 1 Registers (Am186CC and Am186CH Microcontrollers Only)																				
SD1CON	3-14	158h	FD58h	0000h	*Res*		TEPI	TBUI	TTCI	REPI	RBUI	RTCI	TXSO	RXSO	P		POLL	*Res*	TXST	RXST
SD1TRCAL	3-20	15Ah	FD5Ah	0000h	TRA[15–4]												*Res*	TRC		
SD1TRAH	3-21	15Ch	FD5Ch	0000h	*Res*												TRA[19–16]			
SD1RRCAL	3-22	15Eh	FD5Eh	0000h	RRA[15–4]												*Res*	RRC		
SD1RRAH	3-23	160h	FD60h	0000h	*Res*												RRA[19–16]			
SD1STAT	3-24	162h	FD62h	0000h	*Res*		TEP	TBU	TTC	REP	RBU	RTC	*Res*							
SD1CBD	3-26	164h	FD64h	0000h	*Res*	CRBD							*Res*	CTBD						
SD1CTAD	3-27	166h	FD66h	0000h	CTAD															
SD1CRAD	3-28	168h	FD68h	0000h	CRAD															
SmartDMA Channel Pair 2 Registers (Am186CC and Am186CU Microcontrollers Only)																				
SD2CON	3-17	170h	FD70h	0000h	*Res*		TEPI	TBUI	TTCI	REPI	RBUI	RTCI	TXSO	RXSO	P		POLL	DSEL **!**	TXST	RXST
SD2TRCAL	3-20	172h	FD72h	0000h	TRA[15–4]												*Res*	TRC		
SD2TRAH	3-21	174h	FD74h	0000h	*Res*												TRA[19–16]			
SD2RRCAL	3-22	176h	FD76h	0000h	RRA[15–4]												*Res*	RRC		
SD2RRAH	3-23	178h	FD78h	0000h	*Res*												RRA[19–16]			
SD2STAT	3-24	17Ah	FD7Ah	0000h	*Res*		TEP	TBU	TTC	REP	RBU	RTC	*Res*							
SD2CBD	3-26	17Ch	FD7Ch	0000h	*Res*	CRBD							*Res*	CTBD						
SD2CTAD	3-27	17Eh	FD7Eh	0000h	CTAD															
SD2CRAD	3-28	180h	FD80h	0000h	CRAD															
NAME	SEE PAGE	OFFSET	DEFAULT LOCATION	DEFAULT VALUE	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

! *= See register or bit description for controller-specific details.*

Table A-1 Am186CC/CH/CU Microcontrollers Register Summary (Continued)

NAME	SEE PAGE	OFFSET	DEFAULT LOCATION	DEFAULT VALUE	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SmartDMA Channel Pair 3 Registers (Am186CC and Am186CU Microcontrollers Only)																				
SD3CON	3-17	188h	FD88h	0000h	*Res*		TEPI	TBUI	TTCI	REPI	RBUI	RTCI	TXSO	RXSO	P		POLL	DSEL !	TXST	RXST
SD3TRCAL	3-20	18Ah	FD8Ah	0000h	TRA[15–4]												*Res*	TRC		
SD3TRAH	3-21	18Ch	FD8Ch	0000h	*Res*												TRA[19–16]			
SD3RRCAL	3-22	18Eh	FD8Eh	0000h	RRA[15–4]												*Res*	RRC		
SD3RRAH	3-23	190h	FD90h	0000h	*Res*												RRA[19–16]			
SD3STAT	3-24	192h	FD92h	0000h	*Res*		TEP	TBU	TTC	REP	RBU	RTC	*Res*							
SD3CBD	3-26	194h	FD94h	0000h	*Res*	CRBD							*Res*	CTBD						
SD3CTAD	3-27	196h	FD96h	0000h	CTAD															
SD3CRAD	3-28	198h	FD98h	0000h	CRAD															
Universal Serial Bus (USB) General Configuration Registers (Am186CC and Am186CU Microcontrollers Only)																				
UISTAT1	4-3	1E0h	FDE0h	0000h	*Res*				D_EP_STATINT	D_EP_ACT	C_EP_STATINT	C_EP_ACT	B_EP_STATINT	B_EP_ACT	A_EP_STATINT	A_EP_ACT	OTHER_INT	INT_EP_ACT	CNT_EP_NEW	CNT_EP_ACT
UIMASK1	4-5	1E2h	FDE2h	0008h	*Res*				D_EP_STATINT	D_EP_ACT	C_EP_STATINT	C_EP_ACT	B_EP_STATINT	B_EP_ACT	A_EP_STATINT	A_EP_ACT	OI_UNM	INT_EP_ACT	CNT_EP_NEW	CNT_EP_ACT
UISTAT2	4-7	1E4h	FDE4h	0000h	USB_RST	USB_SUS	USB_RES	*Res*									TSTMP_M	POS_UP	SOF_GEN	MS_SOF
UIMASK2	4-9	1E6h	FDE6h	0000h	USB_RST	USB_SUS	USB_RES	*Res*									TSTMP_M	POS_UP	SOF_GEN	MS_SOF
USBMFR	4-10	1E8h	FDE8h	0008h	*Res*									PUP_XCVER	SUSP	S_RES	S_POWER	DIS_XCVER	RWAKE	RWAKE_EN
RTFMCNT	4-12	1EAh	FDEAh	0000h	*Res*		RTFCNT													
TSTMP	4-13	1ECh	FDECh	0000h	*Res*					TSTMP										
TSTMPM	4-14	1EEh	FDEEh	0000h	*Res*					TSTMPM										
ISCTL	4-15	1F0h	FDF0h	0000h	ESOF_EN	*Res*					BYTES_SAM		*Res*			BCNT_LRATE			SAM_CLK_SEL !	
FPMCNT	4-17	1F2h	FDF2h	0000h	*Res*		FPM_CNT													
USB Control Endpoint Registers (Am186CC and Am186CU Microcontrollers Only)																				
CNTCTL	4-18	200h	FE00h	0000h	EP_EN	EP_NOT_STALLED	NOT_FLUSH	ACT_REQ	NEW_COMMAND	COMMAND_BUSY	*Res*									
CNTSIZ	4-21	202h	FE02h	0000h	*Res*												RCV_PKT_SIZE			
CNTDAT	4-22	206h	FE06h	00xxh	*Res*								D							
CNTRPK	4-23	208h	FE08h	00xxh	*Res*								D							
CNTDEF1	4-24	20Ah	FE0Ah	0000h	EP_NUM				EP_CFG		*Res*	EP_INT		*Res*	EP_ASET			EP_DIR	EP_TYPE	
CNTDEF2	4-26	20Ch	FE0Ch	0008h	*Res*				FIFO_SIZE		EP_MX_PCT									
NAME	SEE PAGE	OFFSET	DEFAULT LOCATION	DEFAULT VALUE	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

! = See register or bit description for controller-specific details.

Table A-1 Am186CC/CH/CU Microcontrollers Register Summary (Continued)

NAME	SEE PAGE	OFFSET	DEFAULT LOCATION	DEFAULT VALUE	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
USB Interrupt Endpoint Registers (Am186CC and Am186CU Microcontrollers Only)																				
IEPCTL	4-27	210h	FE10h	0000h	EP_EN	EP_NOT_STALLED	NOT_FLUSH	ACT_REQ	*Res*											
IEPDAT	4-29	216h	FE16h	00xxh	*Res*								D							
IEPDEF1	4-30	21Ah	FE1Ah	1003h	EP_NUM				EP_CFG		*Res*	EP_INT		*Res*	EP_ASET			EP_DIR	EP_TYPE	
IEPDEF2	4-32	21Ch	FE1Ch	0410h	*Res*				FIFO_SIZE		EP_MX_PCT									
USB Data A Endpoint Registers (Am186CC and Am186CU Microcontrollers Only)																				
AEPCTL	4-33	220h	FE20h	0000h	EP_EN	EP_NOT_STALLED	NOT_FLUSH	ACT_REQ	STAT_INT	*Res*	NOT_ZERO	NOT_LAST_BYTE	*Res*	ISO_START	ISO_STOP	ISO_MS	FULL_PKT	SHORT_PKT	BUF_ERR	OTHER_ERR
AEPSIZ	4-39	222h	FE22h	0000h	*Res*						RPS									
AEPBUFS	4-40	224h	FE24h	0000h	*Res*											BUF_STAT				
AEPDAT	4-42	226h	FE26h	00xxh	*Res*								D							
ARCVPK	4-43	228h	FE28h	00xxh	*Res*								D							
AEPDEF1	4-44	22Ah	FE2Ah	2006h	EP_NUM				EP_CFG		*Res*	EP_INT		*Res*	EP_ASET			EP_DIR	EP_TYPE	
AEPDEF2	4-46	22Ch	FE2Ch	0408h	*Res*					FIFO_SIZE	EP_MX_PCT									
AEPDEF3	4-50	22Eh	FE2Eh	0018h	*Res*		AUTO_RATE_EN	ISO_MS_IMSK	FULL_PKT_IMSK	SHRT_PKT_IMSK	BUF_ERR_IMSK	OTH_ERR_IMSK	MODE			ISO_MS_SMSK	FULL_PKT_SMSK	SHRT_PKT_SMSK	BUF_ERR_SMSK	OTH_ERR_SMSK
USB Data B Endpoint Registers (Am186CC and Am186CU Microcontrollers Only)																				
BEPCTL	4-33	230h	FE30h	0000h	EP_EN	EP_NOT_STALLED	NOT_FLUSH	ACT_REQ	STAT_INT	*Res*	NOT_ZERO	NOT_LAST_BYTE	*Res*	ISO_START	ISO_STOP	ISO_MS	FULL_PKT	SHORT_PKT	BUF_ERR	OTHER_ERR
BEPSIZ	4-39	232h	FE32h	0000h	*Res*						RPS									
BEPBUFS	4-40	234h	FE34h	0000h	*Res*											BUF_STAT				
BEPDAT	4-42	236h	FE36h	00xxh	*Res*								D							
BRCVPK	4-43	238h	FE38h	00xxh	*Res*								D							
BEPDEF1	4-44	23Ah	FE3Ah	3006h	EP_NUM				EP_CFG		*Res*	EP_INT		*Res*	EP_ASET			EP_DIR	EP_TYPE	
BEPDEF2	4-46	23Ch	FE3Ch	0408h	*Res*					FIFO_SIZE	EP_MX_PCT									
BEPDEF3	4-50	23Eh	FE3Eh	0018h	*Res*		AUTO_RATE_EN	ISO_MS_IMSK	FULL_PKT_IMSK	SHRT_PKT_IMSK	BUF_ERR_IMSK	OTH_ERR_IMSK	MODE			ISO_MS_SMSK	FULL_PKT_SMSK	SHRT_PKT_SMSK	BUF_ERR_SMSK	OTH_ERR_SMSK
NAME	SEE PAGE	OFFSET	DEFAULT LOCATION	DEFAULT VALUE	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

! *= See register or bit description for controller-specific details.*

Table A-1 Am186CC/CH/CU Microcontrollers Register Summary (Continued)

NAME	SEE PAGE	OFFSET	DEFAULT LOCATION	DEFAULT VALUE	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
USB Data C Endpoint Registers (Am186CC and Am186CU Microcontrollers Only)																				
CEPCTL	4-33	240h	FE40h	0000h	EP_EN	EP_NOT_ STALLED	NOT_ FLUSH	ACT_REQ	STAT_INT	*Res*	NOT_ ZERO	NOT_ LAST_ BYTE	*Res*	ISO_ START	ISO_ STOP	ISO_MS	FULL_ PKT	SHORT_ PKT	BUF_ERR	OTHER_ ERR
CEPSIZ	4-39	242h	FE42h	0000h	*Res*						RPS									
CEPBUFS	4-41	244h	FE44h	0000h	*Res*									BUF_STAT						
CEPDAT	4-42	246h	FE46h	00xxh	*Res*								D							
CRCVPK	4-43	248h	FE48h	00xxh	*Res*								D							
CEPDEF1	4-44	24Ah	FE4Ah	4006h	EP_NUM				EP_CFG		*Res*	EP_INT		*Res*	EP_ASET			EP_DIR	EP_TYPE	
CEPDEF2	4-48	24Ch	FE4Ch	0C08h	*Res*				FIFO_SIZE		EP_MX_PCT									
CEPDEF3	4-50	24Eh	FE4Eh	0018h	*Res*		AUTO_ RATE_EN	ISO_MS_ IMSK	FULL_PKT_ IMSK	SHRT_PKT_ IMSK	BUF_ERR_ IMSK	OTH_ERR_ IMSK	MODE			ISO_MS_ SMSK	FULL_PKT_ SMSK	SHRT_PKT_ SMSK	BUF_ERR_ SMSK	OTH_ERR_ SMSK
USB Data D Endpoint Registers (Am186CC and Am186CU Microcontrollers Only)																				
DEPCTL	4-33	250h	FE50h	0000h	EP_EN	EP_NOT_ STALLED	NOT_ FLUSH	ACT_REQ	STAT_INT	*Res*	NOT_ ZERO	NOT_ LAST_ BYTE	*Res*	ISO_ START	ISO_ STOP	ISO_MS	FULL_ PKT	SHORT_ PKT	BUF_ERR	OTHER_ ERR
DEPSIZ	4-39	252h	FE52h	0000h	*Res*						RPS									
DEPBUFS	4-41	254h	FE54h	0000h	*Res*									BUF_STAT						
DEPDAT	4-42	256h	FE56h	00xxh	*Res*								D							
DRCVPK	4-43	258h	FE58h	00xxh	*Res*								D							
DEPDEF1	4-44	25Ah	FE5Ah	5006h	EP_NUM				EP_CFG		*Res*	EP_INT		*Res*	EP_ASET			EP_DIR	EP_TYPE	
DEPDEF2	4-48	25Ch	FE5Ch	0C08h	*Res*				FIFO_SIZE		EP_MX_PCT									
DEPDEF3	4-50	25Eh	FE5Eh	0018h	*Res*		AUTO_ RATE_EN	ISO_MS_ IMSK	FULL_PKT_ IMSK	SHRT_PKT_ IMSK	BUF_ERR_ IMSK	OTH_ERR_ IMSK	MODE			ISO_MS_ SMSK	FULL_PKT_ SMSK	SHRT_PKT_ SMSK	BUF_ERR_ SMSK	OTH_ERR_ SMSK
NAME	SEE PAGE	OFFSET	DEFAULT LOCATION	DEFAULT VALUE	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

! = See register or bit description for controller-specific details.

Table A-1 Am186CC/CH/CU Microcontrollers Register Summary (Continued)

NAME	SEE PAGE	OFFSET	DEFAULT LOCATION	DEFAULT VALUE	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
High-Speed Asynchronous Serial Port (High-Speed UART) Registers																				
HSPCON0	5-3	260h	FE60h	0000h	*Res*			RSIE	BRK	AB	FC	TXIE	RXIE	TMODE	RMODE	EVN	PEN	ABEN	D7	STP2
HSPCON1	5-5	262h	FE62h	0000h	TFEN	RFEN	TFLUSH	RFLUSH	ABAUD	*Res*			MEN	MAB2	MAB1	MAB0	BRKVAL	EXDWR	EXDRD	XTRN !
HSPSTAT	5-7	264h	FE64h	0000h	RTHRSH	TTHRSH	*Res*	OERIM	*Res*	MATCH	BRK	AB	RDR	THRE	FER	OER	PER	TEMT	IDLED	IDLE
HSPIMSK	5-10	266h	FE66h	02F8h	RTHRSH	TTHRSH	*Res*	OERIM	*Res*	MATCH	BRK	AB	RDR	THRE	FER	OER	PER	TEMT	IDLED	IDLE
HSPTXD	5-13	268h	FE68h	0000h	*Res*							AB	TDATA							
HSPRXD	5-14	26Ah	FE6Ah	0000h	RDR	THRE	FER	OER	PER	MATCH	BRK	AB	RDATA							
HSPRXDP	5-16	26Ch	FE6Ch	0000h	RDR	THRE	FER	OER	PER	MATCH	BRK	AB	RDATA							
HSPBDV	5-18	26Eh	FE6Eh	0000h	BAUDDIV															
HSPM0	5-20	270h	FE70h	0000h	MCHR1								MCHR0							
HSPM1	5-21	272h	FE72h	0000h	MCHR3								MCHR2							
HSPM2	5-22	274h	FE74h	0000h	MCHR5								MCHR4							
HSPAB0	5-23	276h	FE76h	0000h	ABDIV0								ABTHRSH0							
HSPAB1	5-26	278h	FE78h	0000h	ABDIV1								ABTHRSH1							
HSPAB2	5-26	27Ah	FE7Ah	0000h	ABDIV2								ABTHRSH2							
HSPAB3	5-27	27Ch	FE7Ch	0000h	ABDIV3								ABTHRSH3							
Asynchronous Serial Port (UART) Registers																				
SPCON0	5-28	280h	FE80h	0000h	*Res*			RSIE	BRK	AB	FC	TXIE	RXIE	TMODE	RMODE	EVN	PEN	ABEN	D7	STP2
SPCON1	5-30	282h	FE82h	0000h	*Res*												BRKVAL	EXDWR	EXDRD	XTRN !
SPSTAT	5-31	284h	FE84h	0000h	*Res*						BRK	AB	RDR	THRE	FER	OER	PER	TEMT	IDLED	IDLE
SPIMSK	5-33	286h	FE86h	02F8h	*Res*						BRK	AB	RDR	THRE	FER	OER	PER	TEMT	IDLED	IDLE
SPTXD	5-35	288h	FE88h	0000h	*Res*							AB	TDATA							
SPRXD	5-36	28Ah	FE8Ah	0000h	RDR	THRE	FER	OER	PER	*Res*	BRK	AB	RDATA							
SPRXDP	5-38	28Ch	FE8Ch	0000h	RDR	THRE	FER	OER	PER	*Res*	BRK	AB	RDATA							
SPBDV	5-40	28Eh	FE8Eh	0000h	BAUDDIV															
NAME	SEE PAGE	OFFSET	DEFAULT LOCATION	DEFAULT VALUE	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

! *= See register or bit description for controller-specific details.*

Table A-1 Am186CC/CH/CU Microcontrollers Register Summary (Continued)

NAME	SEE PAGE	OFFSET	DEFAULT LOCATION	DEFAULT VALUE	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
General Circuit Interface (GCI) Registers (Am186CC Microcontroller Only)																				
GPCON	6-2	2A0h	FEA0h	0000h	*Res*							PCMFSC	MCARV	MARQ	MCHEN	MCHSEL	MEOMRQ	ICSEL	GCIACT	BRDIS
GISTAT	6-4	2A2h	FEA2h	0002h 0102h	*Res*						IC	DCLST	CHGCI1	CHGCI0	MRAD	MCD	MTARD	MEOMRD	MXBA	MRDA
GIMSK	6-6	2A4h	FEA4h	0000h	*Res*						IC	DCLST	CHGCI1	CHGCI0	MRAD	MCD	MTARD	MEOMRD	MXBA	MRDA
GTIC	6-8	2A6h	FEA6h	0007h	*Res*								TICEN	ECHOEN	*Res*			TICAD		
GICTD	6-9	2A8h	FEA8h	00FFh	*Res*								IC12T							
GICRD	6-10	2AAh	FEAAh	0000h	*Res*								IC12R							
GICRDP	6-11	2ACh	FEACh	0000h	*Res*								IC12P							
GCITD0	6-12	2AEh	FEAEh	000Fh	*Res*								BAR	*Res*			CI0T			
GCIRD0	6-13	2B0h	FEB0h	000Fh	*Res*												CI0R			
GCIRD0P	6-14	2B2h	FEB2h	000Fh	*Res*												CI0P			
GCITD1	6-15	2B4h	FEB4h	003Fh	*Res*										CI1T					
GCIRD1	6-16	2B6h	FEB6h	003Fh	*Res*										CI1R					
GCIRD1P	6-17	2B8h	FEB8h	003Fh	*Res*										CI1P					
GMTD	6-18	2BAh	FEBAh	00FFh	*Res*								MON01T							
GMRD	6-19	2BCh	FEBCh	0000h	*Res*								MON01R							
GMRDP	6-20	2BEh	FEBEh	0000h	*Res*								MON01P							
Time Slot Assigner (TSA) Channel A Registers (Am186CC and Am186CH Microcontrollers Only)																				
TSACON	7-2	2C0h	FEC0h	0000h	EN	*Res*					MODE !		*Res*		FSCP	DRVLVL	*Res*	ESADJ		
TSASTART	7-5	2C2h	FEC2h	0000h	*Res*				BPSTART											
TSASTOP	7-7	2C4h	FEC4h	0000h	*Res*				BPSTOP											
Time Slot Assigner (TSA) Channel B Registers (Am186CC and Am186CH Microcontrollers Only)																				
TSBCON	7-2	2C8h	FEC8h	0000h	EN	*Res*					MODE !		*Res*		FSCP	DRVLVL	*Res*	ESADJ		
TSBSTART	7-5	2CAh	FECAh	0000h	*Res*				BPSTART											
TSBSTOP	7-7	2CCh	FECCh	0000h	*Res*				BPSTOP											
Time Slot Assigner (TSA) Channel C Registers (Am186CC Microcontroller Only)																				
TSCCON	7-2	2D0h	FED0h	0000h	EN	*Res*					MODE !		*Res*		FSCP	DRVLVL	*Res*	ESADJ		
TSCSTART	7-5	2D2h	FED2h	0000h	*Res*				BPSTART											
TSCSTOP	7-7	2D4h	FED4h	0000h	*Res*				BPSTOP											
NAME	SEE PAGE	OFFSET	DEFAULT LOCATION	DEFAULT VALUE	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

! *= See register or bit description for controller-specific details.*

Table A-1 Am186CC/CH/CU Microcontrollers Register Summary (Continued)

NAME	SEE PAGE	OFFSET	DEFAULT LOCATION	DEFAULT VALUE	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Time Slot Assigner (TSA) Channel D Registers (Am186CC Microcontroller Only)																				
TSDCON	7-2	2D8h	FED8h	0000h	EN	*Res*				MODE !			*Res*		FSCP	DRVLVL	*Res*	ESADJ		
TSDSTART	7-5	2DAh	FEDAh	0000h	*Res*				BPSTART											
TSDSTOP	7-7	2DCh	FEDCh	0000h	*Res*				BPSTOP											
Synchronous Serial Interface (SSI) Registers																				
SSSTAT	8-2	2F0h	FEF0h	0000h	ENHCTL	*Res*												RE/TE	DR/DT	PB
SSCON	8-3	2F2h	FEF2h	0400h	*Res*				CLKP	DENP	*Res*	MSBF	*Res*	CLKEXP			*Res*		DE1	DE0
SSTXD1	8-5	2F4h	FEF4h	0000h	*Res*								TXDATA							
SSTXD0	8-5	2F6h	FEF6h	0000h	*Res*								TXDATA							
SSRXD	8-6	2F8h	FEF8h	0000h	*Res*								RXDATA							
Interrupt Controller Registers																				
CH0CON	9-4	300h	FF00h	003Fh	*Res*												MSK	PR		
CH1CON	9-6	302h	FF02h	000Fh	*Res*											LTM	MSK	PR		
CH2CON	9-8	304h	FF04h	000Fh	*Res*										SRC !	LTM	MSK	PR		
CH3CON	9-10	306h	FF06h	000Fh	*Res*										SRC	LTM	MSK	PR		
CH4CON !	9-12	308h	FF08h	003Fh	*Res*												MSK	PR		
CH5CON !	9-14	30Ah	FF0Ah	003Fh	*Res*												MSK	PR		
CH6CON !	9-16	30Ch	FF0Ch	003Fh	*Res*												MSK	PR		
CH7CON !	9-18	30Eh	FF0Eh	003Fh	*Res*												MSK	PR		
CH8CON	9-20	310h	FF10h	000Fh	*Res*										SRC !	LTM	MSK	PR		
CH9CON	9-22	312h	FF12h	000Fh	*Res*										SRC	LTM	MSK	PR		
CH10CON	9-24	314h	FF14h	000Fh	*Res*										SRC	LTM	MSK	PR		
CH11CON	9-26	316h	FF16h	000Fh	*Res*										SRC	LTM	MSK	PR		
CH12CON	9-28	318h	FF18h	000Fh	*Res*											LTM	MSK	PR		
CH13CON	9-30	31Ah	FF1Ah	000Fh	*Res*											LTM	MSK	PR		
CH14CON	9-32	31Ch	FF1Ch	001Fh	*Res*												MSK	PR		
EOI	9-33	320h	FF20h	0000h	NSPEC	*Res*										S				
POLL	9-35	322h	FF22h	0000h	IREQ	*Res*										S				
POLLST	9-36	324h	FF24h	0000h	IREQ	*Res*										S				
IMASK	9-37	326h	FF26h	FFFFh	*Res*	CH14	CH13	CH12	CH11	CH10	CH9	CH8	CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0
PRIMSK	9-39	328h	FF28h	0007h	*Res*													PRM		
INSERV	9-40	32Ah	FF2Ah	0000h	*Res*	CH14	CH13	CH12	CH11	CH10	CH9	CH8	CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0
NAME	SEE PAGE	OFFSET	DEFAULT LOCATION	DEFAULT VALUE	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

! = See register or bit description for controller-specific details.

Table A-1 Am186CC/CH/CU Microcontrollers Register Summary (Continued)

NAME	SEE PAGE	OFFSET	DEFAULT LOCATION	DEFAULT VALUE	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REQST	9-41	32Ch	FF2Ch	0000h	*Res*	CH14	CH13	CH12	CH11	CH10	CH9	CH8	CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0
INTSTS	9-42	32Eh	FF2Eh	0000h	*Res*									DMA3	DMA2	DMA1	DMA0	TIM2	TIM1	TIM0
DMAHLT	9-43	330h	FF30h	0000h	DHLT	*Res*														
SHREQ	9-44	332h	FF32h	0000h	PIO35	PIO34	PIO33	PIO30	PIO29	PIO27	PIO15	PIO5	INT7	INT6	INT5	INT4	INT3	INT2	INT1	*Res*
SHMASK	9-45	334h	FF34h	FFFFh	PIO35	PIO34	PIO33	PIO30	PIO29	PIO27	PIO15	PIO5	INT7	INT6	INT5	INT4	INT3	INT2	INT1	*Res*
INTPOL	9-46	336h	FF36h	FFFFh	*Res*							INT8	INT7	INT6	INT5	INT4	INT3	INT2	INT1	INT0
PIOPOL	9-47	338h	FF38h	FFFFh	PIO35	PIO34	PIO33	PIO30	PIO29	PIO27	PIO15	PIO5	*Res*							
Timer Registers																				
T0CON	10-2	340h	FF40h	0000h	EN	$\overline{\text{INH}}$	INT	RIU	*Res*						MC	RTG	P	EXT	ALT	CONT
T0CNT	10-5	342h	FF42h	0000h	TC															
T0CMPA	10-6	344h	FF44h	0000h	TC															
T0CMPB	10-7	346h	FF46h	0000h	TC															
T1CON	10-8	348h	FF48h	0000h	EN	$\overline{\text{INH}}$	INT	RIU	*Res*						MC	RTG	P	EXT	ALT	CONT
T1CNT	10-11	34Ah	FF4Ah	0000h	TC															
T1CMPA	10-12	34Ch	FF4Ch	0000h	TC															
T1CMPB	10-13	34Eh	FF4Eh	0000h	TC															
T2CON	10-14	350h	FF50h	0000h	EN	$\overline{\text{INH}}$	INT	*Res*							MC	*Res*				CONT
T2CNT	10-16	352h	FF52h	0000h	TC															
T2CMPA	10-17	354h	FF54h	0000h	TC															
Chip Select Registers																				
UMCS	11-2	3A0h	FFA0h	F01Bh F03Bh	*Res*	LB			*Res*				DA	UDEN	USIZ	*Res*		R2	R1	R0
LMCS	11-5	3A2h	FFA2h	0F1Bh	*Res*	UB			*Res*				DA	LDEN	LSIZ	*Res*		R2	R1	R0
PACS	11-9	3A4h	FFA4h	0073h	BA[19–11]									*Res*			R3	R2	R1	R0
MMCS	11-11	3A6h	FFA6h	7FDBh	BA[19–13]							*Res*			MCS0_ ONLY	*Res*		R2	R1	R0
MPCS	11-13	3A8h	FFA8h	8183h	*Res*	M[6–0]							*Res*	MS	OMSIZ	IOSIZ	R3	R2	R1	R0
DRAM Controller Registers																				
CDRAM	12-2	3AAh	FFAAh	0000h	*Res*			RC												
EDRAM	12-3	3ACh	FFACh	0000h	EN	*Res*		T												
NAME	SEE PAGE	OFFSET	DEFAULT LOCATION	DEFAULT VALUE	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

! *= See register or bit description for controller-specific details.*

Table A-1 Am186CC/CH/CU Microcontrollers Register Summary (Continued)

NAME	SEE PAGE	OFFSET	DEFAULT LOCATION	DEFAULT VALUE	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Programmable I/O (PIO) Registers																				
PIOMODE0	13-4	3C0h	FFC0h	0000h	PMODE 15	PMODE 14	PMODE 13	PMODE 12	PMODE 11	PMODE 10	PMODE 9	PMODE 8	PMODE 7	PMODE 6	PMODE 5	PMODE 4	PMODE 3	PMODE 2	PMODE 1	PMODE 0
PIODIR0	13-5	3C2h	FFC2h	1EFFh	PDIR15	PDIR14	PDIR13	PDIR12	PDIR11	PDIR10	PDIR9	PDIR8	PDIR7	PDIR6	PDIR5	PDIR4	PDIR3	PDIR2	PDIR1	PDIR0
PIODATA0	13-6	3C4h	FFC4h	????h	PDATA15	PDATA14	PDATA13	PDATA12	PDATA11	PDATA10	PDATA9	PDATA8	PDATA7	PDATA6	PDATA5	PDATA4	PDATA3	PDATA2	PDATA1	PDATA0
PIOSET0	13-7	3C6h	FFC6h	0000h	PSET15	PSET14	PSET13	PSET12	PSET11	PSET10	PSET9	PSET8	PSET7	PSET6	PSET5	PSET4	PSET3	PSET2	PSET1	PSET0
PIOCLR0	13-8	3C8h	FFC8h	0000h	PCLR15	PCLR14	PCLR13	PCLR12	PCLR11	PCLR10	PCLR9	PCLR8	PCLR7	PCLR6	PCLR5	PCLR4	PCLR3	PCLR2	PCLR1	PCLR0
PIOMODE1	13-9	3CAh	FFCAh	0000h	PMODE 31	PMODE 30	PMODE 29	PMODE 28	PMODE 27	PMODE 26	PMODE 25	PMODE 24	PMODE 23	PMODE 22	PMODE 21	PMODE 20	PMODE 19	PMODE 18	PMODE 17	PMODE 16
PIODIR1	13-10	3CCh	FFCCh	9FFFh	PDIR31	PDIR30	PDIR29	PDIR28	PDIR27	PDIR26	PDIR25	PDIR24	PDIR23	PDIR22	PDIR21	PDIR20	PDIR19	PDIR18	PDIR17	PDIR16
PIODATA1	13-11	3CEh	FFCEh	????h	PDATA31	PDATA30	PDATA29	PDATA28	PDATA27	PDATA26	PDATA25	PDATA24	PDATA23	PDATA22	PDATA21	PDATA20	PDATA19	PDATA18	PDATA17	PDATA16
PIOSET1	13-12	3D0h	FFD0h	0000h	PSET31	PSET30	PSET29	PSET28	PSET27	PSET26	PSET25	PSET24	PSET23	PSET22	PSET21	PSET20	PSET19	PSET18	PSET17	PSET16
PIOCLR1	13-13	3D2h	FFD2h	0000h	PCLR31	PCLR30	PCLR29	PCLR28	PCLR27	PCLR26	PCLR25	PCLR24	PCLR23	PCLR22	PCLR21	PCLR20	PCLR19	PCLR18	PCLR17	PCLR16
PIOMODE2	13-14	3D4h	FFD4h	0000h	PMODE 47	PMODE 46	PMODE 45	PMODE 44	PMODE 43	PMODE 42	PMODE 41	PMODE 40	PMODE 39	PMODE 38	PMODE 37	PMODE 36	PMODE 35	PMODE 34	PMODE 33	PMODE 32
PIODIR2	13-15	3D6h	FFD6h	FFF1h	PDIR47	PDIR46	PDIR45	PDIR44	PDIR43	PDIR42	PDIR41	PDIR40	PDIR39	PDIR38	PDIR37	PDIR36	PDIR35	PDIR34	PDIR33	PDIR32
PIODATA2	13-16	3D8h	FFD8h	????h	PDATA47	PDATA46	PDATA45	PDATA44	PDATA43	PDATA42	PDATA41	PDATA40	PDATA39	PDATA38	PDATA37	PDATA36	PDATA35	PDATA34	PDATA33	PDATA32
PIOSET2	13-17	3DAh	FFDAh	0000h	PSET47	PSET46	PSET45	PSET44	PSET43	PSET42	PSET41	PSET40	PSET39	PSET38	PSET37	PSET36	PSET35	PSET34	PSET33	PSET32
PIOCLR2	13-18	3DCh	FFDCh	0000h	PCLR47	PCLR46	PCLR45	PCLR44	PCLR43	PCLR42	PCLR41	PCLR40	PCLR39	PCLR38	PCLR37	PCLR36	PCLR35	PCLR34	PCLR33	PCLR32
Reset Configuration Register																				
RESCON	14-2	3DEh	FFDEh	????h	RCD15	RCD14	RCD13	RCD12	RCD11	RCD10	RCD9	RCD8	RCD7	RCD6	RCD5	RCD4	RCD3	RCD2	RCD1	RCD0
Watchdog Timer Register																				
WDTCON	15-2	3E0h	FFE0h	C180h	ENA	WRST	RSTFLAG	NMIFLAG	*Res*			EXRST	ES							
Miscellaneous Registers																				
SYSCON	16-2	3F0h	FFF0h	0000h	*Res*		DSDEN	PWD	DISMEM	DISIO	ITF4 !		EXSYNC !	*Res*			DISCLK	*Res*		
PRL	16-4	3F4h	FFF4h	4001h	PRL															
RELOC	16-5	3FEh	FFFEh	20FCh	DUAL	*Res*		M/$\overline{\text{IO}}$	R[19–10]										*Res*	
NAME	SEE PAGE	OFFSET	DEFAULT LOCATION	DEFAULT VALUE	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

! *= See register or bit description for controller-specific details.*

INDEX