Élan™SC520 Microcontroller

Register Set Manual

Order #22005B

AMD

Trademarks

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™ 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 Support link for the latest AMD technical support phone numbers, software, and Frequently Asked Questions.

For technical support questions on all E86 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 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 **web.feedback@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, go to **www.amd.com/support/literature.html** or, in the U.S. and Canada, call (800) 222-9323.

Third-Party Support

AMD FusionE86℠ 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

ÉLAN™SC520 MICROCONTROLLER

The Élan™SC520 microcontroller is a full-featured microcontroller developed for the general embedded market. The ÉlanSC520 microcontroller combines a 32-bit, low-voltage $Am5_x86$® CPU with a complete set of integrated peripherals suitable for both real-time and PC/AT-compatible embedded applications.

PURPOSE OF THIS MANUAL

This manual includes in reference format the complete set of registers required to configure the ÉlanSC520 microcontroller and control its peripherals. This manual does not document the $Am5_x86$ processor registers.

Intended Audience

This reference manual is intended primarily for programmers who are developing code for the ÉlanSC520 microcontroller. Computer software and hardware architects and system engineers who are designing or are considering designing systems based on this microcontroller may also be interested in the information contained in this document. For more information on programming this microcontroller, see the *Élan™SC520 Microcontroller User's Manual*, order #22004.

Overview of this Manual

The manual is organized into the following chapters:

- Chapter 1 contains an overview of all the microcontroller's **configuration registers**.
- Chapter 2 describes the **system address mapping registers**.
- Chapter 3 describes the **reset generation registers**.
- Chapter 4 describes the **$Am5_x86$ CPU registers**.
- Chapter 5 describes the **system arbitration registers**.
- Chapter 6 describes the **PCI host bridge registers**.
- Chapter 7 describes the **synchronous DRAM (SDRAM) controller registers**.
- Chapter 8 describes the **write buffer and read buffer register**.
- Chapter 9 describes the **ROM/Flash memory controller registers**.
- Chapter 10 describes the **general-purpose (GP) bus controller registers**.
- Chapter 11 describes the **GP bus DMA controller registers**.
- Chapter 12 describes the **programmable interrupt controller (PIC) registers**.
- Chapter 13 describes the **programmable interval timer (PIT) registers**.
- Chapter 14 describes the **general-purpose (GP) timer registers**.
- Chapter 15 describes the **software timer registers**.
- Chapter 16 describes the **watchdog timer (WDT) registers**.
- Chapter 17 describes the **real-time clock (RTC) registers**.

- Chapter 18 describes the **UART registers**.
- Chapter 19 describes the **synchronous serial interface (SSI) registers**.
- Chapter 20 describes the **programmable I/O (PIO) registers**.
- The Index lists all registers and bits alphabetically by name and mnemonic.

Each chapter describes the function's memory-mapped configuration region (MMCR) registers first, followed by direct-mapped and then indexed register descriptions, if any. Within each chapter, the registers of each type are listed in ascending hexadecimal order unless descriptions for identical registers (for example, direct-mapped UART registers) are combined.

RELATED DOCUMENTS

The following documents contain additional information that will be useful in designing an embedded application based on the ÉlanSC520 microcontroller.

AMD Documentation

In addition to this manual, the documentation set for the ÉlanSC520 microcontroller includes the following documents:

- *Élan™SC520 Microcontroller User's Manual*, order #22004, provides a functional description of the microcontroller for both hardware and software designers.
- *Élan™SC520 Microcontroller Data Sheet*, order #22003, includes complete pin lists, pin state tables, timing and thermal characteristics, and package dimensions for the ÉlanSC520 microcontroller.

Other information of interest:

- The *Am486® Microprocessor Software User's Manual*, order #18497, includes the complete instruction set for the integrated $Am5_x86$ CPU.
- *Am5x86® Microprocessor Family Data Sheet*, order #19751
- *Am486®DX/DX2 Microprocessor Hardware Reference Manual*, order #17965
- *E86™ Family Products and Development Tools CD*, order #21058, provides a single-source multimedia tool for customer evaluation of AMD products, as well as FusionE86 partner tools and technologies that support the E86 family. Technical documentation is included on the CD in PDF format.

To order literature, contact the nearest AMD sales office or call the literature center at one of the numbers listed on the back cover of this manual. In addition, all these documents are available in PDF form on the AMD web site. To access the AMD home page, go to **www.amd.com**. Then follow the Embedded Processor link for information about AMD's E86 family of microcontrollers.

Additional Information

The following non-AMD documents and sources provide additional information that may be of interest to ÉlanSC520 microcontroller users:

- *PCI Local Bus Specification,* Revision 2.2, December 18, 1998, PCI Special Interest Group, 800-433-5177 (US), 503-693-6360 (International), www.pcisig.com.
- *IEEE Std 1149.1-1990 Standard Test Access Port and Boundary-Scan Architecture,* (order #SH16626-NYF), Institute of Electrical and Electronic Engineers, Inc., 800-678-4333, www.ieee.org.
- *PCI System Architecture*, Mindshare, Inc., Reading, MA: Addison-Wesley, 1995, ISBN 0-201-40993-3.
- *ISA System Architecture*, Mindshare, Inc., Reading, MA: Addison-Wesley, 1995, ISBN 0-201-40996-8.
- *80486 System Architecture*, Mindshare, Inc., Reading, MA: Addison-Wesley, 1995, ISBN 0-201-40994-1
- *The Indispensable PC Hardware Book*, Hans-Peter Messmer, Wokingham, England: Addison-Wesley, 1995, ISBN 0-201-87697-3.

DOCUMENTATION CONVENTIONS

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

Table 0-1 Documentation Notation

Notation	Meaning
Reset Default Values	
Default	Value after a system reset
0	Low
1	Active or High
x	No value is guaranteed
?	Determined by sources external to the ÉlanSC520 microcontroller
Read/Write Attributes	
R	The bit field is read-only. A write to the register at this bit field has no effect. The contents may or may not be changed by hardware.
W	The bit field is write-only. Reading this register at this bit field does not return a meaningful value and has no side effects.
R/W	The bit field is read/write. Reading the register at this bit field always returns the last value written. Reads have no side effects.
R/W!	The bit field is read/write with conditions. The “!” indicates that there are side effects to using this bit. For example, reading a bit or register might not always return the last value written. Note that both reads and writes can have side effects. If you see a “!”, be sure to read the bit description and programming notes.
RSV	The bit field is reserved for internal test/debug or future expansion. This bit field should be written to 0 for normal system operation. This bit field always returns 0 when read.
RSV!	The bit field is reserved for compatibility purposes. For example, the bit field might be ignored during writes to maintain software compatibility. If you see a “!”, be sure to read the bit description and programming notes.
Reference Notation	
MMCR offset 00h	ÉlanSC520 microcontroller Memory-Mapped Configuration Region (MMCR) offset register 00h
PCI index 00h	PCI indexed register 00h
Port 00h	Direct-mapped I/O register 00h
RTC index 00h	RTC and configuration RAM indexed register 00h

Table 0-1 Documentation Notation (Continued)

Notation	Meaning
Pin Naming	
{ }	Pin function during hardware reset
[ ]	Alternative pin function selected by software configuration
$\overline{\text{ROMCS1}}$	An overbar indicates that the signal assumes the logic Low state when asserted.
GPRESET	The absence of an overbar indicates that the signal assumes the logic High state when asserted.
$\overline{\text{ads}}$, hold	A signal name in all lowercase indicates an internal signal.
$\overline{\text{ROMCS2}}$–$\overline{\text{ROMCS1}}$	Two ROM chip select signals
$\overline{\text{ROMCSx}}$	Any of the two ROM chip select signals
Numbers	
b	Binary number
d	Decimal number Decimal is the default radix
h	Hexadecimal number
x in register address	Any of several legal values; e.g., using 0xF8h for the UART Transmit Holding register is either 02F8h or 03F8h, depending on the UART
[X–Y]	The bit field that consists of bits X through Y. Example: The SB_ADDR[23–16] bit field.
33 MHz	Refers to the system clock frequency being used. This can be either 33.000 MHz or 33.333 MHz. See the *Élan™SC520 Microcontroller User's Manual*, order #22004, for more information about clock generation.
General	
field	Bit field in a register (one or more consecutive and related bits)
can	It is possible to perform an action if properly configured
will	A certain action is going to occur
Set the ENB bit.	Write the ENB bit to 1. ***Note:*** *The bit referred to is either in the register being described, or the register is referred to explicitly in the surrounding text.*
Clear the ENB bit.	Change the ENB bit to 0. Usually a bit is cleared by writing a 0 to it; however, some bits are cleared by writing a 1.
Reset the ENB bit.	Context-sensitive. Can refer either to resetting the bit to its default value or to clearing the bit.

CHAPTER

1 CONFIGURATION REGISTER OVERVIEW

The Élan™SC520 microcontroller has four different types of configuration registers:

- **Memory-Mapped Configuration Region (MMCR) Registers**—These are memory-mapped peripherals and configuration registers that are specific to the ÉlanSC520 microcontroller's control and status functions, such as the SDRAM and GP bus controllers. These registers are 8-bits, 16-bits, or 32-bits wide and reside in memory space.
- **Direct-Mapped Registers**—These include the Configuration Base Address (CBAR) register, the PCI Configuration Address and Data (PCICFGADR and PCICFGDATA) registers, and PC/AT-compatible peripherals. All direct-mapped I/O registers reside in fixed I/O space. The CBAR, PCICFGADR, and PCICFGDATA registers are 32 bits wide. All other direct-mapped peripheral configuration registers are 8 bits wide.
- **PCI Host Bridge Indexed Configuration Registers**—These registers are located in the PCI bus configuration space, which is defined in the *PCI Local Bus Specification,* Revision 2.2, to be accessed through two 32-bit I/O locations at 0CF8h (index) and 0CFCh (data).
- **RTC Indexed Registers**—These registers are located in the PC/AT-compatible real-time clock (RTC) configuration space, which is accessed using I/O ports 0070h (index) and 0071h (data).

Register descriptions are organized within this manual by function, e.g., GP bus, SDRAM, or UART. Each function's chapter describes the MMCR registers first, followed by direct-mapped, and then indexed register descriptions, if any. In each chapter, registers of each type are listed in ascending hexadecimal order unless descriptions for identical registers (for example, direct-mapped UART registers) are combined.

The remainder of this chapter presents an overview of the registers by type.

1.1 MEMORY-MAPPED CONFIGURATION REGION (MMCR) REGISTERS

The ÉlanSC520 microcontroller's memory-mapped configuration region (MMCR) contains all internal peripheral control and configuration registers that are not defined as direct-mapped I/O, PCI indexed, or RTC indexed registers.

After reset, the MMCR registers are located in the 4-Kbyte region in memory address space from FFFEF000–FFFEFFFFh. The MMCR registers can be aliased to any 4-Kbyte region in the lower 1-Gbyte address space (00000000h–1FFFFFFh) via the I/O-mapped CBAR register (see page 2-9). The MMCR is available at its original location in high memory even if it is aliased via the CBAR register. See the memory and I/O space chapter in the *Élan™SC520 Microcontroller User's Manual*, order #22004, for more detail.

Table 1-1 on page 1-2 lists all the MMCR registers included in the ÉlanSC520 microcontroller.

Table 1-1 Memory-Mapped Configuration Region (MMCR) Registers (By Offset)

Register Name	Mnemonic	MMCR Offset	Page Number
CPU		**00–02h**	
ÉlanSC520 Microcontroller Revision ID	REVID	00h	page 4-2
Am5$_x$86® CPU Control	CPUCTL	02h	page 4-3
SDRAM Controller		**10–28h**	
SDRAM Control	DRCCTL	10h	page 7-4
SDRAM Timing Control	DRCTMCTL	12h	page 7-4
SDRAM Bank Configuration	DRCCFG	14h	page 7-5
SDRAM Bank 0–3 Ending Address	DRCBENDADR	18h	page 7-7
ECC Control	ECCCTL	20h	page 7-9
ECC Status	ECCSTA	21h	page 7-10
ECC Check Bit Position	ECCCKBPOS	22h	page 7-11
ECC Check Code Test	ECCCKTEST	23h	page 7-12
ECC Single-Bit Error Address	ECCSBADD	24h	page 7-14
ECC Multi-Bit Error Address	ECCMBADD	28h	page 7-15
SDRAM Buffer		**40h**	
SDRAM Buffer Control	DBCTL	40h	page 8-2
ROM/Flash Controller		**50–56h**	
$\overline{\text{BOOTCS}}$ Control	BOOTCSCTL	50h	page 9-2
$\overline{\text{ROMCS1}}$ Control	ROMCS1CTL	54h	page 9-4
$\overline{\text{ROMCS2}}$ Control	ROMCS2CTL	56h	page 9-6
PCI Bus Host Bridge		**60–6Ch**	
Host Bridge Control	HBCTL	60h	page 6-3
Host Bridge Target Interrupt Control	HBTGTIRQCTL	62h	page 6-5
Host Bridge Target Interrupt Status	HBTGTIRQSTA	64h	page 6-7
Host Bridge Master Interrupt Control	HBMSTIRQCTL	66h	page 6-9
Host Bridge Master Interrupt Status	HBMSTIRQSTA	68h	page 6-12
Host Bridge Master Interrupt Address	MSTINTADD	6Ch	page 6-14
System Arbitration		**70–74h**	
System Arbiter Control	SYSARBCTL	70h	page 5-2
PCI Bus Arbiter Status	PCIARBSTA	71h	page 5-3
System Arbiter Master Enable	SYSARBMENB	72h	page 5-4
Arbiter Priority Control	ARBPRICTL	74h	page 5-6

Table 1-1 Memory-Mapped Configuration Region (MMCR) Registers (By Offset) (Continued)

Register Name	Mnemonic	MMCR Offset	Page Number
System Address Mapping		**80–C4h**	
Address Decode Control	ADDDECCTL	80h	page 2-2
Write-Protect Violation Status	WPVSTA	82h	page 2-4
Programmable Address Region 0	PAR0	88h	page 2-5
Programmable Address Region 1	PAR1	8Ch	page 2-5
Programmable Address Region 2	PAR2	90h	page 2-5
Programmable Address Region 3	PAR3	94h	page 2-5
Programmable Address Region 4	PAR4	98h	page 2-5
Programmable Address Region 5	PAR5	9Ch	page 2-5
Programmable Address Region 6	PAR6	A0h	page 2-5
Programmable Address Region 7	PAR7	A4h	page 2-5
Programmable Address Region 8	PAR8	A8h	page 2-5
Programmable Address Region 9	PAR9	ACh	page 2-5
Programmable Address Region 10	PAR10	B0h	page 2-5
Programmable Address Region 11	PAR11	B4h	page 2-5
Programmable Address Region 12	PAR12	B8h	page 2-5
Programmable Address Region 13	PAR13	BCh	page 2-5
Programmable Address Region 14	PAR14	C0h	page 2-5
Programmable Address Region 15	PAR15	C4h	page 2-5
GP Bus Controller		**C00–C10h**	
GP Echo Mode	GPECHO	C00h	page 10-2
GP Chip Select Data Width	GPCSDW	C01h	page 10-3
GP Chip Select Qualification	GPCSQUAL	C02h	page 10-5
GP Chip Select Recovery Time	GPCSRT	C08h	page 10-7
GP Chip Select Pulse Width	GPCSPW	C09h	page 10-8
GP Chip Select Offset	GPCSOFF	C0Ah	page 10-9
GP Read Pulse Width	GPRDW	C0Bh	page 10-10
GP Read Offset	GPRDOFF	C0Ch	page 10-11
GP Write Pulse Width	GPWRW	C0Dh	page 10-12
GP Write Offset	GPWROFF	C0Eh	page 10-13
GPALE Pulse Width	GPALEW	C0Fh	page 10-14
GPALE Offset	GPALEOFF	C10h	page 10-15

Table 1-1 Memory-Mapped Configuration Region (MMCR) Registers (By Offset) (Continued)

Register Name	Mnemonic	MMCR Offset	Page Number
Programmable Input/Output		**C20–C3Ah**	
PIO15–PIO0 Pin Function Select	PIOPFS15_0	C20h	page 20-3
PIO31–PIO16 Pin Function Select	PIOPFS31_16	C22h	page 20-5
Chip Select Pin Function Select	CSPFS	C24h	page 20-7
Clock Select	CLKSEL	C26h	page 20-9
Drive Strength Control	DSCTL	C28h	page 20-10
PIO15–PIO0 Direction	PIODIR15_0	C2Ah	page 20-12
PIO31–PIO16 Direction	PIODIR31_16	C2Ch	page 20-14
PIO15–PIO0 Data	PIODATA15_0	C30h	page 20-16
PIO31–PIO16 Data	PIODATA31_16	C32h	page 20-18
PIO15–PIO0 Set	PIOSET15_0	C34h	page 20-20
PIO31–PIO16 Set	PIOSET31_16	C36h	page 20-22
PIO15–PIO0 Clear	PIOCLR15_0	C38h	page 20-24
PIO31–PIO16 Clear	PIOCLR31_16	C3Ah	page 20-26
Software Timer		**C60–C62h**	
Software Timer Millisecond Count	SWTMRMILLI	C60h	page 15-2
Software Timer Microsecond Count	SWTMRMICRO	C62h	page 15-3
Software Timer Configuration	SWTMRCFG	C64h	page 15-4
General-Purpose Timers		**C70–C8Eh**	
GP Timers Status	GPTMRSTA	C70h	page 14-2
GP Timer 0 Mode/Control	GPTMR0CTL	C72h	page 14-3
GP Timer 0 Count	GPTMR0CNT	C74h	page 14-6
GP Timer 0 Maxcount Compare A	GPTMR0MAXCMPA	C76h	page 14-7
GP Timer 0 Maxcount Compare B	GPTMR0MAXCMPB	C78h	page 14-8
GP Timer 1 Mode/Control	GPTMR1CTL	C7Ah	page 14-9
GP Timer 1 Count	GPTMR1CNT	C7Ch	page 14-12
GP Timer 1 Maxcount Compare A	GPTMR1MAXCMPA	C7Eh	page 14-13
GP Timer 1 Maxcount Compare B	GPTMR1MAXCMPB	C80h	page 14-14
GP Timer 2 Mode/Control	GPTMR2CTL	C82h	page 14-15
GP Timer 2 Count	GPTMR2CNT	C84h	page 14-17
GP Timer 2 Maxcount Compare A	GPTMR2MAXCMPA	C8Eh	page 14-18
Watchdog Timer		**CB0–CB6h**	
Watchdog Timer Control	WDTMRCTL	CB0h	page 16-2
Watchdog Timer Count Low	WDTMRCNTL	CB2h	page 16-4
Watchdog Timer Count High	WDTMRCNTH	CB4h	page 16-5
Reserved	—	CB6h	—

Table 1-1 Memory-Mapped Configuration Region (MMCR) Registers (By Offset) (Continued)

Register Name	Mnemonic	MMCR Offset	Page Number
UART Serial Ports		**CC0–CC6h**	
UART 1 General Control	UART1CTL	CC0h	page 18-3
UART 1 General Status	UART1STA	CC1h	page 18-4
UART 1 FIFO Control Shadow	UART1FCRSHAD	CC2h	page 18-5
UART 2 General Control	UART2CTL	CC4h	page 18-3
UART 2 General Status	UART2STA	CC5h	page 18-4
UART 2 FIFO Control Shadow	UART2FCRSHAD	CC6h	page 18-5
Synchronous Serial Interface		**CD0–CD4h**	
SSI Control	SSICTL	CD0h	page 19-2
SSI Transmit	SSIXMIT	CD1h	page 19-4
SSI Command	SSICMD	CD2h	page 19-5
SSI Status	SSISTA	CD3h	page 19-6
SSI Receive	SSIRCV	CD4h	page 19-7
Programmable Interrupt Controller		**D00–D5Ah**	
Interrupt Control	PICICR	D00h	page 12-4
Master PIC Interrupt Mode	MPICMODE	D02h	page 12-6
Slave 1 PIC Interrupt Mode	SL1PICMODE	D03h	page 12-8
Slave 2 PIC Interrupt Mode	SL2PICMODE	D04h	page 12-9
Software Interrupt 16–1 Control	SWINT16_1	D08h	page 12-10
Software Interrupt 22–17/NMI Control	SWINT22_17	D0Ah	page 12-13
Interrupt Pin Polarity	INTPINPOL	D10h	page 12-15
PCI Host Bridge Interrupt Mapping	PCIHOSTMAP	D14h	page 12-17
ECC Interrupt Mapping	ECCMAP	D18h	page 12-19
GP Timer 0 Interrupt Mapping	GPTMR0MAP	D1Ah	page 12-21
GP Timer 1 Interrupt Mapping	GPTMR1MAP	D1Bh	page 12-21
GP Timer 2 Interrupt Mapping	GPTMR2MAP	D1Ch	page 12-21
PIT 0 Interrupt Mapping	PIT0MAP	D20h	page 12-21
PIT 1 Interrupt Mapping	PIT1MAP	D21h	page 12-21
PIT 2 Interrupt Mapping	PIT2MAP	D22h	page 12-21
UART 1 Interrupt Mapping	UART1MAP	D28h	page 12-21
UART 2 Interrupt Mapping	UART2MAP	D29h	page 12-21
PCI Interrupt A Mapping	PCIINTAMAP	D30h	page 12-21
PCI Interrupt B Mapping	PCIINTBMAP	D31h	page 12-21
PCI Interrupt C Mapping	PCIINTCMAP	D32h	page 12-21
PCI Interrupt D Mapping	PCIINTDMAP	D33h	page 12-21
DMA Buffer Chaining Interrupt Mapping	DMABCINTMAP	D40h	page 12-21
SSI Interrupt Mapping	SSIMAP	D41h	page 12-21
Watchdog Timer Interrupt Mapping	WDTMAP	D42h	page 12-21
RTC Interrupt Mapping	RTCMAP	D43h	page 12-21
Write-Protect Violation Interrupt Mapping	WPVMAP	D44h	page 12-21
AMDebug™ Technology RX/TX Interrupt Mapping	ICEMAP	D45h	page 12-21

Table 1-1 Memory-Mapped Configuration Region (MMCR) Registers (By Offset) (Continued)

Register Name	Mnemonic	MMCR Offset	Page Number
Floating Point Error Interrupt Mapping	FERRMAP	D46h	page 12-21
GPIRQ0 Interrupt Mapping	GP0IMAP	D50h	page 12-21
GPIRQ1 Interrupt Mapping	GP1IMAP	D51h	page 12-21
GPIRQ2 Interrupt Mapping	GP2IMAP	D52h	page 12-21
GPIRQ3 Interrupt Mapping	GP3IMAP	D53h	page 12-21
GPIRQ4 Interrupt Mapping	GP4IMAP	D54h	page 12-21
GPIRQ5 Interrupt Mapping	GP5IMAP	D55h	page 12-21
GPIRQ6 Interrupt Mapping	GP6IMAP	D56h	page 12-21
GPIRQ7 Interrupt Mapping	GP7IMAP	D57h	page 12-21
GPIRQ8 Interrupt Mapping	GP8IMAP	D58h	page 12-21
GPIRQ9 Interrupt Mapping	GP9IMAP	D59h	page 12-21
GPIRQ10 Interrupt Mapping	GP10IMAP	D5Ah	page 12-21
Reset Generation		**D70–D74h**	
System Board Information	SYSINFO	D70h	page 3-2
Reset Configuration	RESCFG	D72h	page 3-3
Reset Status	RESSTA	D74h	page 3-5
GP DMA Controller		**D80–DBEh**	
GP-DMA Control	GPDMACTL	D80h	page 11-4
GP-DMA Memory-Mapped I/O	GPDMAMMIO	D81h	page 11-5
GP-DMA Resource Channel Map A	GPDMAEXTCHMAPA	D82h	page 11-6
GP-DMA Resource Channel Map B	GPDMAEXTCHMAPB	D84h	page 11-8
GP-DMA Channel 0 Extended Page	GPDMAEXTPG0	D86h	page 11-10
GP-DMA Channel 1 Extended Page	GPDMAEXTPG1	D87h	page 11-11
GP-DMA Channel 2 Extended Page	GPDMAEXTPG2	D88h	page 11-12
GP-DMA Channel 3 Extended Page	GPDMAEXTPG3	D89h	page 11-13
GP-DMA Channel 5 Extended Page	GPDMAEXTPG5	D8Ah	page 11-14
GP-DMA Channel 6 Extended Page	GPDMAEXTPG6	D8Bh	page 11-15
GP-DMA Channel 7 Extended Page	GPDMAEXTPG7	D8Ch	page 11-16
GP-DMA Channel 3 Extended Transfer Count	GPDMAEXTTC3	D90h	page 11-17
GP-DMA Channel 5 Extended Transfer Count	GPDMAEXTTC5	D91h	page 11-18
GP-DMA Channel 6 Extended Transfer Count	GPDMAEXTTC6	D92h	page 11-19
GP-DMA Channel 7 Extended Transfer Count	GPDMAEXTTC7	D93h	page 11-20
Buffer Chaining Control	GPDMABCCTL	D98h	page 11-21
Buffer Chaining Status	GPDMABCSTA	D99h	page 11-22
Buffer Chaining Interrupt Enable	GPDMABSINTENB	D9Ah	page 11-24
Buffer Chaining Valid	GPDMABCVAL	D9Bh	page 11-25
GP-DMA Channel 3 Next Address Low	GPDMANXTADDL3	DA0h	page 11-26
GP-DMA Channel 3 Next Address High	GPDMANXTADDH3	DA2h	page 11-27
GP-DMA Channel 5 Next Address Low	GPDMANXTADDL5	DA4h	page 11-28
GP-DMA Channel 5 Next Address High	GPDMANXTADDH5	DA6h	page 11-29
GP-DMA Channel 6 Next Address Low	GPDMANXTADDL6	DA8h	page 11-30

Table 1-1 Memory-Mapped Configuration Region (MMCR) Registers (By Offset) (Continued)

Register Name	Mnemonic	MMCR Offset	Page Number
GP-DMA Channel 6 Next Address High	GPDMANXTADDH6	DAAh	page 11-31
GP-DMA Channel 7 Next Address Low	GPDMANXTADDL7	DACh	page 11-32
GP-DMA Channel 7 Next Address High	GPDMANXTADDH7	DAEh	page 11-33
GP-DMA Channel 3 Next Transfer Count Low	GPDMANXTTCL3	DB0h	page 11-34
GP-DMA Channel 3 Next Transfer Count High	GPDMANXTTCH3	DB2h	page 11-35
GP-DMA Channel 5 Next Transfer Count Low	GPDMANXTTCL5	DB4h	page 11-36
GP-DMA Channel 5 Next Transfer Count High	GPDMANXTTCH5	DB6h	page 11-37
GP-DMA Channel 6 Next Transfer Count Low	GPDMANXTTCL6	DB8h	page 11-38
GP-DMA Channel 6 Next Transfer Count High	GPDMANXTTCH6	DBAh	page 11-39
GP-DMA Channel 7 Next Transfer Count Low	GPDMANXTTCL7	DBCh	page 11-40
GP-DMA Channel 7 Next Transfer Count High	GPDMANXTTCH7	DBEh	page 11-41

1.2 DIRECT-MAPPED I/O REGISTERS

The direct-mapped I/O registers include the Configuration Base Address (CBAR) register and PC/AT-compatible peripheral registers, such as the programmable interval timer (PIT), programmable interrupt controller (PIC), direct memory access (DMA) controller, the real-time clock (RTC) index and data registers, the PCI configuration address and data registers, two universal asynchronous receive/transmit (UART) devices, and miscellaneous control registers defined for compatibility. (The microcontroller's third PIC and the CBAR register are not found in PC/AT-compatible systems.)

Table 1-2 lists all of the direct-mapped I/O registers in the ÉlanSC520 microcontroller.

Table 1-2 Direct-Mapped I/O Registers

Register Name	Mnemonic	I/O Address	Page Number
Slave DMA		**0000–000Fh**	
Slave DMA Channel 0 Memory Address	GPDMA0MAR	0000h	page 11-42
Slave DMA Channel 0 Transfer Count	GPDMA0TC	0001h	page 11-43
Slave DMA Channel 1 Memory Address	GPDMA1MAR	0002h	page 11-44
Slave DMA Channel 1 Transfer Count	GPDMA1TC	0003h	page 11-45
Slave DMA Channel 2 Memory Address	GPDMA2MAR	0004h	page 11-46
Slave DMA Channel 2 Transfer Count	GPDMA2TC	0005h	page 11-47
Slave DMA Channel 3 Memory Address	GPDMA3MAR	0006h	page 11-48
Slave DMA Channel 3 Transfer Count	GPDMA3TC	0007h	page 11-49
Slave DMA Channel 0–3 Status	SLDMASTA	0008h	page 11-50
Slave DMA Channel 0–3 Control	SLDMACTL	0008h	page 11-51
Slave Software DRQ(n) Request	SLDMASWREQ	0009h	page 11-53
Slave DMA Channel 0–3 Mask	SLDMAMSK	000Ah	page 11-54
Slave DMA Channel 0–3 Mode	SLDMAMODE	000Bh	page 11-55
Slave DMA Clear Byte Pointer	SLDMACBP	000Ch	page 11-57
Slave DMA Controller Reset	SLDMARST	000Dh	page 11-58
Slave DMA Controller Temporary	SLDMATMP	000Dh	page 11-59
Slave DMA Mask Reset	SLDMAMSKRST	000Eh	page 11-60
Slave DMA General Mask	SLDMAGENMSK	000Fh	page 11-61

Table 1-2 Direct-Mapped I/O Registers (Continued)

Register Name	Mnemonic	I/O Address	Page Number
Master Programmable Interrupt Controller		**0020, 0021h**	
Master PIC Interrupt Request	MPICIR	0020h	page 12-24
Master PIC In-Service	MPICISR	0020h	page 12-25
Master PIC Initialization Control Word 1	MPICICW1	0020h	page 12-26
Master PIC Operation Control Word 2	MPICOCW2	0020h	page 12-28
Master PIC Operation Control Word 3	MPICOCW3	0020h	page 12-30
Master PIC Initialization Control Word 2	MPICICW2	0021h	page 12-32
Master PIC Initialization Control Word 3	MPICICW3	0021h	page 12-33
Master PIC Initialization Control Word 4	MPICICW4	0021h	page 12-35
Master PIC Interrupt Mask	MPICINTMSK	0021h	page 12-36
Slave 2 PIC		**0024h, 0025h**	
Slave 2 PIC Interrupt Request	S2PICIR	0024h	page 12-37
Slave 2 PIC In-Service	S2PICISR	0024h	page 12-38
Slave 2 PIC Initialization Control Word 1	S2PICICW1	0024h	page 12-39
Slave 2 PIC Operation Control Word 2	S2PICOCW2	0024h	page 12-41
Slave 2 PIC Operation Control Word 3	S2PICOCW3	0024h	page 12-43
Slave 2 PIC Initialization Control Word 2	S2PICICW2	0025h	page 12-45
Slave 2 PIC Initialization Control Word 3	S2PICICW3	0025h	page 12-46
Slave 2 PIC Initialization Control Word 4	S2PICICW4	0025h	page 12-47
Slave 2 PIC Interrupt Mask	S2PICINTMSK	0025h	page 12-48
Programmable Interval Timer		**0040–0043h**	
PIT Channel 0 Count	PIT0CNT	0040h	page 13-2
PIT Channel 1 Count	PIT1CNT	0041h	page 13-3
PIT Channel 2 Count	PIT2CNT	0042h	page 13-4
PIT 0 Status	PIT0STA	0040h	page 13-5
PIT 1 Status	PIT1STA	0041h	page 13-5
PIT 2 Status	PIT2STA	0042h	page 13-5
PIT Mode Control	PITMODECTL	0043h	page 13-7
PIT Counter Latch Command	PITCNTLAT	0043h	page 13-10
PIT Read-Back Command	PITRDBACK	0043h	page 13-11
SCP Data Port	**SCPDATA**	**0060h**	page 3-7
System Control Port B	**SYSCTLB**	**0061h**	page 13-13
SCP Command Port	**SCPCMD**	**0064h**	page 3-8
RTC Index, Data		**0070h, 0071h**	
RTC/CMOS RAM Index	RTCIDX	0070h	page 17-2
RTC/CMOS RAM Data Port	RTCDATA	0071h	page 17-3
DMA Page and General Registers		**0080–008Fh**	
General 0	GPDMAGR0	0080h	page 11-62
Slave DMA Channel 2 Page	GPDMA2PG	0081h	page 11-63
Slave DMA Channel 3 Page	GPDMA3PG	0082h	page 11-64
Slave DMA Channel 1 Page	GPDMA1PG	0083h	page 11-65

Table 1-2 Direct-Mapped I/O Registers (Continued)

Register Name	Mnemonic	I/O Address	Page Number
General 1	GPDMAGR1	0084h	page 11-66
General 2	GPDMAGR2	0085h	page 11-67
General 3	GPDMAGR3	0086h	page 11-68
Slave DMA Channel 0 Page	GPDMA0PG	0087h	page 11-69
General 4	GPDMAGR4	0088h	page 11-70
Master DMA Channel 6 Page	GPDMA6PG	0089h	page 11-71
Master DMA Channel 7 Page	GPDMA7PG	008Ah	page 11-72
Master DMA Channel 5 Page	GPDMA5PG	008Bh	page 11-73
General 5	GPDMAGR5	008Ch	page 11-74
General 6	GPDMAGR6	008Dh	page 11-75
General 7	GPDMAGR7	008Eh	page 11-76
General 8	GPDMAGR8	008Fh	page 11-77
System Control Port A	**SYSCTLA**	**0092h**	page 3-9
Slave 1 PIC		**00A0–00A1h**	
Slave 1 PIC Interrupt Request	S1PICIR	00A0h	page 12-49
Slave 1 PIC In-Service	S1PICISR	00A0h	page 12-50
Slave 1 PIC Initialization Control Word 1	S1PICICW1	00A0h	page 12-51
Slave 1 PIC Operation Control Word 2	S1PICOCW2	00A0h	page 12-53
Slave 1 PIC Operation Control Word 3	S1PICOCW3	00A0h	page 12-55
Slave 1 PIC Initialization Control Word 2	S1PICICW2	00A1h	page 12-57
Slave 1 PIC Initialization Control Word 3	S1PICICW3	00A1h	page 12-58
Slave 1 PIC Initialization Control Word 4	S1PICICW4	00A1h	page 12-59
Slave 1 PIC Interrupt Mask	S1PICINTMSK	00A1h	page 12-60
Master DMA		**00C0–00DEh**	
Master DMA Channel 4 Memory Address	GPDMA4MAR	00C0h	page 11-78
Master DMA Channel 4 Transfer Count	GPDMA4TC	00C2h	page 11-79
Master DMA Channel 5 Memory Address	GPDMA5MAR	00C4h	page 11-80
Master DMA Channel 5 Transfer Count	GPDMA5TC	00C6h	page 11-81
Master DMA Channel 6 Memory Address	GPDMA6MAR	00C8h	page 11-82
Master DMA Channel 6 Transfer Count	GPDMA6TC	00CAh	page 11-83
Master DMA Channel 7 Memory Address	GPDMA7MAR	00CCh	page 11-84
Master DMA Channel 7 Transfer Count	GPDMA7TC	00CEh	page 11-85
Master DMA Channel 4–7 Status	MSTDMASTA	00D0h	page 11-86
Master DMA Channel 4–7 Control	MSTDMACTL	00D0h	page 11-87
Master Software DRQ(n) Request	MSTDMASWREQ	00D2h	page 11-89
Master DMA Channel 4–7 Mask	MSTDMAMSK	00D4h	page 11-90
Master DMA Channel 4–7 Mode	MSTDMAMODE	00D6h	page 11-91
Master DMA Clear Byte Pointer	MSTDMACBP	00D8h	page 11-93
Master DMA Controller Reset	MSTDMARST	00DAh	page 11-94
Master DMA Controller Temporary	MSTDMATMP	00DAh	page 11-95
Master DMA Mask Reset	MSTDMAMSKRST	00DCh	page 11-96
Master DMA General Mask	MSTDMAGENMSK	00DEh	page 11-97

Table 1-2 Direct-Mapped I/O Registers (Continued)

Register Name	Mnemonic	I/O Address	Page Number
Floating Point Error Interrupt Clear	**FPUERRCLR**	**F0h**	page 12-61
UART 2		**02F8–02FFh**	
UART 2 Transmit Holding	UART2THR	02F8h	page 18-7
UART 2 Receive Buffer	UART2RBR	02F8h	page 18-8
UART 2 Baud Clock Divisor Latch LSB	UART2BCDL	02F8h	page 18-9
UART 2 Baud Clock Divisor Latch MSB	UART2BCDH	02F9h	page 18-10
UART 2 Interrupt Enable	UART2INTENB	02F9h	page 18-11
UART 2 Interrupt ID	UART2INTID	02FAh	page 18-12
UART 2 FIFO Control	UART2FCR	02FAh	page 18-15
UART 2 Line Control	UART2LCR	02FBh	page 18-17
UART 2 Modem Control	UART2MCR	02FCh	page 18-19
UART 2 Line Status	UART2LSR	02FDh	page 18-21
UART 2 Modem Status	UART2MSR	02FEh	page 18-23
UART 2 Scratch Pad	UART2SCRATCH	02FFh	page 18-25
UART 1		**03F8–03FFh**	
UART 1 Transmit Holding	UART1THR	03F8h	page 18-7
UART 1 Receive Buffer	UART1RBR	03F8h	page 18-8
UART 1 Baud Clock Divisor Latch LSB	UART1BCDL	03F8h	page 18-9
UART 1 Baud Clock Divisor Latch MSB	UART1BCDH	03F9h	page 18-10
UART 1 Interrupt Enable	UART1INTENB	03F9h	page 18-11
UART 1 Interrupt ID	UART1INTID	03FAh	page 18-12
UART 1 FIFO Control	UART1FCR	03FAh	page 18-15
UART 1 Line Control	UART1LCR	03FBh	page 18-17
UART 1 Modem Control	UART1MCR	03FCh	page 18-19
UART 1 Line Status	UART1LSR	03FDh	page 18-21
UART 1 Modem Status	UART1MSR	03FEh	page 18-23
UART 1 Scratch Pad	UART1SCRATCH	03FFh	page 18-25
PCI Bus Configuration Space Index/Data (32-bit)		**0CF8h, 0CFCh**	
PCI Configuration Address	PCICFGADR	0CF8h	page 6-15
PCI Configuration Data	PCICFGDATA	0CFCh	page 6-17
Configuration Base Address	**CBAR**	**FFFCh**	page 2-9

1.3 PCI HOST BRIDGE INDEXED CONFIGURATION REGISTERS

The ÉlanSC520 microcontroller's PCI Host Bridge supports the required PCI configuration space header registers, plus a Host Bridge-specific Master Retry Time-Out (PCIMRETRYTO) register defined in the PCI configuration space. Note that additional microcontroller-specific PCI Host Bridge configuration registers are provided in the MMCR space (see Table 1-1).

PCI indexed registers are accessed via two 32-bit direct-mapped I/O locations: Port 0CF8h and Port 0CFCh. The PCI configuration mechanism can be used to access either the Host Bridge-specific PCI indexed registers, or the device-specific PCI indexed registers for any

other device that is connected to the PCI bus. Refer to the *PCI Local Bus Specification,* Revision 2.2, for details on PCI configuration.

Table 1-3 lists all of the PCI indexed registers in the ÉlanSC520 microcontroller.

Table 1-3 PCI Indexed Registers

Register Name	Mnemonic	I/O Address	PCI Index	Page Number
Device/Vendor ID	PCIDEVID	0CF8h/0CFCh	00h	page 6-18
Status/Command	PCISTACMD	0CF8h/0CFCh	04h	page 6-19
Class Code/Revision ID	PCICCREVID	0CF8h/0CFCh	08h	page 6-22
Header Type	PCIHEADTYPE	0CF8h/0CFCh	0Eh	page 6-23
Master Retry Time-Out	PCIMRETRYTO	0CF8h/0CFCh	41h	page 6-24

1.4 RTC AND CMOS RAM INDEXED REGISTERS

Real-time clock and CMOS RAM indexed registers are accessed using I/O ports 0070h (index) and 0071h (data). These registers provide PC/AT-compatible setup, control, and status functions for the RTC, as well as user CMOS RAM locations.

Table 1-4 lists all of the RTC indexed registers in the ÉlanSC520 microcontroller.

Table 1-4 Real-Time Clock Indexed Registers

Register Name	Mnemonic	I/O Address	RTC Index	Page Number
RTC Current Second	RTCCURSEC	70h/71h	00h	page 17-4
RTC Alarm Second	RTCALMSEC	70h/71h	01h	page 17-5
RTC Current Minute	RTCCURMIN	70h/71h	02h	page 17-6
RTC Alarm Minute	RTCALMMIN	70h/71h	03h	page 17-7
RTC Current Hour	RTCCURHR	70h/71h	04h	page 17-8
RTC Alarm Hour	RTCALMHR	70h/71h	05h	page 17-9
RTC Current Day of the Week	RTCCURDOW	70h/71h	06h	page 17-10
RTC Current Day of the Month	RTCCURDOM	70h/71h	07h	page 17-11
RTC Current Month	RTCCURMON	70h/71h	08h	page 17-12
RTC Current Year	RTCCURYR	70h/71h	09h	page 17-13
RTC Control A	RTCCTLA	70h/71h	0Ah	page 17-14
RTC Control B	RTCCTLB	70h/71h	0Bh	page 17-16
RTC Status C	RTCSTAC	70h/71h	0Ch	page 17-18
RTC Status D	RTCSTAD	70h/71h	0Dh	page 17-20
General-Purpose CMOS RAM (114 bytes)	RTCCMOS	70h/71h	0E–7Fh	page 17-21

CHAPTER

2 SYSTEM ADDRESS MAPPING REGISTERS

2.1 OVERVIEW

This chapter describes the system memory and I/O address mapping registers of the ÉlanSC520 microcontroller.

The system address mapping register set consists of 18 memory-mapped configuration region (MMCR) registers and one direct-mapped I/O register used to define the memory and I/O map in an ÉlanSC520 microcontroller-based system, and to control specific attributes of SDRAM and ROM regions. See the *Élan™SC520 Microcontroller User's Manual*, order #22004, for details about memory and I/O space.

Table 2-1 and Table 2-2 list each type of memory and I/O space register in offset order, with the corresponding description's page number.

2.2 REGISTERS

Table 2-1 System Address Mapping MMCR Registers

Register Name	Mnemonic	MMCR Offset	Page Number
Address Decode Control	ADDDECCTL	80h	page 2-2
Write-Protect Violation Status	WPVSTA	82h	page 2-4
Programmable Address Region 0	PAR0	88h	page 2-5
Programmable Address Region 1	PAR1	8Ch	page 2-5
Programmable Address Region 2	PAR2	90h	page 2-5
Programmable Address Region 3	PAR3	94h	page 2-5
Programmable Address Region 4	PAR4	98h	page 2-5
Programmable Address Region 5	PAR5	9Ch	page 2-5
Programmable Address Region 6	PAR6	A0h	page 2-5
Programmable Address Region 7	PAR7	A4h	page 2-5
Programmable Address Region 8	PAR8	A8h	page 2-5
Programmable Address Region 9	PAR9	ACh	page 2-5
Programmable Address Region 10	PAR10	B0h	page 2-5
Programmable Address Region 11	PAR11	B4h	page 2-5
Programmable Address Region 12	PAR12	B8h	page 2-5
Programmable Address Region 13	PAR13	BCh	page 2-5
Programmable Address Region 14	PAR14	C0h	page 2-5
Programmable Address Region 15	PAR15	C4h	page 2-5

Table 2-2 System Address Mapping Direct-Mapped Register

Register Name	Mnemonic	I/O Address	Page Number
Configuration Base Address	CBAR	FFFCh	page 2-9

Address Decode Control (ADDDECCTL)

Memory-Mapped MMCR Offset 80h

	7	6	5	4	3	2	1	0
Bit	WPV_INT_ ENB	Reserved		IO_HOLE_ DEST	Reserved	RTC_DIS	UART2_DIS	UART1_DIS
Reset	0	0	0	0	0	0	0	0
R/W	R/W	RSV		R/W	RSV	R/W	R/W	R/W

Register Description

This register controls miscellaneous functions in the ÉlanSC520 microcontroller address decode block.

Bit Definitions

Bit	Name	Function
7	WPV_INT_ ENB	**Write-Protect Violation Interrupt Enable** This bit enables an interrupt request to be generated when a write-protect violation occurs. 0 = Write-protect violations do not generate an interrupt request to the CPU. 1 = Write-protect violations generate an interrupt request to the CPU. Before the WPV_INT_ENB bit is set, the WPVMAP register (see page 12-21) must be configured to route the interrupt to the appropriate interrupt request level and priority. The initiator of the access that caused the write-protect violation can be the Am5$_x$86 CPU, any PCI bus master, or the GP bus DMA controller. This interrupt can only be generated if at least one of the PARx register windows is enabled and has the write-protect attribute selected (see page 2-6).
6-5	Reserved	**Reserved** This bit field should be written to 0 for normal system operation.
4	IO_HOLE_ DEST	**I/O Hole Access Destination** This bit determines the destination of accesses performed by the Am5$_x$86 CPU to certain I/O addresses in the range 0000h to 03FFh. 0 = The accesses are forwarded to the external GP bus (default). 1 = The accesses are forwarded to the PCI bus I/O space. I/O addresses in the range 0000h to 03FFh that are not occupied by the internal GP bus peripherals are normally reserved for PC/AT-compatible peripherals. Such addresses are referred to as *holes* in the I/O address space. See the memory and I/O chapter of the *Élan™SC520 Microcontroller User's Manual*, order #22004, for details about these holes. If a PARx register (see page 2-6) is configured to address GP bus I/O space within a hole, accesses in the defined region are forwarded to the GP bus regardless of the IO_HOLE_DEST bit value. The PARx window must not overlap any of the internal peripherals' direct-mapped I/O addresses.
3	Reserved	**Reserved** This bit field should be written to 0 for normal system operation.

Bit	Name	Function
2	RTC_DIS	**RTC Disable** This bit causes the integrated RTC to be disabled. 0 = The integrated RTC is enabled. 1 = The integrated RTC is not used, and accesses to the RTC address space are forwarded externally to the GP bus. When the internal RTC is disabled, the corresponding interrupt request is not automatically disconnected from the PIC. If an external RTC is to drive the RTC interrupt request, then all interrupt enables in the RTCCTLB register must be cleared prior to disabling the internal RTC (see page 17-16). After the internal RTC is disabled, a PARx window can be defined to target a GPCSx chip select in the RTC I/O address space. If a PARx window is not used, the external RTC must fully decode the addresses.
1	UART2_DIS	**UART 2 Disable** This bit causes the integrated UART 2 to be disabled. 0 = The integrated UART 2 is enabled. 1 = The integrated UART 2 is not used, and accesses to the UART 2 I/O address space are forwarded externally to the GP bus. When the internal UART 2 is disabled, the corresponding interrupt request is not automatically disconnected from the PIC. If an external UART is to drive the UART 2 interrupt request, then all interrupt enables in the UART2CTL and UART2INTENB registers must be cleared (see page 18-3 and page 18-11). After the internal UART 2 is disabled, a PARx window can be defined to target a GPCSx chip select in the UART 2 I/O address space. If a PARx window is not used, the external UART must fully decode the addresses.
0	UART1_DIS	**UART 1 Disable** This bit causes the integrated UART 1 to be disabled. 0 = The integrated UART 1 is enabled. 1 = The integrated UART 1 is not used, and accesses to the UART 1 I/O address space are forwarded externally to the GP bus. When the internal UART 1 is disabled, the corresponding interrupt request is not automatically disconnected from the PIC. If an external UART is to drive the UART 1 interrupt request, then all interrupt enables in the UART1CTL and UART1INTENB registers must be cleared (see page 18-3 and page 18-11). After the internal UART 1 is disabled, a PARx window can be defined to target a GPCSx chip select in the UART 1 I/O address space. If a PARx window is not used, the external UART must fully decode the addresses.

Programming Notes

When the ÉlanSC520 microcontroller comes out of reset, the internal RTC and UARTs are enabled. If the system application requires the use of an external RTC or UARTs, the internal devices should be disabled during the boot process and initialization to prevent potential conflicts.

When the integrated UARTs are disabled, only the I/O space accesses associated with these peripherals are forwarded externally. The accesses to the MMCR registers for the UARTs are not forwarded externally because these registers are specific to the integrated peripherals. Therefore, the UART MMCR registers should not be used while the integrated UARTs are disabled.

Even if the internal RTC or UARTs are disabled, setting the IO_HOLE_DEST bit does *not* redirect accesses to the disabled peripherals to PCI bus I/O space.

Write-Protect Violation Status (WPVSTA)

Memory-Mapped
MMCR Offset 82h

	15	14	13	12	11	10	9	8
Bit	WPV_STA	Reserved					WPV_MSTR[1–0]	
Reset	0	0	0	0	0	0	0	0
R/W	R/W!	RSV					R	

	7	6	5	4	3	2	1	0
Bit	Reserved				WPV_WINDOW[3–0]			
Reset	0	0	0	0	0	0	0	0
R/W	RSV				R			

Register Description

This register provides write-protect violation status for the ÉlanSC520 microcontroller's address decode block.

Bit Definitions

Bit	Name	Function
15	WPV_STA	**Write-Protect Violation Interrupt Status** This bit indicates whether an interrupt request is pending due to a write-protect violation. 0 = Write-protect violations interrupt request not pending. 1 = Write-protect violations interrupt request pending. Software must write a 1 to this bit to clear the interrupt (and the status bit) before the hardware can latch subsequent write-protect violations. This bit provides the status of the interrupt regardless of the setting of the WPV_INT_ENB bit in the ADDDECCTL register (see page 2-2).
14–10	Reserved	**Reserved** This bit field should be written to 0 for normal system operation.
9–8	WPV_MSTR [1–0]	**Write-Protect Violation Master** This bit field identifies the master that caused the write-protect violation. 00 = The Am5$_x$86 CPU caused the violation. 01 = A PCI bus master caused the violation. 10 = The GP bus DMA controller caused the violation. 11 = Reserved. This bit field contains valid information only while the WPV_STA bit is set. The master's identity is latched when a write-protect violation occurs. Subsequent write-protect violations are not captured until software clears the interrupt by writing a 1 to the WPV_STA bit.
7–4	Reserved	**Reserved** This bit field should be written to 0 for normal system operation.
3–0	WPV_WINDOW [3–0]	**Write-Protect Violation Window Number** This bit field identifies the programmable address region window (PAR0–PAR15) in which the write-protect violation occurred. This bit field contains valid information only while the WPV_STA bit is set. The PARx window number is latched when a write-protect violation occurs. Subsequent write-protect violations are not captured until software clears the interrupt by writing a 1 to the WPV_STA bit.

Programming Notes

	Memory-Mapped
Programmable Address Region 0 (PAR0)	**MMCR Offset 88h**
Programmable Address Region 1 (PAR1)	**MMCR Offset 8Ch**
Programmable Address Region 2 (PAR2)	**MMCR Offset 90h**
Programmable Address Region 3 (PAR3)	**MMCR Offset 94h**
Programmable Address Region 4 (PAR4)	**MMCR Offset 98h**
Programmable Address Region 5 (PAR5)	**MMCR Offset 9Ch**
Programmable Address Region 6 (PAR6)	**MMCR Offset A0h**
Programmable Address Region 7 (PAR7)	**MMCR Offset A4h**
Programmable Address Region 8 (PAR8)	**MMCR Offset A8h**
Programmable Address Region 9 (PAR9)	**MMCR Offset ACh**
Programmable Address Region 10 (PAR10)	**MMCR Offset B0h**
Programmable Address Region 11 (PAR11)	**MMCR Offset B4h**
Programmable Address Region 12 (PAR12)	**MMCR Offset B8h**
Programmable Address Region 13 (PAR13)	**MMCR Offset BCh**
Programmable Address Region 14 (PAR14)	**MMCR Offset C0h**
Programmable Address Region 15 (PAR15)	**MMCR Offset C4h**

	31	30	29	28	27	26	25	24
Bit	TARGET[2–0]			ATTR[2–0]			PG_SZ	SZ_ST_ADR[24]
Reset	0	0	0	0	0	0	0	0
R/W	R/W			R/W			R/W	R/W

	23	22	21	20	19	18	17	16
Bit	SZ_ST_ADR[23–16]							
Reset	0	0	0	0	0	0	0	0
R/W	R/W							

	15	14	13	12	11	10	9	8
Bit	SZ_ST_ADR[15–8]							
Reset	0	0	0	0	0	0	0	0
R/W	R/W							

	7	6	5	4	3	2	1	0
Bit	SZ_ST_ADR[7–0]							
Reset	0	0	0	0	0	0	0	0
R/W	R/W							

Register Description

These registers can be used to map windows of memory address space to SDRAM, ROM, PCI bus, or GP bus; or to map windows of I/O space to the GP bus. These registers are also used to apply noncacheability, write-protect, and nonexecutable attributes to SDRAM and ROM regions.

Bit Definitions

Bit	Name	Function
31–29	TARGET[2–0]	**Target of the PARx Window** This bit field defines the target (destination) of the PARx window being configured. If no PARx window is enabled (i.e., the TARGET bit field is 000b in all PARx registers), the default memory map for the lower 1 Gbyte address range (00000000h–3FFFFFFFh) is either SDRAM (if enabled) or PCI. Accesses to any external GP bus devices or devices connected to $\overline{\text{ROMCS1}}$ or $\overline{\text{ROMCS2}}$ require a PARx window to be enabled and configured to access them. 000 = The window is disabled. All other bits in the PARx register are ignored. 001 = The window targets a GP bus I/O region. This is the only I/O space access possible through a PARx window. All other settings refer to the memory address space. For GP bus windows, the ATTR bit field selects a GP bus chip select. See the programming notes for this register, on page 2-8, for GP-bus I/O address restrictions. 010 = The window targets a GP bus memory region. For GP bus windows, the ATTR bit field selects a GP bus chip select. 011 = The window targets a PCI bus memory region. Only PAR0 and PAR1 support a PCI bus memory region as a target. Define a PCI bus window only when attempting to forward cycles to the PCI bus in the lower 256 Mbyte address range (00000000h to 0FFFFFFFh). 100 = The window targets a $\overline{\text{BOOTCS}}$ region (boot ROM or Flash memory device). The ÉlanSC520 microcontroller forces the $\overline{\text{BOOTCS}}$ signal active at system reset, but the boot code must initialize a PARx register to decode the required space for the ROM. Refer to the *Élan™SC520 Microcontroller User's Manual*, order #22004, for details on configuring the boot ROM during system initialization. 101 = The window targets a $\overline{\text{ROMCS1}}$ region (ROM or Flash memory devices). 110 = The window targets a $\overline{\text{ROMCS2}}$ region (ROM or Flash memory devices). 111 = The window targets an SDRAM region. A PARx register is not required to access SDRAM. Target an SDRAM region only when applying one of the attributes described in the ATTR bit field.

Bit	Name	Function
28–26	ATTR[2–0]	**Attribute** If the TARGET bit field selects the PCI bus (for PAR0 and PAR1 only), this bit field (ATTR) is ignored. If the TARGET bit field selects GP bus I/O or memory, this bit field (ATTR) specifies the GP bus chip select that is targeted, as follows: 000 = $\overline{\text{GPCS0}}$ 001 = $\overline{\text{GPCS1}}$ 010 = $\overline{\text{GPCS2}}$ 011 = $\overline{\text{GPCS3}}$ 100 = $\overline{\text{GPCS4}}$ 101 = $\overline{\text{GPCS5}}$ 110 = $\overline{\text{GPCS6}}$ 111 = $\overline{\text{GPCS7}}$ If the TARGET bit field selects SDRAM or one of the ROM/Flash chip selects, each bit in this bit field (ATTR) specifies an attribute. The attribute bits for the region can be set or cleared independently, with the following effect: 0xx = Code execution is allowed in programmed memory region. 1xx = Code execution is prevented in programmed memory region. An attempt to execute code results in a processor exception and returns an illegal operand code (FFFFh). This attribute is useful for debugging software; for example, to prevent a program from erroneously executing out of a data region. x0x = Caching is enabled for the programmed memory region ($Am5_x86$ CPU cache). x1x = Caching is disabled for the programmed memory region. This attribute can be used to minimize cache write-back cycle overhead in regions that are shared between the $Am5_x86$ CPU and PCI bus masters or GP bus DMA devices. Software must flush the cache immediately after setting this attribute. xx0 = Writes are enabled in the programmed memory region. xx1 = Write protection is enabled (writes are disabled) in the programmed memory region. Write protection prevents writes from occurring in the region's address range. Write-protect violations are reported by the WPVSTA register (see page 2-4), and a corresponding interrupt can be enabled via the WPV_INT_ENB bit in the ADDDECCTL register (see page 2-2).
25	PG_SZ	**Page Size** This bit selects the page size of the memory region defined by the SZ_ST_ADR bit field. This bit is ignored during I/O cycles. 0 = 4-Kbyte page size selected 1 = 64-Kbyte page size selected Each PARx window's total size in memory space is defined by the combined settings of the PG_SZ and SZ_ST_ADR bit fields. The PG_SZ bit value affects the window start address resolution, the window size resolution, and the total size that is possible for the window.

Bit	Name	Function
24–0	SZ_ST_ADR [24–0]	**Region Size/Start Address** This bit field, in conjunction with the PG_SZ bit, is used to specify the total region size and the starting address of the programmed address space. This bit field is used in one of three ways: ***For Memory Space Regions with 4-Kbyte Pages:*** ■ The PG_SZ bit is 0. ■ The SZ_ST_ADR[24–18] bit field is used to specify a memory space size of up to 128 pages, each 4 Kbytes in size, for a maximum PARx window size of 512 Kbytes. Pages start on 4-Kbyte boundaries. (A value of 00h specifies one page; 7Fh specifies 128 pages.) ■ The SZ_ST_ADR[17–0] bit field is used to define the starting page of the region within the memory address space. The SZ_ST_ADR[17–0] bit field is compared to internal $Am5_x86$ CPU bus signals a29–a12. ***For Memory Space Regions with 64-Kbyte Pages:*** ■ The PG_SZ bit is 1. ■ The SZ_ST_ADR[24–14] bit field is used to specify a memory space size of up to 2048 pages, each 64 Kbytes in size, for a maximum PARx window size of 128 Mbytes. Pages start on 64-Kbyte boundaries. (A value of 000h specifies one page; 7FFh specifies 2048 pages.) ■ The S Z_ST_ADR[13–0] bit field is used to define the starting page of the region within the memory address space. The SZ_ST_ADR[13–0] bit field is compared to internal $Am5_x86$ CPU bus signals a29–a16. ***For I/O Space Regions:*** ■ The PG_SZ bit is ignored. ■ The SZ_ST_ADR[24–16] bit field is used to specify an I/O space size of byte granularity, for a total size up to 512 bytes. (A value of 000h specifies one byte; 1FFh specifies 512 bytes.) ■ The SZ_ST_ADR[15–0] bit field is used to define the starting address of the window within the 64 -Kbyte I/O space. The SZ_ST_ADR[15–0] bit field is compared to internal $Am5_x86$ CPU bus signals a15–a0. ***Note:*** *If a larger window than the maximum size is required, multiple PARx registers can be used.*

Programming Notes

Each PARx register must be written as a full 32-bit doubleword.

The basic trade-off with setting the page size in a PARx register is the granularity of the memory region. The smaller page size restricts the total size of the region, but allows the smaller granularity of 4 Kbytes. The larger 64-Kbyte page size is an option when the total region size must be larger than 512 Kbytes, but this requires the pages to start on 64-Kbyte boundaries.

If two PARx windows overlap, the lower-numbered PARx register's target has priority. For example, if a memory or I/O address falls within the windows defined by both the PAR4 register and the PAR 13 register, reads or writes to that address go to the PAR4 register's target, not the one defined in the PAR13 register.

If a PARx window overlaps the MMCR alias defined by the CBAR register (see page 2-9), the MMCR alias has priority, with the following exception: if a PAR window is configured for PCI, *and* the CBAR register is programmed to overlap with this PAR window, *and* the PAR window is placed below the top of DRAM, then the MMCR is not given priority over the PCI access. This configuration could result in system errors due to concurrence of both PCI and internal MMCR accesses.

PARx windows in the GP bus I/O space must not include any of the following direct-mapped register addresses: CBAR (Port FFFCh), PCICFGADR (Port 0CF8h), or PCICFGDATA (Port 0CFCh). Also, the PARx window must not overlap any direct-mapped I/O address belonging to an internal peripheral (i.e., GP bus DMA, PIC, PIT, system control ports, RTC, or UART I/O registers).

When programming a PARx register for GP bus I/O space, it is best to start the space on a doubleword boundary. If an unaligned byte region is specified for I/O access, the software that accesses the region must directly address the correct byte or bytes. For example, if a PARx register is programmed for an I/O region starting at address xxx1h (i.e., byte1 of the associated doubleword), then when the CPU performs a word or doubleword access, the entire access is redirected to the PCI bus, and byte 1 is not accessed on the GP bus as programmed. In this case the byte requested *must* be directly accessed by the CPU at I/O address xxx1h.

Configuration Base Address (CBAR)

Direct-Mapped I/O Address FFFCh

	31	30	29	28	27	26	25	24
Bit	ENABLE	Reserved	ADR[29–24]					
Reset	0	0	0	0	0	0	0	0
R/W	R/W!	RSV	R/W					

	23	22	21	20	19	18	17	16
Bit	ADR[23–16]							
Reset	0	0	0	0	0	0	0	0
R/W	R/W							

	15	14	13	12	11	10	9	8
Bit	ADR[15–12]				Reserved			
Reset	0	0	0	0	0	0	0	0
R/W	R/W				RSV			

	7	6	5	4	3	2	1	0
Bit	MATCH[7–0]							
Reset	0	0	0	0	0	0	0	0
R/W	W							

Register Description

This register can be used to alias the memory-mapped configuration region (MMCR) that is used to access many of the ÉlanSC520 microcontroller's integrated peripheral functions. The MMCR is always accessible at FFFEF000–FFFEFFFFh, in the region directly below the boot ROM space, but it can also be aliased to any 4-Kbyte boundary within the first 1 Gbyte of memory space.

Bit Definitions

Bit	Name	Function
31	ENABLE	**Enable** This bit must be set to enable write access to the CBAR register. Upon reading, the ENABLE bit returns the state of the MMCR alias. 0 = MMCR alias is disabled. 1 = MMCR alias is enabled. Writes to this bit are ignored if the value CBh is not written to the MATCH bit field in the same write cycle.
30	Reserved	**Reserved** This bit field should be written to 0 for normal system operation.
29–12	ADR[29–12]	**Start Address** This bit field defines the starting address of the memory-mapped configuration region on a 4-Kbyte boundary. The address programmed in this bit field is compared to internal $Am5_x86$ CPU bus signals a29–a12. Writes to this bit field are ignored unless the ENABLE bit set and the value CBh is written to the MATCH bit field in the same write cycle. Both of these conditions are required for the write to take effect.

Bit	Name	Function
11–8	Reserved	**Reserved** This bit field should be written to 0 for normal system operation.
7–0	MATCH[7–0]	**Match** The bit field is used to prevent illegal writes to this register. The data pattern written to this byte must be CBh. If any other data pattern is written to these bits, the entire 32-bit write is ignored. This bit field returns 0 when read.

Programming Notes

This register must be written as a full 32-bit doubleword.

When the MMCR alias is enabled, the 4-Kbyte address range to which the alias is mapped becomes the MMCR, regardless of any PARx registers that are enabled in the same 4-Kbyte address range. In effect, the MMCR alias has higher priority than any of the PARx windows that are enabled.

CHAPTER

3 RESET GENERATION REGISTERS

3.1 OVERVIEW

This chapter describes the reset generation and reset-related registers of the ÉlanSC520 microcontroller.

The ÉlanSC520 microcontroller provides several microcontroller-specific and PC/AT-compatible reset functions.

The reset generation register set includes two groups of registers:

- Three memory-mapped configuration region (MMCR) registers are used to provide system board information at reset, configure reset behavior, and provide reset status.
- Three direct-mapped I/O registers are used to provide PC/AT-compatible reset-related functions.

See the *Élan™SC520 Microcontroller User's Manual*, order #22004, for details about reset generation.

Table 3-1 and Table 3-2 list each type of reset register in offset order, with the corresponding description's page number.

3.2 REGISTERS

Table 3-1 Reset Generation MMCR Registers

Register Name	Mnemonic	MMCR Offset	Page Number
System Board Information	SYSINFO	D70h	page 3-2
Reset Configuration	RESCFG	D72h	page 3-3
Reset Status	RESSTA	D74h	page 3-5

Table 3-2 Reset Generation Direct-Mapped Registers

Register Name	Mnemonic	I/O Address	Page Number
SCP Data Port	SCPDATA	0060h	page 3-7
SCP Command Port	SCPCMD	0064h	page 3-8
System Control Port A	SYSCTLA	0092h	page 3-9

System Board Information (SYSINFO)

Memory-Mapped MMCR Offset D70h

	7	6	5	4	3	2	1	0
Bit	RST_LD[7–0]							
Reset	?	?	?	?	?	?	?	?
R/W	R							

Register Description

This read-only register contains the state that was latched on the RSTLD7–RSTLD0 pins at the assertion of PWRGOOD.

Bit Definitions

Bit	Name	Function
7–0	RST_LD[7–0]	**Reset Latched Input Data** The microcontroller initializes this bit field at the assertion of the PWRGOOD signal by latching the state of the shared RSTLD7–RSTLD0 pins. The information in this bit field remains static after it is latched.

Programming Notes

These are status-only bits used to provide static information to software. For example, the system hardware designer can configure pullup resistors as needed on the RSTLD7–RSTLD0 pins to provide board revision information to software. The pullup resistors' value should be approximately 10 KΩ.

Reset Configuration (RESCFG)

Memory-Mapped MMCR Offset D72h

	7	6	5	4	3	2	1	0
Bit	Reserved				ICE_ON_ RST	PRG_RST_ ENB	GP_RST	SYS_RST
Reset	0	0	0	0	0	0	0	0
R/W	RSV				R/W!	R/W	R/W	R/W!

Register Description

This register provides a direct read/write port to program system reset and GP bus reset. It also provides a control bit to enable or disable the programmable system reset function of the PRGRESET pin and another control bit to enable the AMDebug™ technology.

Bit Definitions

Bit	Name	Function
7–4	Reserved	**Reserved** This bit field should be written to 0 for normal system operation.
3	ICE_ON_RST	**Enter AMDebug™ Technology Mode on Next Reset** Setting this bit enables the microprocessor to enter AMDebug technology mode after a hard or soft reset has been asserted to the $Am5_x86$ CPU. Writes to this bit are ignored if the AMDebug technology mode is not active. (The AMDebug technology mode is controlled by the software that drives the AMDebug technology port.) 0 = Do not enter AMDebug technology mode on a CPU hard or soft reset. 1 = Enable the microprocessor to enter AMDebug technology mode on a CPU hard or soft reset.
2	PRG_RST_ ENB	**Programmable Reset Enable** 0 = Programmable reset is disabled. 1 = Programmable reset is enabled. Setting this bit enables programmable resets, during which SDRAM configuration and SDRAM contents are preserved. After this bit is set, programmable resets can be generated by one of four sources: ■ PRGRESET pin ■ Watchdog timer time-out-generated reset ■ Software write to the SYS_RST bit (with the PRG_RST_ENB bit set in the same write) ■ AMDebug technology system reset If this bit (PRG_RST_ENB) is cleared, the PRGRESET pin has no function. The other programmable reset sources, if generated, initiate non-SDRAM-preserving system resets. This bit is cleared by default, so a toggle on the PRGRESET pin does not affect the ÉlanSC520 microcontroller.
1	GP_RST	**Software GP Bus Reset** Software can write to this bit to control the GPRESET pin and read back the pin's state. The internal registers are not affected. 0 = Deassert GP bus reset. 1 = Assert GP bus reset.

Bit	Name	Function
0	SYS_RST	**Software System Reset** Software can write to this bit to trigger a one-shot system reset. The last value written is what is read. 0 = Do not cause a one-shot system reset event to be generated. 1 = Cause a one-shot system reset event to be generated. Setting the SYS_RST bit also causes GP bus and PCI-Bus resets to be generated. If the PRG_RST_ENB bit is 1, setting the SYS_RST bit results in a system reset in which the SDRAM configuration is maintained.

Programming Notes

After a PWRGOOD reset, the programmable system reset feature is disabled and can be enabled only by software writing a 1 to the PRG_RST_ENB bit. If the PRGRESET pin is asserted after it has been enabled as a programmable reset, it causes a programmable reset, which does not reset the SDRAM controller configuration or contents. This allows the contents of SDRAM to be preserved.

The PWRGOOD pin always has higher priority than the PRGRESET pin. For example, if the PRGRESET pin is asserted and PWRGOOD deasserted, PWRGOOD is serviced, and the PWRGOOD reset event disables the PRGRESET function.

Unlike most other registers, the bits in this register (RESCFG) are only returned to their reset value by a PWRGOOD reset. They are not cleared by any other kind of reset.

The AMDebug technology trace information is preserved only if a soft reset is generated to the CPU.

See the *Élan™SC520 Microcontroller User's Manual*, order #22004, for details about reset generation. Table 3-3 on page 3-6 provides a summary of ÉlanSC520 microcontroller reset sources and effects.

Reset Status (RESSTA)

Memory-Mapped
MMCR Offset D74h

	7	6	5	4	3	2	1	0
Bit	Reserved	SCP_RST_DET	ICE_HRST_DET	ICE_SRST_DET	WDT_RST_DET	SD_RST_DET	PRGRST_DET	PWRGOOD_DET
Reset	0	0	0	0	0	0	0	1
R/W	RSV	R/W!	R/W!	R/W!	R/W!	R/W!	R/W!	R/W!

Register Description

This register provides status information on the various reset sources implemented in the ÉlanSC520 microcontroller: emulated system control processor (SCP)-generated reset, CPU shutdown, AMDebug technology system reset, AMDebug technology hard reset, watchdog timer, PWRGOOD signal, and PRGRESET signal.

Bit Definitions

Bit	Name	Function
7	Reserved	**Reserved** This bit field should be written to 0 for normal system operation.
6	SCP_RST_DET	**SCP Reset Detect** This bit is set when a soft reset is generated by the SCP emulation logic. Software clears this bit by writing a 1. 0 = No SCP reset was detected. 1 = The CPU soft reset event was from an SCP reset command. (See the SCPCMD register description on page 3-8). A soft reset event clears the NMI_ENB bit in the PICICR register (see page 12-4). This allows software to initialize the stack pointer before setting the NMI_ENB bit again after a soft reset.
5	ICE_HRST_DET	**AMDebug™ Technology Hard Reset Detect** This bit is set when a AMDebug technology hard reset is detected. Software clears this bit by writing a 1. 0 = No AMDebug technology hard reset was detected. 1 = The CPU hard reset event was from an AMDebug technology hard reset.
4	ICE_SRST_DET	**AMDebug™ Technology System Reset Detect** This bit is set when an AMDebug technology system reset is detected. Software clears this bit by writing a 1. 0 = No AMDebug technology system reset was detected. 1 = The system reset event was from an AMDebug technology system reset. If the PRG_RST_ENB bit is 1 in the RESCFG register (see page 3-3), an AMDebug technology system reset results in a programmable reset in which the SDRAM configuration is maintained.
3	WDT_RST_DET	**Watchdog Timer Reset Detect** This bit is set when a watchdog timer system reset is detected. Software clears this bit by writing a 1. 0 = No watchdog timer reset was detected. 1 = The system reset event was from a watchdog timer time-out. If the PRG_RST_ENB bit is 1 in the RESCFG register (see page 3-3), a watchdog timer reset results in a programmable reset in which the SDRAM configuration is maintained.

Bit	Name	Function
2	SD_RST_DET	**CPU Shutdown Reset Detect** This bit is set when a CPU shutdown cycle (typically caused by a software “triple-fault”) is detected. Software clears this bit by writing a 1. 0 = No CPU shutdown cycle was detected. 1 = The CPU soft reset event was from a shutdown cycle. A soft reset event clears the NMI_ENB bit in the PICICR register (see page 12-4). This allows software to initialize the stack pointer before setting the NMI_ENB bit again after a soft reset.
1	PRGRST_DET	**PRGRESET Detect** This bit is set when a reset from PRGRESET pin is detected. Software clears this bit by writing a 1. 0 = No PRGRESET pin reset was detected. 1 = The system reset event was from the PRGRESET pin. If the PRG_RST_ENB bit is 1 in the RESCFG register (see page 3-3), assertion of the PRGRESET pin while PWRGOOD is asserted results in a programmable reset in which the SDRAM configuration is maintained.
0	PWRGOOD_ DET	**POWERGOOD Reset Detect** This bit is set when a reset from the PWRGOOD pin is detected. Software clears this bit by writing a 1. 0 = No PWRGOOD pin reset was detected. 1 = The system reset event was from the PWRGOOD pin. This reset event has higher priority over PRGRESET and disables the PRGRESET function (if it is enabled) by clearing the PRG_RST_ENB bit in the RESCFG register (see page 3-3).

Programming Notes

Unlike most other registers, the bits in this register are only returned to their reset value by a PWRGOOD reset. They are not cleared by any other kind of reset.

The AMDebug technology trace information is preserved only if a soft reset is generated to the CPU.

See the *Élan™SC520 Microcontroller User’s Manual*, order #22004, for details about reset generation. Table 3-3 provides a summary of ÉlanSC520 microcontroller reset sources and effects.

Table 3-3 Microcontroller Reset Sources

Source	CPU (Hard/Soft)	GPRESET Pin	$\overline{\text{RST}}$ Pin (PCI)	Internal Registers	Notes
PWRGOOD pin	Hard	✔	✔	✔	
PRGRESET pin	Hard	✔	✔	✔	1,2
SYS_RST bit, RESCFG register (see page 3-4)	Hard	✔	✔	✔	2
Watchdog timer reset event (page 16-2)	Hard	✔	✔	✔	2
AMDebug technology system reset	Hard	✔	✔	✔	2
CPU_RST bit, SYSCTLA register (page 3-9)	Soft				3
SCP soft reset, SCPCMD register (page 3-7)	Soft				
CPU shutdown (typically caused by a triple-fault)	Soft				
GP_RST bit, RESCFG register (page 3-3)		✔			
PCI_RST bit, HBCTL register (page 6-3)			✔		

Notes:
1. The PRG_RST_ENB bit must be set in the RESCFG register (see page 3-3) to enable the reset function on this pin.
2. If the PRG_RST_ENB bit is set, the SDRAM controller configuration is maintained to support system reset in which SDRAM contents are also maintained.
3. Any write of a 1 to the CPU_RST bit causes a soft reset, regardless of whether the bit was previously 1 or 0.

SCP Data Port (SCPDATA)

Direct-Mapped I/O Address 0060h

	7	6	5	4	3	2	1	0
Bit	SCP_DATA							
	Reserved						A20_GATE	CPU_RST
Reset	0	0	0	0	0	0	1	0
R/W	W!							

Register Description

This register is used to emulate system control processor (SCP) a20 gate control.

Bit Definitions

Bit	Name	Function
7–0	SCP_DATA	**System Control Processor Data** All reads and writes to this port are echoed to the GP bus.
1	A20_GATE	**A20 Gate Data** The ÉlanSC520 microcontroller has no external input pin for the a20 gate signal that would normally be driven by an external PC/AT-compatible system control processor (SCP). In order to maintain software compatibility, internal logic is provided to watch the GP bus for SCP a20 gate pin control command sequences. Following a write of D1h to the SCPCMD register (see page 3-8), a write to this bit (A20_GATE) has the following effect: 0 = Internal a20 propagation is disabled (address space wraps at 1 Mbyte). 1 = Internal a20 propagation is enabled (addresses above 1 Mbyte can be accessed). This bit (A20_GATE) provides one of two sources of a20 gate control. The other source is the A20G_CTL bit in the SYSCTLA register (see page 3-9). A logical OR of these two sources is used to drive the $Am5_x86$ CPU a20m signal. Therefore, a20 propagates if either source enables a20 to propagate. Note that this bit (A20_GATE) defaults to enabling a20 propagation.
0	CPU_RST	**CPU Reset Control** On the ÉlanSC520 microcontroller, writing to this bit has no effect. On a PC/AT-compatible system, setting this bit to 1 after a write of D1h to the SCP Command Port (SCPCMD) register would hold the system in reset indefinitely.

Programming Notes

There is no internal storage element associated with this address. All accesses to this address go to the GP bus. Additionally, writes to this address are snooped by the ÉlanSC520 microcontroller.

SCP Command Port (SCPCMD)

Direct-Mapped I/O Address 0064h

	7	6	5	4	3	2	1	0
Bit	SCP_CMD							
Reset	0	0	0	0	0	0	0	0
R/W	W!							

Register Description

This register is used to emulate system control processor (SCP) a20 gate and CPU Reset commands.

Bit Definitions

Bit	Name	Function
7–0	SCP_CMD	**SCP Command** The ÉlanSC520 microcontroller has no external input pins for the a20 gate and CPU reset signals that are normally driven by an external PC/AT-compatible system control processor (SCP). In order to maintain software compatibility, internal logic is provided to watch this port for SCP command sequences. D1h = To control the a20 signal, write the value D1h to this port, then write to the A20_GATE bit of the SCPDATA register (see page 3-7). FEh = To reset the CPU, write the value FEh to this port. This pulses the internal CPU sreset signal, generating a CPU soft reset. The Am5$_x$86 CPU cache state and the ÉlanSC520 microcontroller MMCR, indexed, and direct-mapped registers are not affected by this soft reset, with the exception that the NMI_ENB bit in the PICICR register is cleared (see page 12-4). Clearing the NMI_ENB bit allows software to initialize the stack pointer before setting the NMI_ENB bit again after a soft reset. Following this reset, the SCP_RST_DET bit in the RESSTA register is set to indicate the source of this reset (see page 3-5).

Programming Notes

There is no internal storage element associated with this address. All accesses to this address go to the GP bus. Additionally, writes to this address are snooped by the ÉlanSC520 microcontroller.

System Control Port A (SYSCTLA)

Direct-Mapped I/O Address 0092h

	7	6	5	4	3	2	1	0
Bit	Reserved						A20G_CTL	CPU_RST
Reset	0	0	0	0	0	0	0	0
R/W	RSV						R/W	R/W

Register Description

This register is used for fast, alternative control of the CPU soft reset and the a20 signal.

Bit Definitions

Bit	Name	Function
7–2	Reserved	**Reserved** This bit field should be written to 0 for normal system operation.
1	A20G_CTL	**A20 Gate Control** This bit can be used to cause the same type of masking of the CPU a20 signal that was historically performed by an external system control processor (SCP) in a PC/AT-compatible system, but much faster: 0 = Internal a20 propagation is disabled (address space wraps at 1 Mbyte). 1 = Internal a20 propagation is enabled (addresses above 1 Mbyte can be accessed). This bit (A20G_CTL) provides one of two sources of a20 gate control. The other source is the A20_GATE bit in the SCPDATA register (see page 3-7). A logical OR of these two sources is used to drive the $Am5_x86$ CPU a20m signal. Therefore, a20 propagates if either source enables a20 to propagate. Note that this bit (A20G_CTL) defaults to disabling a20 propagation. However, because the default state of the A20_GATE bit enables a20 propagation, a20 propagates by default.
0	CPU_RST	**Alternate CPU Core Reset Control** Writing a 1 to this bit pulses the internal CPU sreset signal. This causes the same type of CPU soft reset to occur that was historically performed by an external system control processor (SCP) in a PC/AT-compatible system, but much faster. The $Am5_x86$ CPU cache state and ÉlanSC520 microcontroller MMCR, indexed, and direct-mapped registers are not affected by this soft reset, with the exception that the NMI_ENB bit in the PICICR register is cleared (see page 12-4). Clearing the NMI_ENB bit allows software to initialize the stack pointer before setting the NMI_ENB bit again after a soft reset. Following the reset, this bit (CPU_RST) remains set until software clears it. This feature can be used by the BIOS as an indication that this bit generated the reset. However, writing a 1 to the CPU_RST bit always generates a soft reset, even if the bit was not cleared after a previous reset. 0 = Do not generate a soft reset to the CPU core. 1 = Pulse the internal CPU sreset signal.

Programming Notes

Using a read-modify-write operation to change the A20G_CTL bit can cause an unintended soft reset. This happens if the CPU_RST bit has been previously set and not cleared by software. Always write 0 to the CPU_RST bit unless a soft reset is desired.

CHAPTER

4 Am5$_x$86® CPU REGISTERS

4.1 OVERVIEW

This chapter describes the Am5$_x$86® CPU configuration registers of the ÉlanSC520 microcontroller.

Am5$_x$86 CPU includes a 16-Kbyte unified write-back cache and an integrated floating-point unit (FPU). The Am5$_x$86 CPU operates at 100 or 133 MHz.

The Am5$_x$86 CPU configuration register set consists of two memory-mapped configuration region (MMCR) registers used to control some of the Am5$_x$86 CPU features and read the ÉlanSC520 microcontroller revision ID. See the *Élan™SC520 Microcontroller User's Manual*, order #22004, for details about the Am5$_x$86 CPU.

Table 4-1 lists the Am5$_x$86 CPU configuration registers in offset order, with the corresponding description's page number.

This chapter does not document Am5$_x$86 processor registers.

4.2 REGISTERS

Table 4-1 Am5$_x$86® CPU MMCR Registers

Register Name	Mnemonic	MMCR Offset	Page Number
ÉlanSC520 Microcontroller Revision ID	REVID	00h	page 4-2
Am5$_x$86® CPU Control	CPUCTL	02h	page 4-3

Élan™SC520 Microcontroller Revision ID (REVID)

Memory-Mapped
MMCR Offset 00h

	15	14	13	12	11	10	9	8
Bit	PRODUCT_ID[7–0]							
Reset	0	0	0	0	0	0	0	0
R/W	R							

	7	6	5	4	3	2	1	0
Bit	MAJORSTEP[3–0]				MINORSTEP[3–0]			
Reset	x	x	x	x	x	x	x	x
R/W	R				R			

Register Description

This register provides bit fields that identify the AMD microcontroller and its version. These values are hard-wired in the device. Different revisions of a product can be distinguished by the MAJORSTEP and MINORSTEP bit fields.

Bit Definitions

Bit	Name	Function
15–8	PRODUCT_ID [7–0]	**Product Type of Élan™SC520 Microcontroller** 00000000 = Identifies the product as the ÉlanSC520 microcontroller.
7–4	MAJORSTEP [3–0]	**Major Stepping Level** This bit field represents the microcontroller's major revision level. A different MAJORSTEP bit field value is assigned to each major revision of the microcontroller.
3–0	MINORSTEP [3–0]	**Minor Stepping Level** This bit field represents the microcontroller's minor revision level. A larger MINORSTEP bit field value indicates a later revision within a particular MAJORSTEP bit field level.

Programming Notes

The value that is read back depends on the current major and minor revision level of the specific ÉlanSC520 microcontroller. Contact your AMD representative for information about available revision levels.

Am5$_x$86® CPU Control (CPUCTL)

Memory-Mapped
MMCR Offset 02h

	7	6	5	4	3	2	1	0
Bit	Reserved			CACHE_ WR_MODE	Reserved		CPU_CLK_SPD[1–0]	
Reset	0	0	0	0	0	0	0	1
R/W	RSV			R/W	RSV		R/W	

Register Description

This register controls the Am5$_x$86 CPU's cache write-back mode and clock speed.

Bit Definitions

Bit	Name	Function
7–5	Reserved	**Reserved** This bit field should be written to 0 for normal system operation.
4	CACHE_WR_ MODE	**Cache Write Mode** This bit enables the write-back mode of the Am5$_x$86 CPU cache. 0 = Write-back mode (default). 1 =Write-through mode. If this bit is set, the cache operates in write-through mode regardless of the setting of the NW bit in the processor's internal CR0 register, or the PWT bit of the corresponding page table entry.
3–2	Reserved	**Reserved** This bit field should be written to 0 for normal system operation.
1–0	CPU_CLK_SPD [1–0]	**CPU Clock Speed** This bit field specifies the clock speed at which the Am5$_x$86 CPU runs. Note that there is a 1 ms delay for the Am5$_x$86 CPU PLL to switch to the new frequency after this bit field is changed. 00 = Reserved 01 = 100 MHz (default) 10 = 133 MHz 11 = Reserved

Programming Notes

The Am5$_x$86 CPU core is reset during a hard reset, and the Am5$_x$86 CPU core clock frequency defaults to 100 MHz.

The Am5$_x$86 CPU core clock frequency remains the same and the cache state and policy are unaffected by a soft reset.

See the *Élan™SC520 Microcontroller User's Manual*, order #22004, for details about hard and soft resets. Table 3-3 on page 3-6 provides a summary of ÉlanSC520 microcontroller reset sources and effects.

CHAPTER

5 SYSTEM ARBITRATION REGISTERS

5.1 OVERVIEW

This chapter describes the system arbiter registers of the ÉlanSC520 microcontroller.

The ÉlanSC520 microcontroller's system arbiter controls access to the PCI bus and the internal CPU bus.

The system arbiter register set consists of four memory-mapped configuration region (MMCR) registers that provide control and status functions. See the *Élan™SC520 Microcontroller User's Manual*, order #22004, for details about the system arbiter.

Table 5-1 lists the system arbiter registers in offset order, with the corresponding description's page number.

5.2 REGISTERS

Table 5-1 System Arbiter MMCR Registers

Register Name	Mnemonic	MMCR Offset	Page Number
System Arbiter Control	SYSARBCTL	70h	page 5-2
PCI Bus Arbiter Status	PCIARBSTA	71h	page 5-3
System Arbiter Master Enable	SYSARBMENB	72h	page 5-4
Arbiter Priority Control	ARBPRICTL	74h	page 5-6

System Arbiter Control (SYSARBCTL)

Memory-Mapped
MMCR Offset 70h

	7	6	5	4	3	2	1	0
Bit	Reserved					BUS_ PARK_SEL	CNCR_ MODE_ ENB	GNT_TO_ INT_ENB
Reset	0	0	0	0	0	0	0	0
R/W	RSV					R/W	R/W	R/W

Register Description

This register contains control bits for the CPU bus arbiter and the PCI bus arbiter.

Bit Definitions

Bit	Name	Function
7–3	Reserved	**Reserved** This bit field should be written to 0 for normal system operation.
2	BUS_PARK_ SEL	**PCI Bus Arbiter Bus Park** This bit controls which PCI master the PCI bus arbiter parks on when the PCI bus is idle. 0 = Park the PCI bus on the $Am5_x86$ CPU. 1 = Park the PCI bus on the last PCI master that acquired the bus. This bit must be 0 when operating in nonconcurrent mode. The BUS_PARK_SEL bit must not be changed except when the PCI bus is currently parked on the $Am5_x86$ CPU. This is the default state after a system reset. See the *Élan™SC520 Microcontroller User's Manual*, order #22004, for information about system arbiter initialization.
1	CNCR_MODE_ ENB	**System Arbiter Concurrent Mode Enable** This bit enables the system arbiter to operate in concurrent mode. When operating in nonconcurrent mode, the BUS_PARK_SEL bit must be configured to park on the $Am5_x86$ CPU. 0 = The system arbiter operates in nonconcurrent mode. 1 = The system arbiter operates in concurrent mode. The CNCR_MODE_ENB bit must not be changed except during system arbiter initialization after a system reset. See the *Élan™SC520 Microcontroller User's Manual*, order #22004, for information about system arbiter initialization.
0	GNT_TO_INT_ ENB	**PCI Bus Arbiter Grant Time-Out Interrupt Enable** This bit is used to enable interrupts that are generated when the PCI bus arbiter detects a grant time-out. 0 = Disable PCI bus arbiter interrupts. 1 = Enable PCI bus arbiter interrupts. Note that the GNT_TO_STA bit of the PCIARBSTA register (see page 5-3) is set on PCI bus arbiter grant time-outs regardless of the GNT_TO_INT_ENB bit value. This interrupt source shares the interrupt controller input used by any host bridge interrupts enabled in the HBTGTIRQCTL and HBMSTIRQCTL registers (see page 6-5 and page 6-9). Before the GNT_TO_INT_ENB bit is set, the PCIHOSTMAP register (see page 12-17) must be configured to route the interrupt to the appropriate interrupt request level and priority.

Programming Notes

PCI Bus Arbiter Status (PCIARBSTA)

Memory-Mapped
MMCR Offset 71h

	7	6	5	4	3	2	1	0
Bit	GNT_TO_STA	Reserved			GNT_TO_ID[3–0]			
Reset	0	0	0	0	1	1	1	1
R/W	R/W!	RSV			R			

Register Description

This register provides grant time-out status of the PCI bus arbiter.

Bit Definitions

Bit	Name	Function
7	GNT_TO_STA	**PCI Bus Arbiter Grant Time-Out Status** This bit is set when the PCI bus arbiter detects a grant time-out (transaction not started within 16 clocks of the PCI bus going idle). The $\overline{GNT}$ that was asserted when this condition occurs can be read in the GNT_TO_ID bit field. 0 = Grant time-out has not occurred. 1 = Grant time-out has occurred. This bit (GNT_TO_STA) is cleared by writing a 1. This bit operates regardless of the corresponding interrupt enable bit (GNT_TO_INT_ENB in the SYSARBCTL register, see page 5-2). When enabled as an interrupt, this source shares the interrupt controller input used by any host bridge interrupts enabled in the HBTGTIRQCTL and HBMSTIRQCTL registers (see page 6-5 and page 6-9).
6–4	Reserved	**Reserved** This bit field should be written to 0 for normal system operation.
3–0	GNT_TO_ID [3–0]	**PCI Bus Arbiter Grant Time-Out Identification** This bit field identifies which $\overline{GNT}$ was asserted when the PCI arbiter detected a grant time-out. This bit field is reset to ones by writing a one to the GNT_TO_STA bit. 0000 = $\overline{GNT0}$ was asserted when a grant time-out was detected. 0001 = $\overline{GNT1}$ was asserted when a grant time-out was detected. 0010 = $\overline{GNT2}$ was asserted when a grant time-out was detected. 0011 = $\overline{GNT3}$ was asserted when a grant time-out was detected. 0100 = $\overline{GNT4}$ was asserted when a grant time-out was detected. 0101–1101 = Reserved 1110 = The $Am5_x86$ CPU $\overline{GNT}$ was asserted when a grant time-out was detected. 1111 = No grant time-out was detected, or $\overline{GNT}$ was asserted when a grant time-out was detected but not latched.

Programming Notes

System Arbiter Master Enable (SYSARBMENB)

Memory-Mapped
MMCR Offset 72h

	15	14	13	12	11	10	9	8
Bit	Reserved							
Reset	0	0	0	0	0	0	0	0
R/W	RSV							

	7	6	5	4	3	2	1	0
Bit	Reserved			REQ4_ENB	REQ3_ENB	REQ2_ENB	REQ1_ENB	REQ0_ENB
Reset	0	0	0	0	0	0	0	0
R/W	RSV			R/W	R/W	R/W	R/W	R/W

Register Description

This register selects the masters that are enabled for arbitration.

Bit Definitions

Bit	Name	Function
15–5	Reserved	**Reserved** This bit field should be written to 0 for normal system operation.
4	REQ4_ENB	**PCI Bus Arbiter Request #4 Enable** This bit enables the PCI master request connected to the $\overline{\text{REQ4}}$ pin. If this request is disabled, the PCI bus arbiter does not assert $\overline{\text{GNT4}}$ to allow the PCI master connected to $\overline{\text{REQ4}}$ and $\overline{\text{GNT4}}$ to access the PCI bus. 0 = The PCI master request connected to the $\overline{\text{REQ4}}$ pin is disabled. 1 = The PCI master request connected to the $\overline{\text{REQ4}}$ pin is enabled.
3	REQ3_ENB	**PCI Bus Arbiter Request #3 Enable** This bit enables the PCI master request connected to the $\overline{\text{REQ3}}$ pin. If this request is disabled, the PCI bus arbiter does not assert $\overline{\text{GNT3}}$ to allow the PCI master connected to the $\overline{\text{REQ3}}$ and $\overline{\text{GNT3}}$ pins to access the PCI bus. 0 = The PCI master request connected to the $\overline{\text{REQ3}}$ pin is disabled. 1 = The PCI master request connected to the $\overline{\text{REQ3}}$ pin is enabled.
2	REQ2_ENB	**PCI Bus Arbiter Request #2 Enable** This bit enables the PCI master request connected to the $\overline{\text{REQ2}}$ pin. If this request is disabled, the PCI bus arbiter does not assert $\overline{\text{GNT2}}$ to allow the PCI master connected to the $\overline{\text{REQ2}}$ and $\overline{\text{GNT2}}$ pins to access the PCI bus. 0 = The PCI master request connected to the $\overline{\text{REQ2}}$ pin is disabled. 1 = The PCI master request connected to the $\overline{\text{REQ2}}$ pin is enabled.
1	REQ1_ENB	**PCI Bus Arbiter Request #1 Enable** This bit enables the PCI master request connected to the $\overline{\text{REQ1}}$ pin. If this request is disabled, the PCI bus arbiter does not assert $\overline{\text{GNT1}}$ to allow the PCI master connected to the $\overline{\text{REQ1}}$ and $\overline{\text{GNT1}}$ pins to access the PCI bus. 0 = The PCI master request connected to the $\overline{\text{REQ1}}$ pin is disabled. 1 = The PCI master request connected to the $\overline{\text{REQ1}}$ pin is enabled.

Bit	Name	Function
0	REQ0_ENB	**PCI Bus Arbiter Request #0 Enable** This bit enables the PCI master request connected to the $\overline{\text{REQ0}}$ pin. If this request is disabled, the PCI bus arbiter does not assert $\overline{\text{GNT0}}$ to allow the PCI master connected to the $\overline{\text{REQ0}}$ and $\overline{\text{GNT0}}$ pins to access the PCI bus. 0 = The PCI master request connected to the $\overline{\text{REQ0}}$ pin is disabled. 1 = The PCI master request connected to the $\overline{\text{REQ0}}$ pin is enabled.

Programming Notes

Arbiter Priority Control (ARBPRICTL)

Memory-Mapped MMCR Offset 74h

	31	30	29	28	27	26	25	24
Bit	CPU_PRI[1–0]		Reserved					
Reset	0	1	0	0	0	0	0	0
R/W	R/W		RSV					

	23	22	21	20	19	18	17	16
Bit	Reserved							
Reset	0	0	0	0	0	0	0	0
R/W	RSV							

	15	14	13	12	11	10	9	8
Bit	Reserved				HI_PRI_1_SEL[3–0]			
Reset	0	0	0	0	1	1	1	1
R/W	RSV				R/W	R/W	R/W	R/W

	7	6	5	4	3	2	1	0
Bit	Reserved				HI_PRI_0_SEL[3–0]			
Reset	0	0	0	0	1	1	1	1
R/W	RSV				R/W	R/W	R/W	R/W

Register Description

This register defines priorities for the PCI bus arbiter.

Bit Definitions

Bit	Name	Function
31–30	CPU_PRI[1–0]	**PCI Bus Arbiter CPU Priority** This bit field defines the relative Am5$_x$86 CPU PCI master priority. 00 = Reserved. 01 = The Am5$_x$86 CPU is granted the PCI Bus after every one external PCI master cycle. 10 = The Am5$_x$86 CPU is granted the PCI Bus after every two external PCI master cycles. 11 = The Am5$_x$86 CPU is granted the PCI Bus after every three external PCI master cycles.
29–12	Reserved	**Reserved** This bit field should be written to 0 for normal system operation.

Bit	Name	Function
11–8	HI_PRI_1_SEL [3–0]	**PCI Bus Arbiter High Priority 1** This bit defines which PCI master is in position 1 of the high-priority queue. 0000 = PCI master connected to $\overline{\text{REQ0}}$ and $\overline{\text{GNT0}}$ is in position 1 of the high-priority queue. 0001 = PCI master connected to $\overline{\text{REQ1}}$ and $\overline{\text{GNT1}}$ is in position 1 of the high-priority queue. 0010 = PCI master connected to $\overline{\text{REQ2}}$ and $\overline{\text{GNT2}}$ is in position 1 of the high-priority queue. 0011 = PCI master connected to $\overline{\text{REQ3}}$ and $\overline{\text{GNT3}}$ is in position 1 of the high-priority queue. 0100 = PCI master connected to $\overline{\text{REQ4}}$ and $\overline{\text{GNT4}}$ is in position 1 of the high-priority queue. 0101–1110 = Reserved. 1111 = No master is in position 1 of the high-priority queue.
7–4	Reserved	**Reserved** This bit field should be written to 0 for normal system operation.
3–0	HI_PRI_0_SEL [3–0]	**PCI Bus Arbiter High Priority 0** This bit defines which PCI master is in position 0 of the high-priority queue. 0000 = PCI master connected to $\overline{\text{REQ0}}$ and $\overline{\text{GNT0}}$ is in position 0 of the high-priority queue. 0001 = PCI master connected to $\overline{\text{REQ1}}$ and $\overline{\text{GNT1}}$ is in position 0 of the high-priority queue. 0010 = PCI master connected to $\overline{\text{REQ2}}$ and $\overline{\text{GNT2}}$ is in position 0 of the high-priority queue. 0011 = PCI master connected to $\overline{\text{REQ3}}$ and $\overline{\text{GNT3}}$ is in position 0 of the high-priority queue. 0100 = PCI master connected to $\overline{\text{REQ4}}$ and $\overline{\text{GNT4}}$ is in position 0 of the high-priority queue. 0101–1110 = Reserved. 1111 = No master is in position 0 of the high-priority queue.

Programming Notes

CHAPTER

6 PCI BUS HOST BRIDGE REGISTERS

6.1 OVERVIEW

This chapter describes the PCI bus host bridge controller registers of the ÉlanSC520 microcontroller.

The ÉlanSC520 microcontroller includes an integrated PCI bus host bridge controller. The host bridge allows the $Am5_x86$ CPU to generate PCI bus master cycles and allows PCI bus masters to access the ÉlanSC520 microcontroller's SDRAM.

The host bridge register set includes three groups of registers:

- Six memory-mapped configuration region (MMCR) registers are used to configure and control most functions specific to the ÉlanSC520 microcontroller host bridge.
- Two direct-mapped I/O addresses are used to access the PCI bus configuration space for both the host bridge and for any other devices present on the PCI bus.
- Five host bridge-specific PCI bus indexed registers in the PCI bus configuration space provide the mandatory header registers that are required for any PCI bus device, plus the Master Retry Time-Out register.

In addition to the host bridge, the PCI bus can contain several other devices, each with its own PCI configuration space that is also accessed via the two PCI direct-mapped registers.

Access from other PCI bus masters to the host bridge registers is not supported. The host bridge implements only those configuration registers that are related to the host bridge functions.

See the *Élan™SC520 Microcontroller User's Manual*, order #22004, for details about the host bridge controller.

Table 6-1, Table 6-2 on page 6-2, and Table 6-3 on page 6-2 list each type of host bridge controller register in offset order with the corresponding description's page number.

6.2 REGISTERS

Table 6-1 PCI Bus Host Bridge MMCR Registers

Register Name	Mnemonic	MMCR Offset	Page Number
Host Bridge Control	HBCTL	60h	page 6-3
Host Bridge Target Interrupt Control	HBTGTIRQCTL	62h	page 6-5
Host Bridge Target Interrupt Status	HBTGTIRQSTA	64h	page 6-7
Host Bridge Master Interrupt Control	HBMSTIRQCTL	66h	page 6-9
Host Bridge Master Interrupt Status	HBMSTIRQSTA	68h	page 6-12
Host Bridge Master Interrupt Address	MSTINTADD	6Ch	page 6-14

Table 6-2 PCI Bus Host Bridge Direct-Mapped Registers

Register Name	Mnemonic	I/O Address	Page Number
PCI Configuration Address	PCICFGADR	0CF8h	page 6-15
PCI Configuration Data	PCICFGDATA	0CFCh	page 6-17

Table 6-3 PCI Bus Host Bridge Indexed Registers

Register Name	Mnemonic	I/O Address	PCI Index	Page Number
Device/Vendor ID	PCIDEVID	0CF8h/0CFCh	00h	page 6-18
Status/Command	PCISTACMD	0CF8h/0CFCh	04h	page 6-19
Class Code/Revision ID	PCICCREVID	0CF8h/0CFCh	08h	page 6-22
Header Type	PCIHEADTYPE	0CF8h/0CFCh	0Eh	page 6-23
Master Retry Time-Out	PCIMRETRYTO	0CF8h/0CFCh	41h	page 6-24

Host Bridge Control (HBCTL)

Memory-Mapped
MMCR Offset 60h

	15	14	13	12	11	10	9	8
Bit	PCI_RST	Reserved				T_PURGE_RD_ENB	T_DLYTR_ENB[1–0]	
Reset	0	0	0	0	0	0	0	0
R/W	R/W!	RSV				R/W	R/W	

	7	6	5	4	3	2	1	0
Bit	Reserved				M_WPOST_ENB	Reserved		
Reset	0	0	0	0	0	0	0	0
R/W	RSV				R/W	RSV		

Register Description

This register contains bit fields for configuring the host bridge controller.

Bit Definitions

Bit	Name	Function
15	PCI_RST	**PCI Bus Reset** This bit controls the PCI bus $\overline{\text{RST}}$ signal pin. Reading this bit returns the value that was written to it. 0 = Deassert the PCI bus reset signal. 1 = Assert the PCI bus reset signal. Note that a PCI bus reset affects the host bridge-specific registers in the PCI configuration space. See the PCI-indexed register descriptions beginning on page 6-18. The PCI_RST bit must be cleared only in accordance with the PCI bus specification.
14–11	Reserved	**Reserved** This bit field should be written to 0 for normal system operation.
10	T_PURGE_RD_ENB	**Target FIFO Purge Enable** This bit is provided for data coherency. It forces the host bridge target controller to snoop the target read FIFOs when a PCI bus write transaction occurs. 0 = The target read FIFOs are not snooped during write transactions. 1 = The target read FIFOs are snooped during PCI bus write transactions. The target controller purges any data in the read FIFOs when the write transaction falls within the same cache line for a read or read-line command, or within the same 64 doublewords for a read-multiple command. The T_PURGE_RD_ENB bit must not be changed except during PCI bus initialization after a system reset. See the *Élan™SC520 Microcontroller User's Manual*, order #22004, for information about PCI bus initialization.

Bit	Name	Function
9–8	T_DLYTR_ENB [1–0]	**Automatic Delayed Transaction Enable** This bit forces all PCI bus master reads of microcontroller SDRAM to be treated as delayed transactions unless the host bridge has already completed the transaction from a previous request. Setting this bit allows unused PCI bus bandwidth to be used by other PCI bus masters while the read transaction from the first master is being serviced. 00 = PCI bus reads to SDRAM are not automatically retried as delayed transactions. Instead, the originating PCI bus master is held in wait states while the host bridge begins the SDRAM read transaction. If 32 PCI bus clock cycles pass before the first doubleword has been read from SDRAM by the host bridge target controller, the bridge then issues a retry (as required for PCI compliance). 01 = All PCI bus reads to SDRAM are automatically retried as delayed transactions unless the host bridge has already completed the transaction from a previous request. This occurs when a PCI bus master has already issued the transaction previously, and was issued a retry by the host bridge target controller. 10 = Reserved 11 = Reserved
7–4	Reserved	**Reserved** This bit field should be written to 0 for normal system operation.
3	M_WPOST_ ENB	**Master Controller Write Posting Enable** This bit enables Am5$_x$86 CPU-to-PCI bus memory write cycles to be posted writes. 0 = Disables Am5$_x$86 CPU-to-PCI bus write posting. 1 = Enables Am5$_x$86 CPU-to-PCI bus write posting. Note that this bit should not be set while the microcontroller is configured for non-concurrent arbitration mode (i.e., while the CNCR_MODE_ENB bit is clear in the SYSARBCTL register, see page page 5-2).
2–0	Reserved	**Reserved** This bit field should be written to 0 for normal system operation.

Programming Notes

This register is reset by a system reset. The bits in this register are not affected by a PCI bus reset. A PCI bus reset is initiated by setting the PCI_RST bit.

Host Bridge Target Interrupt Control (HBTGTIRQCTL)

Memory-Mapped
MMCR Offset 62h

	15	14	13	12	11	10	9	8
Bit	Reserved					T_DLYTO_ IRQ_SEL	T_APER_ IRQ_SEL	T_DPER_ IRQ_SEL
Reset	0	0	0	0	0	0	0	0
R/W	RSV					R/W	R/W	R/W

	7	6	5	4	3	2	1	0
Bit	Reserved					T_DLYTO_ IRQ_ENB	T_APER_ IRQ_ENB	T_DPER_ IRQ_ENB
Reset	0	0	0	0	0	0	0	0
R/W	RSV					R/W	R/W	R/W

Register Description

This register contains bit fields for configuring and enabling host bridge target interrupts.

Bit Definitions

Bit	Name	Function
15–11	Reserved	**Reserved** This bit field should be written to 0 for normal system operation.
10	T_DLYTO_ IRQ_SEL	**Target Delayed Transaction Time-Out Interrupt Select** This bit allows delayed transaction time-outs to generate an NMI instead of a maskable interrupt. 0 = Delayed transaction time-outs generate a maskable interrupt. 1 = Delayed transaction time-outs generate an NMI.
9	T_APER_ IRQ_SEL	**Target Address Parity Interrupt Select** This bit allows address parity errors detected by the target controller to generate an NMI instead of a maskable interrupt. 0 = Address parity errors generate a maskable interrupt. 1 = Address parity errors generate an NMI.
8	T_DPER_ IRQ_SEL	**Target Data Parity Interrupt Select** This bit allows data parity errors detected by the target controller to generate an NMI instead of a maskable interrupt. 0 = Data parity errors generate a maskable interrupt. 1 = Data parity errors generate an NMI.
7–3	Reserved	**Reserved** This bit field should be written to 0 for normal system operation.
2	T_DLYTO_ IRQ_ENB	**Target Delayed Transaction Time-Out Interrupt Enable** This bit allows delayed transaction time-outs to generate an interrupt. 0 = Delayed transaction time-outs do not generate an interrupt. 1 = Delayed transaction time-outs generate an interrupt.
1	T_APER_ IRQ_ENB	**Target Address Parity Interrupt Enable** This bit allows address parity errors detected by the target controller to generate an interrupt. 0 = Address parity errors do not generate an interrupt. 1 = Address parity errors generate an interrupt.

Bit	Name	Function
0	T_DPER_ IRQ_ENB	**Target Data Parity Interrupt Enable** This bit allows data parity errors detected by the target controller to generate an interrupt. 0 = Data parity errors do not generate an interrupt. 1 = Data parity errors generate an interrupt.

Programming Notes

This register is reset by a system reset. The bits in this register are not affected by a PCI bus reset. A PCI bus reset is initiated by setting the PCI_RST bit in the HBCTL register (see page 6-3).

Interrupt status bits are set whenever the associated event occurs regardless of the corresponding interrupt enable bit.

For a host bridge NMI to propagate to the CPU, host bridge NMIs must be enabled via the PCI_NMI_ENB bit in the PCIHOSTMAP register (see page 12-17), and NMIs must be enabled via the NMI_ENB bit in the PICICR register (see page 12-4).

Before host bridge interrupts are enabled, the PCIHOSTMAP register (see page 12-17) must be configured to route the interrupt to the appropriate interrupt request level and priority.

The interrupt enabled via the GNT_TO_INT_ENB bit in the SYSARBCTL register (see page 5-2) shares the interrupt controller input used by host bridge interrupts.

Host Bridge Target Interrupt Status (HBTGTIRQSTA)

Memory-Mapped
MMCR Offset 64h

	15	14	13	12	11	10	9	8
Bit	Reserved				T_IRQ_ID[3–0]			
Reset	0	0	0	0	1	1	1	1
R/W	RSV				R			

	7	6	5	4	3	2	1	0
Bit	Reserved					T_DLYTO_ IRQ_STA	T_APER_ IRQ_STA	T_DPER_ IRQ_STA
Reset	0	0	0	0	0	0	0	0
R/W	RSV					R/W!	R/W!	R/W!

Register Description

This register contains the host bridge target interrupt status bits and active master identification.

Bit Definitions

Bit	Name	Function
15–12	Reserved	**Reserved** This bit field should be written to 0 for normal system operation.
11–8	T_IRQ_ID[3–0]	**Target Interrupt Identification** This bit field reports which PCI bus master was active when the target controller detected an error condition (delay transaction time-out, address parity error, data parity error). This bit field is only valid when an interrupt status bit is set. It is reset to 1111b when the interrupt status is cleared. 0000 = PCI bus master 0 was active when the error was detected. 0001 = PCI bus master 1 was active when the error was detected. 0010 = PCI bus master 2 was active when the error was detected. 0011 = PCI bus master 3 was active when the error was detected. 0100 = PCI bus master 4 was active when the error was detected. 1111 = No error was detected or the bus master value was not latched. For example, a bus master value is not latched if multiple interrupts are pending and one interrupt status was cleared. Other =Reserved. When multiple interrupts are pending, this bit field represents the PCI bus master active when the first interrupt occurred, but there is no way for software to detect which interrupt this is.
7–3	Reserved	**Reserved** This bit field should be written to 0 for normal system operation.
2	T_DLYTO_ IRQ_STA	**Target Delayed Transaction Time-Out Interrupt Status** This bit is set when the target controller issues a delayed transaction retry, the PCI bus master does not retry the transaction within 2^{15} clocks, and the target controller is idle (i.e., no PCI-to-SDRAM transaction is active on the PCI bus). 0 = Delayed transaction time-out has not occurred. 1 = Delayed transaction time-out has occurred. This bit is cleared by writing a 1. The target controller discards the delayed transaction request when this bit is set. This bit operates regardless of the corresponding interrupt enable bit (T_DLYTO_IRQ_ENB in the HBTGTIRQCTL register, see page 6-5).

Bit	Name	Function
1	T_APER_IRQ_STA	**Target Address Parity Interrupt Status** This bit is set when the target controller detects an address parity error. 0 = Address parity error has not occurred. 1 = Address parity error has occurred. This bit is cleared by writing a 1. This bit operates regardless of the corresponding interrupt enable bit (T_APER_IRQ_ENB in the HBTGTIRQCTL register, see page 6-5), or the Parity Error Response bit (PERR_RES in the PCISTACMD register, see page 6-21).
0	T_DPER_IRQ_STA	**Target Data Parity Interrupt Status** This bit is set when the target controller detects a data parity error. 0 = Data parity error has not occurred. 1 = Data parity error has occurred. This bit is cleared by writing a 1. This bit operates regardless of the corresponding interrupt enable bit (T_DPER_IRQ_ENB in the HBTGTIRQCTL register, see page 6-6), or the Parity Error Response bit (PERR_RES in the PCISTACMD register, see page 6-21).

Programming Notes

This register is reset by a system reset. The bits in this register are not affected by a PCI bus reset. A PCI bus reset is initiated by setting the PCI_RST bit in the HBCTL register (see page 6-3).

Host Bridge Master Interrupt Control (HBMSTIRQCTL)

Memory-Mapped
MMCR Offset 66h

	15	14	13	12	11	10	9	8
Bit	Reserved		M_RTRTO_ IRQ_SEL	M_TABRT_ IRQ_SEL	M_MABRT_ IRQ_SEL	M_SERR_ IRQ_SEL	M_RPER_ IRQ_SEL	M_DPER_ IRQ_SEL
Reset	0	0	0	0	0	0	0	0
R/W	RSV		R/W	R/W	R/W	R/W	R/W	R/W

	7	6	5	4	3	2	1	0
Bit	Reserved		M_RTRTO_ IRQ_ENB	M_TABRT_ IRQ_ENB	M_MABRT_ IRQ_ENB	M_SERR_ IRQ_ENB	M_RPER_ IRQ_ENB	M_DPER_ IRQ_ENB
Reset	0	0	0	0	0	0	0	0
R/W	RSV		R/W	R/W	R/W	R/W	R/W	R/W

Register Description

This register contains bit fields for configuring and enabling host bridge master interrupts.

Bit Definitions

Bit	Name	Function
15–14	Reserved	**Reserved** This bit field should be written to 0 for normal system operation.
13	M_RTRTO_ IRQ_SEL	**Master Retry Time-Out Interrupt Select** This bit allows master retry time-outs to generate an NMI instead of a maskable interrupt. 0 = Master retry time-outs generate a maskable interrupt. 1 = Master retry time-outs generate an NMI.
12	M_TABRT_ IRQ_SEL	**Master Target Abort Interrupt Select** This bit allows master controller transactions that are terminated with a target abort to generate an NMI instead of a maskable interrupt. 0 = Master transactions that are terminated with a master abort generate a maskable interrupt. 1 = Master transactions that are terminated with a master abort generate an NMI.
11	M_MABRT_ IRQ_SEL	**Master Abort Interrupt Select** This bit allows master controller transactions that are terminated with a master abort to generate an NMI instead of a maskable interrupt. 0 = Master transactions that are terminated with a master abort generate a maskable interrupt. 1 = Master transactions that are terminated with a master abort generate an NMI.
10	M_SERR_ IRQ_SEL	**Master System Error Interrupt Select** This bit allows the assertion of the system error signal ($\overline{\text{SERR}}$) to generate an NMI instead of a maskable interrupt. 0 = Assertion of the system error signal generates a maskable interrupt. 1 = Assertion of the system error signal generates an NMI.

Bit	Name	Function
9	M_RPER_ IRQ_SEL	**Master Received Parity Error Interrupt Select** This bit allows the assertion of the parity error signal ($\overline{\text{PERR}}$) during a master controller write transaction or during the address phase of a master controller read transaction to generate an NMI instead of a maskable interrupt. 0 = Master write transactions or master read address phase cycles that detect the parity error signal asserted generate a maskable interrupt. 1 = Master write transactions or master read address phase cycles that detect the parity error signal asserted generate an NMI.
8	M_DPER_ IRQ_SEL	**Master Detected Parity Error Interrupt Select** This bit allows parity errors detected by the master controller during a read transaction to generate an NMI instead of a maskable interrupt. 0 = Master read transactions that detect a parity error generate a maskable interrupt. 1 = Master read transactions that detect a parity error generate an NMI.
7–6	Reserved	**Reserved** This bit field should be written to 0 for normal system operation.
5	M_RTRTO_ IRQ_ENB	**Master Retry Time-Out Interrupt Enable** This bit allows master retry time-outs to generate an interrupt. 0 = Master retry time-outs do not generate an interrupt. 1 = Master retry time-outs generate an interrupt.
4	M_TABRT_ IRQ_ENB	**Master Target Abort Interrupt Enable** This bit allows master controller transactions that are terminated with a target abort to generate an interrupt. 0 = Master transactions that are terminated with a target abort do not generate an interrupt. 1 = Master transactions that are terminated with a target abort generate an interrupt.
3	M_MABRT_ IRQ_ENB	**Master Abort Interrupt Enable** This bit allows master controller transactions that are terminated with a master abort to generate an interrupt. 0 = Master transactions that are terminated with a master abort do not generate an interrupt. 1 = Master transactions that are terminated with a master abort generate an interrupt.
2	M_SERR_ IRQ_ENB	**Master System Error Interrupt Enable** This bit allows the assertion of the system error signal ($\overline{\text{SERR}}$) by a PCI bus agent to generate an interrupt. 0 = Assertion of the system error signal does not generate an interrupt. 1 = Assertion of the system error signal generates an interrupt.
1	M_RPER_ IRQ_ENB	**Master Received Parity Error Interrupt Enable** This bit allows the assertion of the parity error signal ($\overline{\text{PERR}}$) during a master controller write transaction or during the address phase of a master controller read transaction to generate an interrupt. 0 = Master write transactions or master read address phase cycles that detect the parity error signal asserted do not generate an interrupt. 1 = Master write transactions or master read address phase cycles that detect the parity error signal asserted generate an interrupt.
0	M_DPER_ IRQ_ENB	**Master Detected Parity Error Interrupt Enable** This bit allows parity errors detected by the master controller during a read transaction to generate an interrupt. 0 = Master read transactions that detect a parity error do not generate an interrupt. 1 = Master read transactions that detect a parity error generate an interrupt.

Programming Notes

This register is reset by a system reset. The bits in this register are not affected by a PCI bus reset. A PCI bus reset is initiated by setting the PCI_RST bit in the HBCTL register (see page 6-3).

Interrupt status bits are set whenever the associated event occurs regardless of the corresponding interrupt enable bit.

For a host bridge NMI to propagate to the CPU, host bridge NMIs must be enabled via the PCI_NMI_ENB bit in the PCIHOSTMAP register (see page 12-17), and NMIs must be enabled via the NMI_ENB bit in the PICICR register (see page 12-4).

Before host bridge interrupts are enabled, the PCIHOSTMAP register (see page 12-17) must be configured to route the interrupt to the appropriate interrupt request level and priority.

The interrupt enabled via the GNT_TO_INT_ENB bit in the SYSARBCTL register (see page 5-2) shares the interrupt controller input used by host bridge interrupts.

Host Bridge Master Interrupt Status (HBMSTIRQSTA)

Memory-Mapped
MMCR Offset 68h

	15	14	13	12	11	10	9	8
Bit	Reserved				M_CMD_IRQ_ID[3–0]			
Reset	0	0	0	0	0	0	0	0
R/W	RSV				R			

	7	6	5	4	3	2	1	0
Bit	Reserved		M_RTRTO_ IRQ_STA	M_TABRT_ IRQ_STA	M_MABRT_ IRQ_STA	M_SERR_ IRQ_STA	M_RPER_ IRQ_STA	M_DPER_ IRQ_STA
Reset	0	0	0	0	0	0	0	0
R/W	RSV		R/W!	R/W!	R/W!	R/W!	R/W!	R/W!

Register Description

This register contains host bridge master controller interrupt status bits and command interrupt identification.

Bit Definitions

Bit	Name	Function
15–12	Reserved	**Reserved** This bit field should be written to 0 for normal system operation.
11–8	M_CMD_ IRQ_ID[3–0]	**Master Command Interrupt Identification** This bit field reports the command of the transaction during which the master controller detected an error condition. This bit field is valid only when an interrupt status bit is set. It is cleared when the interrupt status is cleared. 0000 = The command was not latched. For example, the command is not latched if multiple interrupts are pending and one interrupt status was cleared. 0001 = Special Cycle (not used by the ÉlanSC520 microcontroller). 0010 = I/O Read. 0011 = I/O Write. 0100–0101 = Reserved. 0110 = Memory Read. 0111 = Memory Write. 1000–1001 = Reserved. 1010 = Configuration Read. 1011 = Configuration Write. 1100 = Memory Read Multiple (not used by the ÉlanSC520 microcontroller). 1101 = Dual-Address Cycle (not used by the ÉlanSC520 microcontroller). 1110 = Memory Read Line (not used by the ÉlanSC520 microcontroller). 1111 = Memory Write and Invalidate (not used by the ÉlanSC520 microcontroller). If multiple errors are detected, only the command of the first error is latched. When multiple error interrupts are pending, there is no indication of which interrupt the command corresponds to. The MSTINTADD register (see page 6-14) contains the address of the transaction during which the master controller detected an error condition.
7–6	Reserved	**Reserved** This bit field should be written to 0 for normal system operation.

Bit	Name	Function
5	M_RTRTO_IRQ_STA	**Master Retry Time-Out Interrupt Status** This bit is set when the master controller retry time-out counter expires. 0 = Master retry time-out has not occurred. 1 = Master retry time-out has occurred. This bit is cleared by writing a 1. This bit operates regardless of the corresponding interrupt enable bit (M_RTRTO_IRQ_ENB in the HBMSTIRQCTL register, see page 6-10).
4	M_TABRT_IRQ_STA	**Master Target Abort Interrupt Status** This bit is set when a master controller transaction is terminated with a target abort. 0 = Master controller transaction has not been terminated with a target abort. 1 = Master controller transaction has been terminated with a target abort. This bit is cleared by writing a 1. This bit operates regardless of the corresponding interrupt enable bit (M_TABRT_IRQ_ENB in the HBMSTIRQCTL register, see page 6-10).
3	M_MABRT_IRQ_STA	**Master Abort Interrupt Status** This bit is set when a master controller transaction is terminated with a master abort. 0 = Master controller transaction has not been terminated with a master abort. 1 = Master controller transaction has been terminated with a master abort. This bit is cleared by writing a 1. This bit operates regardless of the corresponding interrupt enable bit (M_MABRT_IRQ_ENB in the HBMSTIRQCTL register, see page 6-10).
2	M_SERR_IRQ_STA	**Master System Error Interrupt Status** This bit is set when the master controller detects the system error signal asserted. 0 = Master controller has not detected the system error signal asserted. 1 = Master controller has detected the system error signal asserted. This bit is cleared by writing a 1. This bit operates regardless of the corresponding interrupt enable bit (M_SERR_IRQ_ENB in the HBMSTIRQCTL register, see page 6-10).
1	M_RPER_IRQ_STA	**Master Received Parity Error Interrupt Status** This bit is set when the master controller detects the parity error signal asserted during a master controller write transaction or during the address phase of a master controller read transaction. 0 = Master controller has not detected the parity error signal asserted. 1 = Master controller has detected the parity error signal asserted. This bit is cleared by writing a 1. This bit operates regardless of the corresponding interrupt enable bit (M_RPER_IRQ_ENB in the HBMSTIRQCTL register, see page 6-10).
0	M_DPER_IRQ_STA	**Master Detected Parity Error Interrupt Status** This bit is set when the master controller detects a parity error during a master controller read transaction. 0 = Master controller has not detected a parity error. 1 = Master controller has detected a parity error. This bit is cleared by writing a 1. This bit operates regardless of the corresponding interrupt enable bit (M_DPER_IRQ_ENB in the HBMSTIRQCTL register, see page 6-10).

Programming Notes

This register is reset by a system reset. The bits in this register are not affected by a PCI bus reset. A PCI bus reset is initiated by setting the PCI_RST bit in the HBCTL register (see page 6-3).

Host Bridge Master Interrupt Address (MSTINTADD)

Memory-Mapped MMCR Offset 6Ch

	31	30	29	28	27	26	25	24
Bit	M_AD_IRQ_ID[31–24]							
Reset	0	0	0	0	0	0	0	0
R/W	R							

	23	22	21	20	19	18	17	16
Bit	M_AD_IRQ_ID[23–16]							
Reset	0	0	0	0	0	0	0	0
R/W	R							

	15	14	13	12	11	10	9	8
Bit	M_AD_IRQ_ID[15–8]							
Reset	0	0	0	0	0	0	0	0
R/W	R							

	7	6	5	4	3	2	1	0
Bit	M_AD_IRQ_ID[7–0]							
Reset	0	0	0	0	0	0	0	0
R/W	R							

Register Description

This bit field reports the address of the transaction during which the master controller detected an error condition.

Bit Definitions

Bit	Name	Function
31–0	M_AD_IRQ_ID [31–0]	**Master Address Interrupt Identification** This bit field reports the address of the transaction during which the master controller detected an error condition. If multiple errors are detected, only the address of the first error is latched. This bit field is only valid when an interrupt status bit is set. It is cleared when any interrupt status bit is cleared. A value of 00000000h means the address was not latched (unless the error happened to occur at address 00000000h). For example, the address is not latched if multiple interrupts are pending and one interrupt status was cleared. When multiple interrupts are pending, there is no indication of which interrupt the address corresponds to. The M_CMD_IRQ_ID bit field in the HBMSTIRQSTA register (see page 6-12) contains the command of the transaction during which the master controller detected an error condition.

Programming Notes

This register is reset by a system reset. The bits in this register are not affected by a PCI bus reset. A PCI bus reset is initiated by setting the PCI_RST bit in the HBCTL register (see page 6-3).

PCI Configuration Address (PCICFGADR)

Direct-Mapped I/O Address 0CF8h

	31	30	29	28	27	26	25	24
Bit	ENABLE	Reserved						
Reset	0	0	0	0	0	0	0	0
R/W	R/W!	RSV						

	23	22	21	20	19	18	17	16
Bit	BUS_NUM[7–0]							
Reset	0	0	0	0	0	0	0	0
R/W	R/W!							

	15	14	13	12	11	10	9	8
Bit	DEVICE_NUM[4–0]					FUNCTION_NUM[2–0]		
Reset	0	0	0	0	0	0	0	0
R/W	R/W!					R/W!		

	7	6	5	4	3	2	1	0
Bit	REGISTER_NUM[5–0]						Reserved	
Reset	0	0	0	0	0	0	0	0
R/W	R/W!						RSV!	

Register Description

This register is used to specify the information to be driven out during the address phase of the next configuration cycle, which is initiated by a subsequent read or write of the PCICFGDATA register (see page 6-17).

Bit Definitions

Bit	Name	Function
31	ENABLE	**Enable** 0 = Accesses to the PCICFGDATA register (see page 6-17) are not converted to configuration cycles on the PCI bus. They remain normal I/O cycles. 1 = Accesses to the PCICFGDATA register are converted to configuration cycles on the PCI bus.
30–24	Reserved	**Reserved** This bit field should be written to 0 for normal system operation.
23–16	BUS_NUM[7–0]	**Bus Number** This bit field specifies the PCI bus number to which the configuration cycle is addressed. 00h = Specifies the PCI bus that is connected to the ÉlanSC520 microcontroller's host bridge. This causes the host bridge to perform a type zero configuration cycle. Other values = The host bridge performs a type one configuration cycle. In a type one configuration cycle, the contents of this bit field (BUS_NUM) are driven unchanged on the PCI bus during the address phase of the cycle. If the BUS_NUM and DEVICE_NUM bit fields are both 0, the configuration cycle does not appear on the PCI bus because this combination addresses the internal PCI-indexed registers of the host bridge.

Bit	Name	Function
15–11	DEVICE_NUM [4–0]	**Device Number** This bit field specifies the device to address on the bus that is specified by the BUS_NUM bit field. For type zero configuration cycles (if the BUS_NUM bit field is 00h), the DEVICE_NUM bit field value selects one of the following bit patterns to be driven on the PCI bus AD31–AD11 pins during the address phase of the cycle: 00d = Address the ÉlanSC520 microcontroller's host bridge (if the BUS_NUM bit field is 00h). The cycle is not visible on the PCI bus. (I.e., no bit pattern is driven externally.) 01d = The AD12 pin is driven High, and the other pins of AD31–AD11 are driven Low. 02d = The AD13 pin is driven High, and the other pins of AD31–AD11 are driven Low. 03d = The AD14 pin is driven High, and the other pins of AD31–AD11 are driven Low. . . . . . 19d = The AD30 pin is driven High, and the other pins of AD31–AD11 are driven Low. 20d = The AD31 pin is driven High, and the other pins of AD31–AD11 are driven Low. 21–31d = All pins AD31–AD11 are driven Low (to 0); but the host bridge does not accept configuration accesses using these DEVICE_NUM bit field values, so configuration reads or writes with these values result in a Master Abort. Performing a configuration read with these values returns FFFFh in the data. In a typical system design, one of the address pins AD31–AD12 is resistively coupled to each PCI bus device's IDSEL input, so a DEVICE_NUM bit field value in the range 1–20d selects the corresponding PCI bus device during a type zero configuration cycle. For type one configuration cycles (if the BUS_NUM bit field is not 00h), the contents of this bit field are driven unchanged on the PCI bus during the address phase of the cycle.
10–8	FUNCTION_ NUM[2–0]	**Function Number** This bit field specifies the function number within the device specified by the DEVICE_NUM bit field. For host bridge configuration cycles (if the BUS_NUM and DEVICE_NUM bit fields are both 0), the function number is ignored because the host bridge is a single-function device. For all other configuration cycles (type zero or type one), the contents of this bit field are driven unchanged on the PCI bus during the address phase of the configuration cycle.
7–2	REGISTER_ NUM[5–0]	**Register Number** This bit field specifies the register number within the function specified by the FUNCTION_NUM bit field. For host bridge configuration cycles (if the BUS_NUM and DEVICE_NUM bit fields are both 0), the REGISTER_NUM bit field is used to address the host bridge PCI-indexed registers (see the descriptions beginning on page 6-18). REGISTER_NUM bits 5–0 are used as bits 7–2 of the PCI index address to address doublewords in the configuration space. Byte locations within a doubleword are addressed by accessing the corresponding bytes of the PCICFGDATA register (see page 6-17). For all other configuration cycles (type zero or type one), the contents of this bit field are driven unchanged on the PCI bus during the address phase of the configuration cycle.
1–0	Reserved	**Reserved** This bit field is ignored during writes to this register. It always returns 0 when read. During the address phase of a configuration cycle, address pins AD1–AD0 are driven to 00b to indicate a type zero configuration cycle, or to 01b to indicate a type one configuration cycle, depending on value of the BUS_NUM bit field.

Programming Notes

After this register (PCICFGADR) is written with the requisite information, an access must be made to the PCICFGDATA register (see page 6-17) to cause the PCI bus configuration cycle to occur on the PCI bus.

This register is reset by a system reset. The bits in this register are not affected by a PCI bus reset. A PCI bus reset is initiated by setting the PCI_RST bit in the HBCTL register (see page 6-3).

This register must be accessed as a doubleword. Accesses that are less than a doubleword in width are treated as normal PCI bus I/O or (if so mapped) GP bus I/O accesses. No configuration cycles are generated if the access is not to the entire doubleword.

In the ÉlanSC520 microcontroller, the doubleword starting at 0CF8h must not be mapped to the GP bus.

The ÉlanSC520 microcontroller does not generate special cycles. In other words, setting the BUS_NUM bit field to 00h, the DEVICE_NUM bit field to 11111b, the FUNCTION_NUM bit field to 111b, and the REGISTER_NUM field to 00000b does not generate a special cycle. Instead, this setup generates a PCI bus configuration write cycle.

PCI Configuration Data (PCICFGDATA)

Direct-Mapped I/O Address 0CFCh

	31	30	29	28	27	26	25	24
Bit	CFG_DATA[31–24]							
Reset	x	x	x	x	x	x	x	x
R/W	R/W							

	23	22	21	20	19	18	17	16
Bit	CFG_DATA[23–16]							
Reset	x	x	x	x	x	x	x	x
R/W	R/W							

	15	14	13	12	11	10	9	8
Bit	CFG_DATA[15–8]							
Reset	x	x	x	x	x	x	x	x
R/W	R/W							

	7	6	5	4	3	2	1	0
Bit	CFG_DATA[7–0]							
Reset	x	x	x	x	x	x	x	x
R/W	R/W							

Register Description

Software reads or writes this register to initiate a PCI bus configuration cycle after setting up the PCICFGADR register (see page 6-15).

Bit Definitions

Bit	Name	Function
31–0	CFG_DATA [31–0]	**Configuration Data** This bit field contains the data for the configuration access cycle (read or write).

Programming Notes

The ENABLE bit must be set in the PCICFGADR register for an access to this register (PCICFGDATA) to result in a PCI bus configuration cycle. Otherwise the cycle accesses the PCI bus I/O space.

This register can be accessed as a doubleword, word, or byte. The appropriate $\overline{\text{CBEx}}$ signal combinations are driven on the PCI bus during the data phase of the configuration cycle.

In the ÉlanSC520 microcontroller, the doubleword starting at 0CFCh must not be mapped to the GP bus.

Device/Vendor ID (PCIDEVID)

I/O Address 0CF8h/0CFCh
PCI Index 00h

	31	30	29	28	27	26	25	24
Bit	DEV_ID[15–8]							
Reset	0	0	1	1	0	0	0	0
R/W	R							

	23	22	21	20	19	18	17	16
Bit	DEV_ID[7–0]							
Reset	0	0	0	0	0	0	0	0
R/W	R							

	15	14	13	12	11	10	9	8
Bit	VDR_ID[15–8]							
Reset	0	0	0	1	0	0	0	0
R/W	R							

	7	6	5	4	3	2	1	0
Bit	VDR_ID[7–0]							
Reset	0	0	1	0	0	0	1	0
R/W	R							

Register Description

This register contains the PCI configuration header space vendor and device identification for the ÉlanSC520 microcontroller host bridge.

Bit Definitions

Bit	Name	Function
31–16	DEV_ID[15–0]	**Device ID** 3000h = This bit field identifies the ÉlanSC520 microcontroller host bridge PCI bus device.
15–0	VDR_ID[15–0]	**Vendor ID** 1022h = This bit field identifies Advanced Micro Devices, Inc., as the vendor of the ÉlanSC520 microcontroller host bridge PCI bus device.

Programming Notes

This register (PCIDEVID) is register number 0 in the host bridge-specific PCI configuration space.

Status/Command (PCISTACMD)

I/O Address 0CF8h/0CFCh
PCI Index 04h

	31	30	29	28	27	26	25	24
Bit	PERR_DET	SIG_SERR	R_MST_ ABT	R_TGT_ ABT	S_TGT_ ABT	S_DVSL_TIM[1–0]		D_PERR_ DET
Reset	0	0	0	0	0	0	1	0
R/W	R/W!	R	R/W!	R/W!	R/W!	R	R	R/W!

	23	22	21	20	19	18	17	16
Bit	FBTB	UDFS	66M_CAP	Reserved				
Reset	1	0	0	0	0	0	0	0
R/W	R	R	R	RSV				

	15	14	13	12	11	10	9	8
Bit	Reserved							SERR_ENB
Reset	0	0	0	0	0	0	0	0
R/W	RSV							R

	7	6	5	4	3	2	1	0
Bit	Reserved	PERR_RES	Reserved			BUS_MAS	MEM_ENB	IO_ENB
Reset	0	0	0	0	0	1	0	0
R/W	RSV	R/W	RSV			R	R/W	R

Register Description

This register contains the PCI configuration header space command and status register bits.

Bit Definitions

Bit	Name	Function
31	PERR_DET	**Parity Error Detected** This bit is set when a parity error is detected by the host bridge master or target controller. 0 = Parity error not detected. 1 = Parity error detected. This bit is cleared by writing a 1.
30	SIG_SERR	**Signaled System Error** This bit is normally used to indicate that the PCI bus agent has asserted the $\overline{\text{SERR}}$ pin, however the host bridge does not drive $\overline{\text{SERR}}$ because the interrupt logic is integrated within the ÉlanSC520 microcontroller. 0 = $\overline{\text{SERR}}$ pin not asserted by the host bridge. This bit is internally fixed to 0.

Bit	Name	Function
29	R_MST_ABT	**Received Master Abort** This bit is set by the host bridge master controller when its transaction is terminated with a master abort. 0 = Transaction was not terminated with a master abort. 1 = Transaction was terminated with a master abort. This bit is cleared by writing a 1.
28	R_TGT_ABT	**Received Target Abort** This bit is set by the host bridge master controller when its transaction is terminated with a target abort. 0 = Transaction was not terminated with a target abort. 1 = Transaction was terminated with a target abort. This bit is cleared by writing a 1.
27	S_TGT_ABT	**Signaled Target Abort** This bit is set by the host bridge target controller when it terminates a transaction with a target abort. The host bridge responds with a target abort when an address parity error is detected. 0 = Target controller did not end a transaction with a target abort. 1 = Target controller ended a transaction with a target abort. This bit is cleared by writing a 1.
26–25	S_DVSL_TIM [1–0]	**Device Select ($\overline{\text{DEVSEL}}$) Timing** These read only bits define the slowest $\overline{\text{DEVSEL}}$ timing for the host bridge target controller. 01 = The host bridge target controller always uses medium $\overline{\text{DEVSEL}}$ timing. This bit field is internally fixed to 01b.
24	D_PERR_DET	**Data Parity Reported** This bit is set by the host bridge master controller when, during a host bridge master controller PCI bus cycle, $\overline{\text{PERR}}$ is asserted by the host bridge or a PCI bus target, and the Parity Error Response bit (PERR_RES in the PCISTACMD register, see page 6-21) is also set. 0 = The master controller did not detect parity error. 1 = The master controller detect parity error. This bit is cleared by writing a 1.
23	FBTB	**Fast Back-to-Back Capable** This read-only bit indicates the host bridge target controller is capable of fast back-to-back transactions. 1 = The host bridge target controller is capable of fast back-to-back transactions.
22	UDFS	**UDF Supported** This read-only bit indicates the host bridge does not support user-definable features. 0 = The host bridge does not support user-definable features.
21	66M_CAP	**66 MHz Capable** This-read only bit indicates the host bridge is not 66-MHz capable. 0 = The host bridge is not 66-MHz capable.
20–9	Reserved	**Reserved** This bit field should be written to 0 for normal system operation.
8	SERR_ENB	**$\overline{\text{SERR}}$ Enable** This bit is normally used to enable the PCI bus agent to drive the $\overline{\text{SERR}}$ pin, however the host bridge does not drive the $\overline{\text{SERR}}$ pin because all interrupt control is integrated within the ÉlanSC520 microcontroller. 0 = The $\overline{\text{SERR}}$ pin is not driven by the ÉlanSC520 microcontroller. This bit is internally fixed to 0.
7	Reserved	**Reserved** This bit field should be written to 0 for normal system operation.

Bit	Name	Function
6	PERR_RES	**Parity Error Response** This bit controls the host bridge's response to parity errors. 0 = The host bridge master and target controllers ignore parity errors. The host bridge treats transactions that have a parity error (address or data) as normal transactions. In other words, it behaves as if nothing is wrong. The D_PERR_DET bit (see page 6-20) is not set for data parity errors, and a target abort is not issued for address parity errors. 1 = The host bridge master and target controllers report parity errors. The host bridge responds to data parity errors by setting the D_PERR_DET bit. The host bridge target controller responds to address parity errors by terminating the transaction with a target abort. The PERR_RES bit must not be changed except during PCI bus initialization after a system reset. See the *Élan™SC520 Microcontroller User's Manual*, order #22004, for information about PCI bus initialization.
5–3	Reserved	**Reserved** This bit field should be written to 0 for normal system operation.
2	BUS_MAS	**Master Enable** This enables the host bridge master controller to generate cycles on the PCI bus. 1 = The host bridge master controller is always enabled. This bit is internally fixed to 1.
1	MEM_ENB	**Memory Access Enable** This bit enables the host bridge target controller to respond to PCI bus master memory cycles. 0 = The host bridge target controller is disabled. 1 = The host bridge target controller is enabled to respond to PCI bus master memory cycles. The MEM_ENB bit must not be changed except during PCI bus initialization after a system reset. See the *Élan™SC520 Microcontroller User's Manual*, order #22004, for information about PCI bus initialization.
0	IO_ENB	**I/O Space Enable** This bit is normally used to enable the host bridge target controller to respond to PCI bus master I/O cycles, however the ÉlanSC520 microcontroller host bridge ignores all I/O cycles from PCI bus masters. 0 = The host bridge does not respond to PCI bus I/O cycles. This bit is internally fixed to 0.

Programming Notes

This register is reset by a system reset or by a PCI bus reset. A PCI bus reset is initiated by setting the PCI_RST bit in the HBCTL register (see page 6-3).

This register (PCISTACMD) is register number 1 in the host bridge-specific PCI configuration space.

The *PCI Local Bus Specification,* Revision 2.2, defines four bit fields in the Status/Command register that are reserved in the ÉlanSC520 microcontroller. These PCI bus functions do not apply to the host bridge controller:

- Wait cycle control (bit 7): the host bridge controller does not support address/data stepping.
- VGA palette snoop enable (bit 5): the host bridge is not a graphics device.
- Memory write and invalidate enable (bit 4): the host bridge does not generate memory write and invalidate commands as a PCI bus master.
- Special cycle recognition (bit 3): the host bridge ignores PCI bus special cycles.

Class Code/Revision ID (PCICCREVID)

I/O Address 0CF8h/0CFCh
PCI Index 08h

	31	30	29	28	27	26	25	24
Bit	CL_CD[7–0]							
Reset	0	0	0	0	0	1	1	0
R/W	R							

	23	22	21	20	19	18	17	16
Bit	SBCL_CD[7–0]							
Reset	0	0	0	0	0	0	0	0
R/W	R							

	15	14	13	12	11	10	9	8
Bit	PRG_IF[7–0]							
Reset	0	0	0	0	0	0	0	0
R/W	R							

	7	6	5	4	3	2	1	0
Bit	REV_ID[7–0]							
Reset	0	0	0	0	0	0	0	0
R/W	R							

Register Description

This register contains the PCI configuration header space class code, program interface and revision identification.

Bit Definitions

Bit	Name	Function
31–24	CL_CD[7–0]	**Base Class Code** This bit field defines the PCI bus base class code of the host bridge. 06h = Bridge Device
23–16	SBCL_CD [7–0]	**Sub Class Code** This bit field defines the PCI bus sub-class code for the host bridge. 00h = Host-PCI bridge
15–8	PRG_IF[7–0]	**Program Interface** This bit field defines the PCI bus program interface type of the host bridge. 00h = Host-PCI bridge
7–0	REV_ID[7–0]	**Revision I.D.** This bit field defines the host bridge revision number. 00h = Revision number

Programming Notes

This register (PCICCREVID) is register number 2 in the host bridge-specific PCI configuration space.

Header Type (PCIHEADTYPE)

I/O Address 0CF8h/0CFCh
PCI Index 0Eh

	7	6	5	4	3	2	1	0
Bit	HDR_TYP[7–0]							
Reset	0	0	0	0	0	0	0	0
R/W	R							

Register Description

This register contains the PCI configuration header space header type.

Bit Definitions

Bit	Name	Function
7–0	HDR_TYP [7–0]	**Header Type** This bit field defines the PCI configuration space header format. 00h = Single-function device, not PCI-to-PCI bridge

Programming Notes

This register (PCIHEADTYPE) is byte 2 of register number 3 in the host bridge-specific PCI configuration space.

Master Retry Time-Out (PCIMRETRYTO)

I/O Address 0CF8h/0CFCh
PCI Index 41h

	7	6	5	4	3	2	1	0
Bit	M_RETRY_TO[7–0]							
Reset	0	0	0	0	0	0	0	0
R/W	R							

Register Description

This register contains the PCI master retry time-out.

Bit Definitions

Bit	Name	Function
7–0	M_RETRY_TO [7–0]	**Master Retry Time-Out** This bit field defines the number of times the master controller retries a transaction before aborting the cycle. For read transactions that are aborted due to a time-out, a data value of FFFFFFFFh is returned to the Am5$_x$86 CPU. 00h = Retry time-out disabled. The master controller continues to retry the transaction until the target responds. This is the default value. Other = The master controller continues to retry the transaction for the number of times programmed into this bit field. For example, if this bit field is set to 80h, the number of retries is 128. Therefore the total number of attempts (including the initial attempt) is the number of retries programmed in this bit field plus 1.

Programming Notes

This register is reset by a system reset or by a PCI bus reset. A PCI bus reset is initiated by setting the PCI_RST bit in the HBCTL register (see page 6-3).

This register (PCIMRETRYTO) is byte 1 of register number 16d in the host bridge-specific PCI configuration space.

This register must not be changed except when there is no outstanding CPU-to-PCI bus transaction pending. This is the default state after a system reset. See the *Élan™SC520 Microcontroller User's Manual*, order #22004, for information about PCI bus initialization.

CHAPTER

7 SDRAM CONTROLLER REGISTERS

7.1 OVERVIEW

This chapter describes the synchronous dynamic random-access memory (SDRAM) controller registers of the ÉlanSC520 microcontroller.

The ÉlanSC520 microcontroller includes an integrated SDRAM controller. The following are some of the SDRAM controller's main features:

- Synchronous DRAM support
- Support for up to 4 banks
- Up to 256 Megabytes of SDRAM
- ECC (single-bit correction/multiple-bit detection) support

The SDRAM controller register set consists of ten memory-mapped configuration region (MMCR) registers used for configuration, control, and status. See the *Élan™SC520 Microcontroller User's Manual*, order #22004, for details about using the SDRAM controller.

Table 7-1 lists the SDRAM controller registers in offset order, with the corresponding description's page number.

7.2 REGISTERS

Table 7-1 SDRAM Controller MMCR Registers

Register Name	Mnemonic	MMCR Offset	Page Number
SDRAM Control	DRCCTL	10h	page 7-2
SDRAM Timing Control	DRCTMCTL	12h	page 7-4
SDRAM Bank Configuration	DRCCFG	14h	page 7-5
SDRAM Bank 0–3 Ending Address	DRCBENDADR	18h	page 7-7
ECC Control	ECCCTL	20h	page 7-9
ECC Status	ECCSTA	21h	page 7-10
ECC Check Bit Position	ECCCKBPOS	22h	page 7-11
ECC Check Code Test	ECCCKTEST	23h	page 7-12
ECC Single-Bit Error Address	ECCSBADD	24h	page 7-14
ECC Multi-Bit Error Address	ECCMBADD	28h	page 7-15

SDRAM Control (DRCCTL)

Memory-Mapped
MMCR Offset 10h

	7	6	5	4	3	2	1	0
Bit	WB_TST_ ENB	Reserved	RFSH_SPD[1–0]		RFSH_ENB	OPMODE_SEL[2–0]		
Reset	0	0	0	1	0	0	0	0
R/W	R/W!	RES	R/W		R/W	R/W		

Register Description

This register controls various features of the SDRAM controller.

Note: *A programmable reset preserves this register's state. See the PRG_RST_ENB bit description on page 3-3.*

Bit Definitions

Bit	Name	Function
7	WB_TST_ ENB	**Write Buffer Test Mode Enable** This bit selects the source of the debugging information available on the three system test pins. 0 = Write buffer test mode is disabled. (System test mode.) The three system test pins function as CF_DRAM, DATASTRB and CF_ROM_GPCS pins. 1 = Write buffer test mode is enabled. The three system test pins function as WBMSTR2–WBMSTR0 for write buffer master trace purposes. If the write buffer is enabled (WB_ENB is set in the DBCTL register, see page 8-3), the WBMSTR2–WBMSTR0 signals provide write buffer master trace information during write cycles from the write buffer to SDRAM. Master trace information specifies which masters contributed to the level (rank) that is currently being written to SDRAM from the write buffer. Contributing masters can be one or more of: $Am5_x86$ CPU, PCI, or GP bus DMA. If the write buffer is disabled, or during read cycles if the write buffer is enabled, the WBMSTR2–WBMSTR0 signals reflect the master that is currently requesting SDRAM access. Software writes to this bit are ignored when the AMDebug™ technology mode is active. This ensures that microcontroller software cannot remove control of the pins from the software driving the AMDebug technology port. If AMDebug technology mode is not active, software can write to this bit. Refer to the *Élan™SC520 Microcontroller User's Manual*, order #22004, for more information about system test mode and write buffer test mode.
6	Reserved	**Reserved** This bit field should be written to 0 for normal system operation.
5–4	RFSH_SPD [1–0]	**SDRAM Refresh Request Speed** These two bits determine the SDRAM refresh request rate. 00 = 7.8 µs 01 = 15.6 µs (default) 10 = 31.2 µs 11 = 62.5 µs
3	RFSH_ENB	**Refresh Enable** 0 = SDRAM refresh is disabled. This mode should be used for SDRAM detection and sizing algorithms. Disabling SDRAM refresh should not be done in normal operation. 1 = SDRAM refresh is enabled. ***Note:*** *Refresh cycles are not generated to SDRAM banks that are not enabled via the BNKx_ENB bits in the DRCBENDADR register (see page 7-7).*

2–0	OPMODE_SEL [2–0]	**SDRAM Operation Mode Select** These commands are used in the SDRAM initialization and detection algorithm. 000 = Normal SDRAM mode. 001 = NOP command enabled. 010 = CPU-to-SDRAM cycles are converted to All Banks Precharge commands. 011 = CPU-to-SDRAM cycles are converted to a Load Mode Register command. The data to be loaded is driven on MA12–MA0. 100 = Auto refresh enabled. 101–111 = Reserved. When specifying NOP, All Banks Precharge, Load Mode Register, or Auto Refresh commands, the command is not actually applied to the SDRAM devices until a CPU access to SDRAM occurs (either a read or write cycle). The specified command is issued to the SDRAM devices during the CPU access rather than the typical SDRAM read or write access. Before using a write cycle to apply any of these SDRAM commands, make sure the WB_ENB bit is clear in the DBCTL register (see page 8-3). Because the command is not issued to the SDRAM until the CPU accesses SDRAM, code executing out of ROM must specifically access SDRAM after selecting a command in the OPMODE_SEL bit field to properly configure the SDRAM device. The Load Mode Register command must be issued to the SDRAM for the setting of the CAS_LAT bit of the DRCTMCTL register to take effect (see page 7-4). In addition to setting the CAS latency as programmed in the CAS_LAT bit, the following fixed parameters are also written to the SDRAM device's mode register when a Load Mode Register command is issued. ■ Burst length: always read burst 4. ■ Burst type: follow non-linear burst. ■ Operating mode: standard operation. ■ Write burst mode: single mode. After a command is issued, the OPMODE_SEL bit field must be cleared to 000b (normal SDRAM mode) before any further SDRAM access.

Programming Notes

SDRAM refresh cycles should only be enabled (via the RFSH_ENB bit) when the OPMODE_SEL bit is 000b.

This register (DRCCTL) should be modified only when the write buffer and the read-ahead feature of the read buffer are disabled in the DBCTL register (see page 8-3).

SDRAM Timing Control (DRCTMCTL)

Memory-Mapped
MMCR Offset 12h

	7	6	5	4	3	2	1	0
Bit	Reserved			CAS_LAT	RAS_PCHG_DLY[1–0]		RAS_CAS_DLY[1–0]	
Reset	0	0	0	1	1	0	1	0
R/W	RSV			R/W	R/W		R/W	

Register Description

This register controls the SDRAM device timing.

Note: *A programmable reset preserves this register's state. See the PRG_RST_ENB bit description on page 3-3.*

Bit Definitions

Bit	Name	Function
7–5	Reserved	**Reserved** This bit field should be written to 0 for normal system operation.
4	CAS_LAT	**SDRAM CAS Latency** This bit controls the $\overline{\text{SCASx}}$ signal latency timing to all the SDRAM banks. 0 = Cycle latency is 2T. 1 = Cycle latency is 3T (default). Where T is 15 ns (one half of the 33-MHz clock cycle). To make the CAS_LAT bit setting take effect, a Load Mode Register command must be issued to the SDRAM devices via the OPMODE_SEL bit field of the DRCCTL register (see page 7-3). Incorrect operation can occur if the CAS_LAT bit is modified and the Load Mode Register command is not issued to the SDRAM devices.
3–2	RAS_PCHG_ DLY[1–0]	**SDRAM RAS Precharge Delay** This bit field determines the $\overline{\text{SRASx}}$ signal precharge delay. 00 = 2T 01 = 3T 10 = 4T (default) 11 = 6T Where T is 15 ns (one half of the 33-MHz clock cycle).
1–0	RAS_CAS_ DLY[1–0]	**SDRAM RAS-to-CAS Delay** This bit field determines the $\overline{\text{SRASx}}$-to-$\overline{\text{SCASx}}$ delay. 00 = 2T 01 = 3T 10 = 4T (default) 11 = Reserved Where T is 15 ns (one half of the 33-MHz clock cycle).

Programming Notes

This register (DRCTMCTL) should be modified only when the write buffer and the read-ahead feature of the read buffer are disabled in the DBCTL register (see page 8-3).

SDRAM Bank Configuration (DRCCFG)

Memory-Mapped
MMCR Offset 14h

	15	14	13	12	11	10	9	8
Bit	BNK3_ BNK_CNT	Reserved	BNK3_COLWDTH[1–0]		BNK2_ BNK_CNT	Reserved	BNK2_COLWDTH[1–0]	
Reset	0	0	0	0	0	0	0	0
R/W	R/W	RSV	R/W		R/W	RSV	R/W	

	7	6	5	4	3	2	1	0
Bit	BNK1_ BNK_CNT	Reserved	BNK1_COLWDTH[1–0]		BNK0_ BNK_CNT	Reserved	BNK0_COLWDTH[1–0]	
Reset	0	0	0	0	0	0	0	0
R/W	R/W	RSV	R/W		R/W	RSV	R/W	

Register Description

This register controls the address column width configuration and SDRAM internal bank count for devices installed in each bank.

Note: *A programmable reset preserves this register's state.See the PRG_RST_ENB bit description on page 3-3.*

Bit Definitions

Bit	Name	Function
15	BNK3_BNK_ CNT	**Bank 3 Internal SDRAM Bank Count** This bit specifies the number of internal banks supported by the SDRAM devices. 0 = 2-bank device 1 = 4-bank device
14	Reserved	**Reserved** This bit field should be written to 0 for normal system operation.
13–12	BNK3_ COLWDTH[1–0]	**Bank 3 Column Address Width** These two bits specify the column address width of the SDRAM devices populated in Bank 3. 00 = 8-bit column address 01 = 9-bit column address 10 = 10-bit column address 11 = 11-bit column address
11	BNK2_BNK_ CNT	**Bank 2 Internal SDRAM Bank Count** This bit specifies the number of internal banks supported by the SDRAM devices. 0 = 2-bank device 1 = 4-bank device
10	Reserved	**Reserved** This bit field should be written to 0 for normal system operation.
9–8	BNK2_ COLWDTH[1–0]	**Bank 2 Column Address Width** These two bits specify the column address width of the SDRAM devices populated in Bank 2. 00 = 8-bit column address 01 = 9-bit column address 10 = 10-bit column address 11 = 11-bit column address

Bit	Name	Function
7	BNK1_BNK_CNT	**Bank 1 Internal SDRAM Bank Count** This bit specifies the number of internal banks supported by the SDRAM devices. 0 = 2-bank device 1 = 4-bank device
6	Reserved	**Reserved** This bit field should be written to 0 for normal system operation.
5–4	BNK1_COLWDTH[1–0]	**Bank 1 Column Address Width** These two bits specify the column address width of the SDRAM devices populated in Bank 1. 00 = 8-bit column address 01 = 9-bit column address 10 = 10-bit column address 11 = 11-bit column address
3	BNK0_BNK_CNT	**Bank 0 Internal SDRAM Bank Count** This bit specifies the number of internal banks supported by the SDRAM devices. 0 = 2-bank device 1 = 4-bank device
2	Reserved	**Reserved** This bit field should be written to 0 for normal system operation.
1–0	BNK0_COLWDTH[1–0]	**Bank 0 Column Address Width** These two bits specify the column address width of the SDRAM devices populated in Bank 0. 00 = 8-bit column address 01 = 9-bit column address 10 = 10-bit column address 11 = 11-bit column address

Programming Notes

This register (DRCCFG) should be modified only when the write buffer and the read-ahead feature of the read buffer are disabled in the DBCTL register (see page 8-3).

Before changing the BNKx_BNK_CNT or BNKx_COLWDTH bit fields, software must issue an All Banks Precharge command to the SDRAM devices via the OPMODE_SEL bit field in the DBCCTL register (see page 7-3). This command returns the SDRAM devices to an idle state and also clears the SDRAM controller's page table entries.

SDRAM Bank 0–3 Ending Address (DRCBENDADR)

Memory-Mapped
MMCR Offset 18h

	31	30	29	28	27	26	25	24
Bit	BNK3_ENB	BNK3_END[28–22]						
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W						

	23	22	21	20	19	18	17	16
Bit	BNK2_ENB	BNK2_END[28–22]						
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W						

	15	14	13	12	11	10	9	8
Bit	BNK1_ENB	BNK1_END[28–22]						
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W						

	7	6	5	4	3	2	1	0
Bit	BNK0_ENB	BNK0_END[28–22]						
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W						

Register Description

This register controls the SDRAM bank enable and bank ending address that specifies the boundary between the banks.

Note: *A programmable reset preserves this register's state.See the PRG_RST_ENB bit description on page 3-3.*

Bit Definitions

Bit	Name	Function
31	BNK3_ENB	**Bank 3 Enable** This bit enables Bank 3. 0 = Disabled 1 = Enabled
30–24	BNK3_END [28–22]	**Bank 3 Ending Address** This bit field determines the Bank 3 boundary, defined in 4-Mbyte increments. This value is compared to physical address bits 28–22 during an SDRAM request to select a bank. Bank 3 is selected if physical address bits 28–22 are less than the BNK3_END bit field value, but greater than or equal to the value specified by the BNK2_END bit field (or the next lower enabled bank's end value if Bank 2 is disabled).
23	BNK2_ENB	**Bank 2 Enable** This bit enables Bank 2. 0 = Disabled 1 = Enabled

Bit	Name	Function
22–16	BNK2_END [28–22]	**Bank 2 Ending Address** This bit field determines the Bank 2 boundary, defined in 4-Mbyte increments. This value is compared to physical address bits 28–22 during an SDRAM request to select a bank. Bank 2 is selected if physical address bits 28–22 are less than the BNK2_END bit field value but greater than or equal to the value specified by the BNK1_END bit field (or the next lower enabled bank's end value if Bank 1 is disabled.
15	BNK1_ENB	**Bank 1 Enable** This bit enables Bank 1. 0 = Disabled 1 = Enabled
14–8	BNK1_END [28–22]	**Bank 1 Ending Address** This bit field determines the Bank 1 boundary, defined in 4-Mbyte increments. This value is compared to physical address bits 28–22 during an SDRAM request to select a bank. Bank 1 is selected if physical address bits 28–22 are less than the BNK1_END bit field value but greater than or equal to the value specified by the BNK0_END bit field (or 0 if Bank 0 is disabled).
7	BNK0_ENB	**Bank 0 Enable** This bit enables Bank 0. 0 = Disabled 1 = Enabled
6–0	BNK0_END [28–22]	**Bank 0 Ending Address** This bit field determines the Bank 0 boundary, defined in 4-Mbyte increments. This value is compared to physical address bits 28–22 during an SDRAM request to select a bank. Bank 0 is selected if physical address bits 28–22 are less than the BNK0_END bit field value.

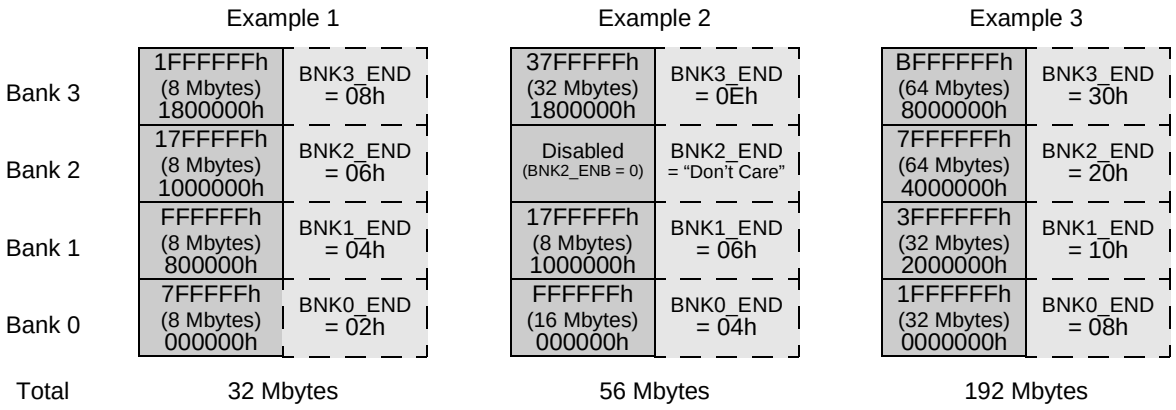
Programming Notes

This register (DRCBENDADR) should be modified only when the write buffer and the read-ahead feature of the read buffer are disabled in the DBCTL register (see page 8-3).

The value specified in each BNKx_END bit field determines the upper address boundary of the corresponding SDRAM bank in 4Mbyte increments. Each bank's lower boundary is determined by the end of the next-lower enabled bank, so the addressable SDRAM space is the concatenation of the enabled banks. The top of the highest configured bank is the top of memory.

If any particular bank is disabled (via its BNKx_ENB bit), the associated BNKx_END bit field value has no effect. Banks do not have to be enabled contiguously. Figure 7-1 gives a few examples of SDRAM bank configuration.

Figure 7-1 Examples of Bank Ending Address Configuration



ECC Control (ECCCTL)

Memory-Mapped
MMCR Offset 20h

	7	6	5	4	3	2	1	0
Bit	Reserved					MULT_INT_ ENB	SGL_INT_ ENB	ECC_ENB
Reset	0	0	0	0	0	0	0	0
R/W	RSV					R/W	R/W	R/W

Register Description

This register controls all the error correction code (ECC) functions.

Note: *A programmable reset preserves this register's state.See the PRG_RST_ENB bit description on page 3-3.*

Bit Definitions

Bit	Name	Function
7–3	Reserved	**Reserved** This bit field should be written to 0 for normal system operation.
2	MULT_INT_ ENB	**Enable Multi-Bit Interrupt** This bit enables the ECC non-maskable interrupt (NMI) source to go active on the detection of a multi-bit error. 0 = Disabled 1 = Enabled The ECC NMI source must also be enabled via the ECC_NMI_ENB bit in the ECCMAP register (see page 12-19).
1	SGL_INT_ ENB	**Enable Single-bit Interrupt** This bit enables the ECC maskable interrupt source to go active on the detection of a single-bit error. 0 = Disabled 1 = Enabled The ECC single-bit interrupt source must also be mapped to an interrupt channel via the ECC_IRQ_MAP bit field in the ECCMAP register (see page 12-20).
0	ECC_ENB	**ECC Enable for All Four Banks** This bit enables ECC for all four banks. 0 = Disabled 1 = Enabled When ECC is enabled, writes to SDRAM include a write of the error correction code (ECC) to the SDRAM device's ECC location, and reads from SDRAM are checked for a correct ECC. Any ECC mismatch is reported in the appropriate registers and the appropriate interrupt is generated if enabled via the MULT_INT_ENB and SGL_INT_ENB bits.

Programming Notes

This register (ECCCTL) should be modified only when the write buffer and the read-ahead feature of the read buffer are disabled in the DBCTL register (see page 8-3).

Before ECC multi-bit or single-bit interrupts are enabled, the ECCMAP register (see page 12-19) must be configured to route the interrupt to the appropriate interrupt request level and priority.

ECC Status (ECCSTA)

Memory-Mapped
MMCR Offset 21h

	7	6	5	4	3	2	1	0
Bit	Reserved						MBIT_ERR	SBIT_ERR
Reset	0	0	0	0	0	0	0	0
R/W	RSV						R/W!	R/W!

Register Description

This register maintains status of the ECC functions if ECC is enabled.

Note: *A programmable reset does not preserve this register's state.*

Bit Definitions

Bit	Name	Function
7–2	Reserved	**Reserved** This bit field should be written to 0 for normal system operation.
1	MBIT_ERR	**Multi-Bit Error Detected** 0 = The event has not occurred, or software cleared this bit by writing a 1. 1 = A multi-bit ECC error has occurred. Software must write a 1 to clear this bit and rearm the logic that captures the physical address where the multi-bit error occurred. The multi-bit error physical address can be read from the ECCMBADD register (see page 7-15).
0	SBIT_ERR	**Single-bit ECC Error** 0 = The event has not occurred, or software cleared this bit by writing a 1. 1 = A single-bit ECC error has occurred. Software must write a 1 to clear this bit and rearm the logic that captures the physical address and bit position where the single-bit error occurred. The single-bit error physical address can be read from the ECCSBADD register (see page 7-14). The single-bit error bit position can be read from the ECCCKBPOS register (see page 7-11).

Programming Notes

Software should write to this register (ECCSTA) only to clear the error status, and only after a status bit was read as 1.

ECC Check Bit Position (ECCCKBPOS)

Memory-Mapped
MMCR Offset 22h

	7	6	5	4	3	2	1	0
Bit	Reserved		ECC_CHK_POS[5–0]					
Reset	0	0	0	0	0	0	0	0
R/W	RSV		R					

Register Description

This register indicates the particular bit in the 32-bit data word or 7-bit check word that caused the single-bit error.

Note: *A programmable reset does not preserve this register's state.*

Bit Definitions

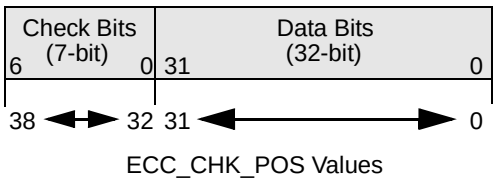
Bit	Name	Function
7–6	Reserved	**Reserved** This bit field should be written to 0 for normal system operation.
5–0	ECC_CHK_ POS[5–0]	**ECC Data Bit Position** This bit field reports the bit location of the single-bit ECC error in either the check bit or data bit field. The bit position is captured upon the detection of a single-bit error, and the bit field is inhibited from capturing subsequent error positions until the SBIT_ERR bit in the ECCSTA register is cleared by writing a 1 (see page 7-10). The data bit field is 32 bits in length and the check bit field is 7 bits in length. Combined, these two bit fields form a 39-bit word. The ECC_CHK_POS bit field contains the numbered bit position in this 39-bit word of the bit that caused the single-bit error. Figure 7-2 relates the encoded bit position to the data and check bit fields.

Programming Notes

Figure 7-2 shows the concatenated check and data bit fields, and the corresponding ECC_CHK_POS values.

Bit 5 of THE ECC_CHK_POS bit field can be used to identify whether the check bit field or the data bit field contains the bit error. If bit 5 is 0, the error is in the data bit field. If bit 5 is 1, the error is in the check bit field.

Figure 7-2 ECC Check Bit and Data Bit Positions



ECC Check Code Test (ECCCKTEST)

Memory-Mapped
MMCR Offset 23h

	7	6	5	4	3	2	1	0
Bit	BAD_CHK_ ENB	FRC_BAD_CHK[6–0]						
Reset	0	0	0	0	0	0	0	0
R/W	R/W!	R/W						

Register Description

This register provides user control of the ECC check code that is written during an SDRAM write cycle. This feature is to be used for test and error-handler development.

Note: *A programmable reset does not preserve this register's state.*

Bit Definitions

Bit	Name	Function
7	BAD_CHK_ ENB	**Enable Bad ECC Check Bits** This bit can be used by test software to enable the FRC_BAD_CHK bit field to replace the correct ECC codes generated by the ÉlanSC520 microcontroller ECC logic for *only* the next single write cycle to SDRAM. 0 = The FRC_BAD_CHK bit field does not replace the generated check bits. The binary pattern in the FRC_BAD_CHK bit field has no effect on the ECC codes written to SDRAM during a write cycle. 1 = The FRC_BAD_CHK bit field replaces the generated check bits for the next SDRAM write. The binary pattern written to the FRC_BAD_CHK bit field is written out as the 7-bit ECC code during the next write cycle to SDRAM. This bit is automatically reset after the FRC_BAD_CHK bit field value is written to the ECC SDRAM during the following write cycle. If this bit (BAD_CHK_ENB) was previously set, then reading 1 from this bit implies that a write cycle to SDRAM did not occur yet, and so the FRC_BAD_CHK bit field value was not yet applied. If the BAD_CHK_ENB bit is read as a 0 after it was previously set, then a write cycle did occur in which the FRC_BAD_CHK bit field value was applied.
6–0	FRC_BAD_ CHK[6–0]	**Force Bad ECC Check Bits** This register provides a way for users to specify their own ECC code for error test purposes. During write cycles to SDRAM, a 7-bit encoded ECC check code (sometimes referred to as a syndrome code) that represents the associated write data is written to ECC SDRAM. This code is automatically generated by the ÉlanSC520 microcontroller when ECC is enabled. If the BAD_CHK_ENB bit is set, the pattern in the FRC_BAD_CHK bit field is written to the ECC storage location on the following write cycle to SDRAM. ***Note:*** *The write buffer should be disabled and the write access should not be cacheable in the Am5$_x$86 CPU write-back cache during this procedure to ensure that the write cycle is propagated to the SDRAM when intended. The write buffer is disabled via the WB_ENB bit in the DBCTL register (see page 8-3).The write can be made non-cacheable, among other ways, by putting the cache in write-through mode or by disabling the cache completely.*

Programming Notes

During a master write access to SDRAM, the ÉlanSC520 microcontroller generates an ECC check code (sometimes referred to as a syndrome code) that is written to the ECC SDRAM devices. During a read cycle, the requested data is read from SDRAM along with the stored ECC check code. A new check code is then generated from the read data and compared to the read ECC check code.

If there is an ECC mismatch, either a single-bit or multiple-bit error has been detected. Single-bit errors are corrected as data is returned to the requesting master (and an interrupt is generated if enabled). Multi-bit errors are not corrected, but an NMI is generated if enabled.

This register (ECCCKTEST) provides a way for the user to alter the ECC check code that is written to the ECC SDRAM. If the ECC check code is altered so that it reflects a single-bit or multiple-bit error, it is detected by the ÉlanSC520 microcontroller when that location is read (if ECC is enabled). This feature can be used for test purposes and also for ECC error-handler development.

Because the write buffer decouples master write requests from the actual SDRAM access, it is advisable to disable the write buffer when using the feature provided by this register. Otherwise, the ECC check code specified in this register might be applied to a write-buffer write access to SDRAM and not the particular access intended. In addition, care should be taken to ensure that the $Am5_x86$ CPU cache allows the write cycle to propagate to the SDRAM.

So that test software need not reproduce the ECC algorithm, Table 7-2 provides a few example ECC check codes that are correct for the associated data.

Table 7-2 Example ECC Check Codes and Associated Data

32-Bit Data Write (Hexadecimal)	Correct ECC Check Code (Binary)
00000000	0000000
FFFFFFFF	0000000
000000BC	1111111
A5A5A5A5	1100011
C0C0C0C0	0101110
00AB0000	1001100
00BC0000	0100000

ECC Single-Bit Error Address (ECCSBADD)

Memory-Mapped
MMCR Offset 24h

	31	30	29	28	27	26	25	24
Bit	Reserved				SB_ADDR[27–24]			
Reset	0	0	0	0	0	0	0	0
R/W	RSV				R			

	23	22	21	20	19	18	17	16
Bit	SB_ADDR[23–16]							
Reset	0	0	0	0	0	0	0	0
R/W	R							

	15	14	13	12	11	10	9	8
Bit	SB_ADDR[15–8]							
Reset	0	0	0	0	0	0	0	0
R/W	R							

	7	6	5	4	3	2	1	0
Bit	SB_ADDR[7–2]						Reserved	
Reset	0	0	0	0	0	0	0	0
R/W	R						RSV	

Register Description

This register contains the physical address of the location in SDRAM that caused a single-bit ECC error.

Note: *A programmable reset does not preserve this register's state.*

Bit Definitions

Bit	Name	Function
31–28	Reserved	**Reserved** This bit field should be written to 0 for normal system operation.
27–2	SB_ADDR [27–2]	**ECC Single-bit Error Address** This bit field contains the physical address (bits 27–2) of the location where a single-bit error occurred. The address is captured upon the detection of a single-bit error, and the bit field is inhibited from capturing subsequent error addresses until the SBIT_ERR bit in the ECCSTA register is cleared by writing a 1 (see page 7-10). ***Note:*** *This register does not include byte enables (BE) from the requesting master, therefore only doubleword resolution is provided by the indication. Use the ECCCKBPOS register to determine which bit was in error (see page 7-11).*
1–0	Reserved	**Reserved** This bit field should be written to 0 for normal system operation.

Programming Notes

ECC Multi-Bit Error Address (ECCMBADD)

Memory-Mapped
MMCR Offset 28h

	31	30	29	28	27	26	25	24
Bit	Reserved				MB_ADDR[27–24]			
Reset	0	0	0	0	0	0	0	0
R/W	RSV				R			

	23	22	21	20	19	18	17	16
Bit	MB_ADDR[23–16]							
Reset	0	0	0	0	0	0	0	0
R/W	R							

	15	14	13	12	11	10	9	8
Bit	MB_ADDR[15–8]							
Reset	0	0	0	0	0	0	0	0
R/W	R							

	7	6	5	4	3	2	1	0
Bit	MB_ADDR[7–2]						Reserved	
Reset	0	0	0	0	0	0	0	0
R/W	R						RSV	

Register Description

This register contains the physical address of the location in SDRAM that caused a multi-bit ECC error.

Note: *A programmable reset does not preserve this register's state.*

Bit Definitions

Bit	Name	Function
31–28	Reserved	**Reserved** This bit field should be written to 0 for normal system operation.
27–2	MB_ADDR [27–2]	**ECC Multi-Bit Error Address** This register contains the physical address (bits 27–2) of the location where a multi-bit error occurred. The address is captured upon the detection of a multi-bit error, and the bit field is inhibited from capturing subsequent error addresses until the MBIT_ERR bit in the ECCSTA register is cleared by writing a 1 (see page 7-10). ***Note:*** *This register does not include byte enables (BE) from the requesting master, therefore only doubleword resolution is provided in the indication.*
1–0	Reserved	**Reserved** This bit field should be written to 0 for normal system operation.

Programming Notes

CHAPTER

8 WRITE BUFFER AND READ BUFFER REGISTER

8.1 OVERVIEW

This chapter describes the write buffer and read buffer register of the ÉlanSC520 microcontroller.

The write buffer and read buffer are two buffering techniques integrated in the ÉlanSC520 microcontroller to increase SDRAM system performance. Although both of these features are tightly integrated with the SDRAM controller, the write buffer and the read buffer's read-ahead feature can be independently enabled via the SDRAM Buffer Control (DBCTL) register.

The write buffer and read buffer register set consists of one memory-mapped configuration region (MMCR) register. See the *Élan™SC520 Microcontroller User's Manual*, order #22004, for details about the write buffer and read buffer.

Table 8-1 lists the DBCTL register and the corresponding description's page number.

8.2 REGISTER

Table 8-1 Write Buffer and Read Buffer MMCR Register

Register Name	Mnemonic	MMCR Offset	Page Number
SDRAM Buffer Control	DBCTL	40h	page 8-2

SDRAM Buffer Control (DBCTL)

Memory-Mapped
MMCR Offset 40h

	7	6	5	4	3	2	1	0
Bit	Reserved			RAB_ENB	WB_WM[1–0]		WB_FLUSH	WB_ENB
Reset	0	0	0	0	0	0	0	0
R/W	RSV			R/W	R/W		R/W!	R/W

Register Description

This register controls all the read buffer read-ahead and write buffer functions.

Note: *A programmable reset does not preserve this register's state.*

Bit Definitions

Bit	Name	Function
7–5	Reserved	**Reserved** This bit field should be written to 0 for normal system operation.
4	RAB_ENB	**Read-Ahead Feature Enable** This bit is used to enable the read-ahead feature of the read buffer. 0 = The read-ahead feature is disabled. 1 = The read-ahead feature is enabled. If the master request is a burst cycle (two or more doublewords), when the current cache line is fetched from SDRAM and stored in the read buffer, the following cache line is also prefetched to take advantage of space locality. If the master request is not a burst cycle, only the rest of the current cache line is fetched from SDRAM into the read buffer. When enabled, the read-ahead feature applies for all bursted read requests from either the $Am5_x86$ CPU or the PCI bus. (The GP bus DMA controller does not perform bursted reads.) During SDRAM sizing or test, disabling the read-ahead feature might improve performance of the sizing or test algorithm. Because most such algorithms test various non-contiguous points in SDRAM, excessive read-ahead thrashing can result. Although this does *not* result in false indications, it can result in a slight performance degradation of the test algorithm. After the SDRAM sizing or test process is complete, the user is free to enable the read-ahead feature of the read buffer when desired.
3-2	WB_WM[1–0]	**Write Buffer Watermark** This bit field specifies the write buffer's watermark setting (i.e., the amount of allocated buffer space above which the write buffer initiates a write to SDRAM). 00 = 28 doublewords (default) 01 = 24 doublewords 10 = 16 doublewords 11 = 8 doublewords As data is written into the write buffer, a new rank of storage is allocated unless the written data can be merged or collapsed into previous ranks. When a write cycle results in a rank allocation that exceeds the watermark setting, the write buffer requests service from the SDRAM controller to initiate write transfers to SDRAM. A higher watermark setting allows the write buffer to fill higher (acquire more master write data) before requesting SDRAM service, resulting in a greater chance of write data merging or collapsing. This is desirable if a large amount of incomplete doubleword writes (i.e., byte, word or three-byte writes) is expected from either the $Am5_x86$ CPU, PCI bus, or GP bus DMA. A lower watermark setting can be used if more complete doublewords are expected, and so merging or collapsing of data is less likely. A lower watermark causes the write buffer to request SDRAM service at a lower threshold, reducing the chance of filling the write buffer.

Bit	Name	Function
1	WB_FLUSH	**Write Buffer Flush** This bit provides manual control over flushing of the write buffer. 0 = Writing 0 has no effect. Reading 0 after first writing 1 indicates that the flush has completed. 1 = Writing 1 causes the write buffer to flush. Reading 1 indicates that the write buffer is still in the process of being flushed. Flushing the write buffer implies that all write buffer data is written out to SDRAM as a high priority before any other SDRAM write or read cycle activity is allowed to take place.
0	WB_ENB	**Write Buffer Enable** This bit is used to enable the write buffer. 0 = The write buffer is disabled. 1 = The write buffer is enabled. The write buffer buffers all write activity from either the $Am5_x86$ CPU, PCI bus, or GP bus DMA. When enabled, the write buffer merges or collapses write data during write cycles and merges read data during read cycles. During SDRAM sizing or test, the write buffer must be disabled to prevent an invalid SDRAM size indicator or false “pass” status during an SDRAM test algorithm. A false pass is possible because the write buffer supports read merging, so that previously written data is returned (read merged) from the write buffer if the read-back occurs before the data has migrated to SDRAM. This can appear as though SDRAM exists even though it might not. If ECC memory is used, the write buffer must also be disabled while the SDRAM initialization software initializes ECC memory by writing to each SDRAM location. After the SDRAM sizing or test process is complete, the user is free to enable the write buffer when desired.

Programming Notes

Software *must* disable the write buffer (i.e., clear the WB_ENB bit) before changing the write buffer watermark (WB_WM bit field).

CHAPTER

9 ROM/FLASH CONTROLLER REGISTERS

9.1 OVERVIEW

This chapter describes the read-only memory (ROM) or Flash memory controller registers of the ÉlanSC520 microcontroller.

The ROM/Flash controller supports up to three ROM device chip select signals, which can be separately enabled. Note that a bank of ROM devices can be accessed with a single chip select (e.g., for building a 32-bit ROM space from four 8-bit ROM devices).

The ROM controller register set consists of three memory-mapped configuration region (MMCR) registers used to configure the ROM controller by programming details about the connected ROM devices' operation mode, device width, device location, and timing. Configuration information is provided for each chip select. See the *Élan™SC520 Microcontroller User's Manual*, order #22004, for details about the ROM controller.

Table 9-1 lists the ROM controller registers in offset order, with the corresponding description's page number.

9.2 REGISTERS

Table 9-1 ROM Controller MMCR Registers

Register Name	Mnemonic	MMCR Offset	Page Number
$\overline{\text{BOOTCS}}$ Control	BOOTCSCTL	50h	page 9-2
$\overline{\text{ROMCS1}}$ Control	ROMCS1CTL	54h	page 9-4
$\overline{\text{ROMCS2}}$ Control	ROMCS2CTL	56h	page 9-6

BOOTCS Control (BOOTCSCTL)

Memory-Mapped MMCR Offset 50h

	15	14	13	12	11	10	9	8
Bit	Reserved			DGP	WIDTH[1–0]		MODE	Reserved
Reset	0	0	0	?	?	?	0	0
R/W	RSV			R	R		R/W	RSV

	7	6	5	4	3	2	1	0
Bit	Reserved		SUB_DLY[1–0]		Reserved	FIRST_DLY[2–0]		
Reset	0	0	1	1	0	1	1	1
R/W	RSV		R/W		RSV	R/W		

Register Description

This register contains configuration information about the location (i.e., SDRAM bus or GP bus), width, operation mode, and timing of the boot ROM that is attached to the ÉlanSC520 microcontroller BOOTCS signal.

Bit Definitions

Bit	Name	Function
15–13	Reserved	**Reserved** This bit field should be written to 0 for normal system operation
12	DGP	**BOOTCS Device SDRAM/GP Bus Select** This bit reflects the location of the boot ROM device that is enabled with the BOOTCS signal. This bit's value is latched from the CFG2 pinstrap when the PWRGOOD pin is asserted. The ROM can be connected to either the SDRAM data bus or to the GP bus. 0 = ROM is on the GP bus. 1 = ROM is on the SDRAM data bus.
11–10	WIDTH[1–0]	**BOOTCS Device Width Select** This bit field reflects the width of the boot ROM. This bit field's value is latched from the CFG1–CFG0 pinstraps when the PWRGOOD pin is asserted. 00 = ROM is 8 bits wide. 01 = ROM is 16 bits wide. 10 = ROM is 32 bits wide. 11 = ROM is 32 bits wide (same as 10b).
9	MODE	**BOOTCS Device Mode** This bit is used to configure the mode of the boot ROM device. 0 = ROM is non-page mode. 1 = ROM is page mode.
8–6	Reserved	**Reserved** This bit field should be written to 0 for normal system operation
5–4	SUB_DLY[1–0]	**BOOTCS Device Delay for Subsequent Access** This bit field is used to configure the number of wait states for all page-mode accesses to the ROM that are subsequent to the first access. This bit field applies only if the MODE bit is 1. 00 = 0 wait states 01 = 1 wait state 10 = 2 wait states 11 = 3 wait states

Bit	Name	Function
3	Reserved	**Reserved** This bit field should be written to 0 for normal system operation.
2–0	FIRST_DLY [2–0]	**$\overline{\text{BOOTCS}}$ Device Delay for First Access** This bit field is used to configure the number of wait states for the first access to the ROM, and for subsequent accesses if the MODE bit is 0. 000 = 0 wait states 001 = 1 wait state 010 = 2 wait states 011 = 3 wait states 100 = 4 wait states 101 = 5 wait states 110 = 6 wait states 111 = 7 wait states

Programming Notes

The device attached to the $\overline{\text{BOOTCS}}$ chip select signal is used as the boot device. Therefore, upon reset this device's DGP and WIDTH bit field values must be delivered to the ROM controller via the CFG2–CFG0 pinstraps. For all other ROM devices, this configuration information is programmed by the initialization software.

ROMCS1 Control (ROMCS1CTL)

Memory-Mapped
MMCR Offset 54h

	15	14	13	12	11	10	9	8
Bit	Reserved			DGP	WIDTH[1–0]		MODE	Reserved
Reset	0	0	0	0	0	0	0	0
R/W	RSV			R/W	R/W		R/W	RSV

	7	6	5	4	3	2	1	0
Bit	Reserved		SUB_DLY[1–0]		Reserved	FIRST_DLY[2–0]		
Reset	0	0	1	1	0	1	1	1
R/W	RSV		R/W		RSV	R/W		

Register Description

This register contains configuration information about the location (i.e., SDRAM bus or GP bus), width, operation mode, and timing of the ROM devices that are attached to the ÉlanSC520 microcontroller ROMCS1 signal.

Bit Definitions

Bit	Name	Function
15–13	Reserved	**Reserved** This bit field should be written to 0 for normal system operation.
12	DGP	**Chip Select 1 Device SDRAM/GP Bus Select** This bit is used to configure the location of the ROM devices that are enabled by ROMCS1. The ROM can be connected either to the SDRAM data bus or to the GP bus. 0 = ROM is on the GP bus. 1 = ROM is on the SDRAM data bus.
11–10	WIDTH[1–0]	**Chip Select 1 Device Width Select** This bit field is used to configure the width of the ROM selected by ROMCS1. 00 = ROM is 8 bits wide. 01 = ROM is 16 bits wide. 10 = ROM is 32 bits wide. 11 = ROM is 32 bits wide.
9	MODE	**Chip Select 1 Device Mode** This bit is used to configure the mode of the ROM selected by ROMCS1. 0 = ROM is non-page mode. 1 = ROM is page mode.
8–6	Reserved	**Reserved** This bit field should be written to 0 for normal system operation.
5–4	SUB_DLY[1–0]	**Chip Select 1 Device Delay for Subsequent Accesses** This bit field is used to configure the number of wait states for all page-mode accesses to the ROM that are subsequent to the first access. This bit field applies only if the MODE bit is 1. 00 = 0 wait states 01 = 1 wait state 10 = 2 wait states 11 = 3 wait states
3	Reserved	**Reserved** This bit field should be written to 0 for normal system operation.

Bit	Name	Function
2–0	FIRST_DLY [2–0]	**Chip Select 1 Device Delay for First Access** This bit field is used to configure the number of wait states for the first access to the ROM, and for subsequent accesses if the MODE bit is 0. 000 = 0 wait states 001 = 1 wait state 010 = 2 wait states 011 = 3 wait states 100 = 4 wait states 101 = 5 wait states 110 = 6 wait states 111 = 7 wait states

Programming Notes

ROMCS2 Control (ROMCS2CTL)

Memory-Mapped
MMCR Offset 56h

	15	14	13	12	11	10	9	8
Bit	Reserved			DGP	WIDTH[1–0]		MODE	Reserved
Reset	0	0	0	0	0	0	0	0
R/W	RSV			R/W	R/W		R/W	RSV

	7	6	5	4	3	2	1	0
Bit	Reserved		SUB_DLY[1–0]		Reserved	FIRST_DLY[2–0]		
Reset	0	0	1	1	0	1	1	1
R/W	RSV		R/W		RSV	R/W		

Register Description

This register contains configuration information about the location (i.e., SDRAM bus or GP bus), width, operation mode, and timing of the ROM devices that are attached to the ÉlanSC520 microcontroller ROMCS2 signal.

Bit Definitions

Bit	Name	Function
15–13	Reserved	**Reserved** This bit field should be written to 0 for normal system operation.
12	DGP	**Chip Select 2 Device SDRAM/GP Bus Select** This bit is used to configure the location of the ROM devices that are enabled by ROMCS2. The ROM can be connected either to the SDRAM data bus or to the GP bus. 0 = ROM is on the GP bus. 1 = ROM is on the SDRAM data bus.
11–10	WIDTH[1–0]	**Chip Select 2 Device Width Select** This bit field is used to configure the width of the ROM selected by ROMCS2. 00 = ROM is 8 bits wide. 01 = ROM is 16 bits wide. 10 = ROM is 32 bits wide. 11 = ROM is 32 bits wide.
9	MODE	**Chip Select 2 Device Mode** This bit is used to configure the mode of the ROM selected by ROMCS2. 0 = ROM is non-page mode. 1 = ROM is page mode.
8–6	Reserved	**Reserved** This bit field should be written to 0 for normal system operation.
5–4	SUB_DLY[1–0]	**Chip Select 2 Device Delay for Subsequent Accesses** This bit field is used to configure the number of wait states for all page-mode accesses to the ROM that are subsequent to the first access. This bit field applies only if the MODE bit is 1. 00 = 0 wait states 01 = 1 wait state 10 = 2 wait states 11 = 3 wait states
3	Reserved	**Reserved** This bit field should be written to 0 for normal system operation.

Bit	Name	Function
2–0	FIRST_DLY [2–0]	**Chip Select 2 Device Delay for First Access** This bit field is used to configure the number of wait states for the first access to the ROM, and for subsequent accesses if the MODE bit is 0. 000 = 0 wait states 001 = 1 wait state 010 = 2 wait states 011 = 3 wait states 100 = 4 wait states 101 = 5 wait states 110 = 6 wait states 111 = 7 wait states

Programming Notes

CHAPTER

10 GENERAL-PURPOSE BUS CONTROLLER REGISTERS

10.1 OVERVIEW

This chapter describes the general-purpose (GP) bus registers of the ÉlanSC520 microcontroller.

The GP bus is used for glueless connection of 8- and 16-bit devices to the ÉlanSC520 microcontroller. To provide the glueless interface, the GP bus interface timing and data bus width is programmable.

The GP bus register set consists of 12 memory-mapped configuration region (MMCR) registers used to configure the GP bus timing and data bus width. See the *Élan™SC520 Microcontroller User's Manual*, order #22004, for details about the GP bus.

Table 10-1 lists the GP bus registers in offset order, with the corresponding description's page number.

10.2 REGISTERS

Table 10-1 GP Bus MMCR Registers

Register Name	Mnemonic	MMCR Offset	Page Number
GP Echo Mode	GPECHO	C00h	page 10-2
GP Chip Select Data Width	GPCSDW	C01h	page 10-3
GP Chip Select Qualification	GPCSQUAL	C02h	page 10-5
GP Chip Select Recovery Time	GPCSRT	C08h	page 10-7
GP Chip Select Pulse Width	GPCSPW	C09h	page 10-8
GP Chip Select Offset	GPCSOFF	C0Ah	page 10-9
GP Read Pulse Width	GPRDW	C0Bh	page 10-10
GP Read Offset	GPRDOFF	C0Ch	page 10-11
GP Write Pulse Width	GPWRW	C0Dh	page 10-12
GP Write Offset	GPWROFF	C0Eh	page 10-13
GPALE Pulse Width	GPALEW	C0Fh	page 10-14
GPALE Offset	GPALEOFF	C10h	page 10-15

GP Echo Mode (GPECHO)

Memory-Mapped
MMCR Offset C00h

	7	6	5	4	3	2	1	0
Bit	Reserved							GP_ECHO_ ENB
Reset	0	0	0	0	0	0	0	0
R/W	RSV							R/W

Register Description

This register is used to enable the GP bus echo mode.

Bit Definitions

Bit	Name	Function
7–1	Reserved	**Reserved** This field should be written to 0 for normal system operation.
0	GP_ECHO_ ENB	**GP Bus Echo Mode Enable** This bit is used to enable the echo mode of GP bus. 0 = Echo mode is disabled. 1 = Echo mode is enabled. Accesses to the microcontroller's integrated peripherals are echoed to the external GP bus

Programming Notes

While echo mode is enabled, the access timing to internal peripherals is modified to adhere to the external timings programmed by the user. This ensures that external peripherals connected to the GP bus still work as programmed by the user. Thus it is possible that access to internal peripherals are slower in echo mode than in normal mode.

While echo mode is enabled, the system designer must ensure that the GP bus timing is not faster than that shown in Table 10-2. The table shows the minimum GP bus timing register values allowed in echo mode. See the corresponding register descriptions beginning on page 10-7.

Table 10-2 GP Bus Echo Mode Minimum Timing

Signal Type	Offset Register Value[1]	Pulse Width Register Value[1]	Recovery Time Register Value[1]
GP Chip Select	1	3	1
GP Read	1	3	—
GP Write	1	3	—
GPALE	0	0	—

Notes:
1. The actual time value is the register value plus one. Times are in units of one internal (33-MHz) clock period.

GP Chip Select Data Width (GPCSDW)

Memory-Mapped
MMCR Offset C01h

	7	6	5	4	3	2	1	0
Bit	GPCS7_DW	GPCS6_DW	GPCS5_DW	GPCS4_DW	GPCS3_DW	GPCS2_DW	GPCS1_DW	GPCS0_DW
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Register Description

This register is used to select the default data width for each of the eight GP bus chip selects.

Bit Definitions

Bit	Name	Function
7	GPCS7_DW	**Data Width Select for $\overline{\text{GPCS7}}$** This bit is used to select the default data width for the GP bus chip select 7 signal. 0 = 8-bit data 1 = 16-bit data
6	GPCS6_DW	**Data Width Select for $\overline{\text{GPCS6}}$** This bit is used to select the default data width for the GP bus chip select 6 signal. 0 = 8-bit data 1 = 16-bit data
5	GPCS5_DW	**Data Width Select for $\overline{\text{GPCS5}}$** This bit is used to select the default data width for the GP bus chip select 5 signal. 0 = 8-bit data 1 = 16-bit data
4	GPCS4_DW	**Data Width Select for $\overline{\text{GPCS4}}$** This bit is used to select the default data width for the GP bus chip select 4 signal. 0 = 8-bit data 1 = 16-bit data
3	GPCS3_DW	**Data Width Select for $\overline{\text{GPCS3}}$** This bit is used to select the default data width for the GP bus chip select 3 signal. 0 = 8-bit data 1 = 16-bit data
2	GPCS2_DW	**Data Width Select for $\overline{\text{GPCS2}}$** This bit is used to select the default data width for the GP bus chip select 2 signal. 0 = 8-bit data 1 = 16-bit data
1	GPCS1_DW	**Data Width Select for $\overline{\text{GPCS1}}$** This bit is used to select the default data width for the GP bus chip select 1 signal. 0 = 8-bit data 1 = 16-bit data
0	GPCS0_DW	**Data Width Select for $\overline{\text{GPCS0}}$** This bit is used to select the default data width for the GP bus chip select 0 signal. 0 = 8-bit data 1 = 16-bit data

Programming Notes

The GPCSx_DW bits are ignored if the $\overline{\text{GPIOCS16}}$ signal is asserted during an I/O access, or if the $\overline{\text{GPMEMCS16}}$ signal is asserted during a memory access.

The $\overline{\text{GPIOCS16}}$ signal is ignored if it is asserted during a memory access. Similarly, the $\overline{\text{GPMEMCS16}}$ signal is ignored if it is asserted during an I/O access.

Before using one of the $\overline{\text{GPCS7}}$–$\overline{\text{GPCS0}}$ signals, software must set the corresponding GPCSx_SEL bit in the CSPFS register (see page 20-7).

GP Chip Select Qualification (GPCSQUAL)

Memory-Mapped
MMCR Offset C02h

	15	14	13	12	11	10	9	8
Bit	GPCS7_RW[1–0]		GPCS6_RW[1–0]		GPCS5_RW[1–0]		GPCS4_RW[1–0]	
Reset	0	0	0	0	0	0	0	0
R/W	R/W		R/W		R/W		R/W	

	7	6	5	4	3	2	1	0
Bit	GPCS3_RW[1–0]		GPCS2_RW[1–0]		GPCS1_RW[1–0]		GPCS0_RW[1–0]	
Reset	0	0	0	0	0	0	0	0
R/W	R/W		R/W		R/W		R/W	

Register Description

This register is used to qualify the GP bus chip selects with $\overline{\text{GPIORD}}$, $\overline{\text{GPIOWR}}$, $\overline{\text{GPMEMRD}}$ or $\overline{\text{GPMEMWR}}$. The qualifiers that can be used depend on the TARGET bit field in the PARx register for the addressed region (see page 2-6). If the TARGET bit field selects GP bus I/O, the GP bus chip selects can be qualified with $\overline{\text{GPIORD}}$ or $\overline{\text{GPIOWR}}$. If the TARGET bit field selects GP bus memory, the GP bus chip selects can be qualified with $\overline{\text{GPMEMRD}}$ or $\overline{\text{GPMEMWR}}$.

Bit Definitions

Bit	Name	Function
15–14	GPCS7_RW [1–0]	**$\overline{\text{GPCS7}}$ Qualifier Selection** This field is used to qualify the GP bus chip select 7 with $\overline{\text{GPIORD}}$, $\overline{\text{GPIOWR}}$, $\overline{\text{GPMEMRD}}$ or $\overline{\text{GPMEMWR}}$. 00 = No qualification. 01 = Qualify the chip select with write strobes ($\overline{\text{GPIOWR}}$ or $\overline{\text{GPMEMWR}}$) 10 = Qualify the chip select with read strobes ($\overline{\text{GPIORD}}$ or $\overline{\text{GPMEMRD}}$) 11 = Qualify the chip select with both strobes ($\overline{\text{GPIORD}}$ and $\overline{\text{GPIOWR}}$, or $\overline{\text{GPMEMRD}}$ and $\overline{\text{GPMEMWR}}$)
13–12	GPCS6_RW [1–0]	**$\overline{\text{GPCS6}}$ Qualifier Selection** This field is used to qualify the GP bus chip select 6 with $\overline{\text{GPIORD}}$, $\overline{\text{GPIOWR}}$, $\overline{\text{GPMEMRD}}$ or $\overline{\text{GPMEMWR}}$. 00 = No qualification. 01 = Qualify the chip select with write strobes ($\overline{\text{GPIOWR}}$ or $\overline{\text{GPMEMWR}}$) 10 = Qualify the chip select with read strobes ($\overline{\text{GPIORD}}$ or $\overline{\text{GPMEMRD}}$) 11 = Qualify the chip select with both strobes ($\overline{\text{GPIORD}}$ and $\overline{\text{GPIOWR}}$, or $\overline{\text{GPMEMRD}}$ and $\overline{\text{GPMEMWR}}$)
11–10	GPCS5_RW [1–0]	**$\overline{\text{GPCS5}}$ Qualifier Selection** This field is used to qualify the GP bus chip select 5 with $\overline{\text{GPIORD}}$, $\overline{\text{GPIOWR}}$, $\overline{\text{GPMEMRD}}$ or $\overline{\text{GPMEMWR}}$. 00 = No qualification. 01 = Qualify the chip select with write strobes ($\overline{\text{GPIOWR}}$ or $\overline{\text{GPMEMWR}}$) 10 = Qualify the chip select with read strobes ($\overline{\text{GPIORD}}$ or $\overline{\text{GPMEMRD}}$) 11 = Qualify the chip select with both strobes ($\overline{\text{GPIORD}}$ and $\overline{\text{GPIOWR}}$, or $\overline{\text{GPMEMRD}}$ and $\overline{\text{GPMEMWR}}$)

Bit	Name	Function
9–8	GPCS4_RW [1–0]	**$\overline{\text{GPCS4}}$ Qualifier Selection** This field is used to qualify the GP bus chip select 4 with $\overline{\text{GPIORD}}$, $\overline{\text{GPIOWR}}$, $\overline{\text{GPMEMRD}}$ or $\overline{\text{GPMEMWR}}$. 00 = No qualification. 01 = Qualify the chip select with write strobes ($\overline{\text{GPIOWR}}$ or $\overline{\text{GPMEMWR}}$) 10 = Qualify the chip select with read strobes ($\overline{\text{GPIORD}}$ or $\overline{\text{GPMEMRD}}$) 11 = Qualify the chip select with both strobes ($\overline{\text{GPIORD}}$ and $\overline{\text{GPIOWR}}$, or $\overline{\text{GPMEMRD}}$ and $\overline{\text{GPMEMWR}}$)
7–6	GPCS3_RW [1–0]	**$\overline{\text{GPCS3}}$ Qualifier Selection** This field is used to qualify the GP bus chip select 3 with $\overline{\text{GPIORD}}$, $\overline{\text{GPIOWR}}$, $\overline{\text{GPMEMRD}}$ or $\overline{\text{GPMEMWR}}$. 00 = No qualification. 01 = Qualify the chip select with write strobes ($\overline{\text{GPIOWR}}$ or $\overline{\text{GPMEMWR}}$) 10 = Qualify the chip select with read strobes ($\overline{\text{GPIORD}}$ or $\overline{\text{GPMEMRD}}$) 11 = Qualify the chip select with both strobes ($\overline{\text{GPIORD}}$ and $\overline{\text{GPIOWR}}$, or $\overline{\text{GPMEMRD}}$ and $\overline{\text{GPMEMWR}}$)
5–4	GPCS2_RW [1–0]	**$\overline{\text{GPCS2}}$ Qualifier Selection** This field is used to qualify the GP bus chip select 2 with $\overline{\text{GPIORD}}$, $\overline{\text{GPIOWR}}$, $\overline{\text{GPMEMRD}}$ or $\overline{\text{GPMEMWR}}$. 00 = No qualification. 01 = Qualify the chip select with write strobes ($\overline{\text{GPIOWR}}$ or $\overline{\text{GPMEMWR}}$) 10 = Qualify the chip select with read strobes ($\overline{\text{GPIORD}}$ or $\overline{\text{GPMEMRD}}$) 11 = Qualify the chip select with both strobes ($\overline{\text{GPIORD}}$ and $\overline{\text{GPIOWR}}$, or $\overline{\text{GPMEMRD}}$ and $\overline{\text{GPMEMWR}}$)
3–2	GPCS1_RW [1–0]	**$\overline{\text{GPCS1}}$ Qualifier Selection** This field is used to qualify the GP bus chip select 1 with $\overline{\text{GPIORD}}$, $\overline{\text{GPIOWR}}$, $\overline{\text{GPMEMRD}}$ or $\overline{\text{GPMEMWR}}$. 00 = No qualification. 01 = Qualify the chip select with write strobes ($\overline{\text{GPIOWR}}$ or $\overline{\text{GPMEMWR}}$) 10 = Qualify the chip select with read strobes ($\overline{\text{GPIORD}}$ or $\overline{\text{GPMEMRD}}$) 11 = Qualify the chip select with both strobes ($\overline{\text{GPIORD}}$ and $\overline{\text{GPIOWR}}$, or $\overline{\text{GPMEMRD}}$ and $\overline{\text{GPMEMWR}}$)
1–0	GPCS0_RW [1–0]	**$\overline{\text{GPCS0}}$ Qualifier Selection** This field is used to qualify the GP bus chip select 0 with $\overline{\text{GPIORD}}$, $\overline{\text{GPIOWR}}$, $\overline{\text{GPMEMRD}}$ or $\overline{\text{GPMEMWR}}$. 00 = No qualification. 01 = Qualify the chip select with write strobes ($\overline{\text{GPIOWR}}$ or $\overline{\text{GPMEMWR}}$) 10 = Qualify the chip select with read strobes ($\overline{\text{GPIORD}}$ or $\overline{\text{GPMEMRD}}$) 11 = Qualify the chip select with both strobes ($\overline{\text{GPIORD}}$ and $\overline{\text{GPIOWR}}$, or $\overline{\text{GPMEMRD}}$ and $\overline{\text{GPMEMWR}}$)

Programming Notes

When this register (GPCSQUAL) is used to qualify a GP bus chip select signal, the external timing relationship between the qualifier and the chip select is not guaranteed. For example, deassertion of the chip select signal could precede the qualifier's deassertion on the external GP bus.

If a single qualifier type (only read or only write) is used to qualify a GP bus chip select signal, the corresponding chip select is not asserted for accesses of the other type. For example, if the GPCS0_RW bit field is set to 01b to qualify GP Chip Select 0 with write strobes only, GP Chip Select 0 is not asserted for read accesses.

Before using one of the $\overline{\text{GPCS7}}$–$\overline{\text{GPCS1}}$ signals, software must set the corresponding GPCSx_SEL bit in the CSPFS register (see page 20-7). Before using the $\overline{\text{GPCS0}}$ signal, software must set the PIO27_SEL bit in the PIOPFS31–16 register (see page 20-5).

GP Chip Select Recovery Time (GPCSRT)

Memory-Mapped
MMCR Offset C08h

	7	6	5	4	3	2	1	0
Bit	GPCS_RECOVR[7–0]							
Reset	0	0	0	0	0	0	0	0
R/W	R/W							

Register Description

This register is used to program the chip select recovery time for all GP bus cycles.

Bit Definitions

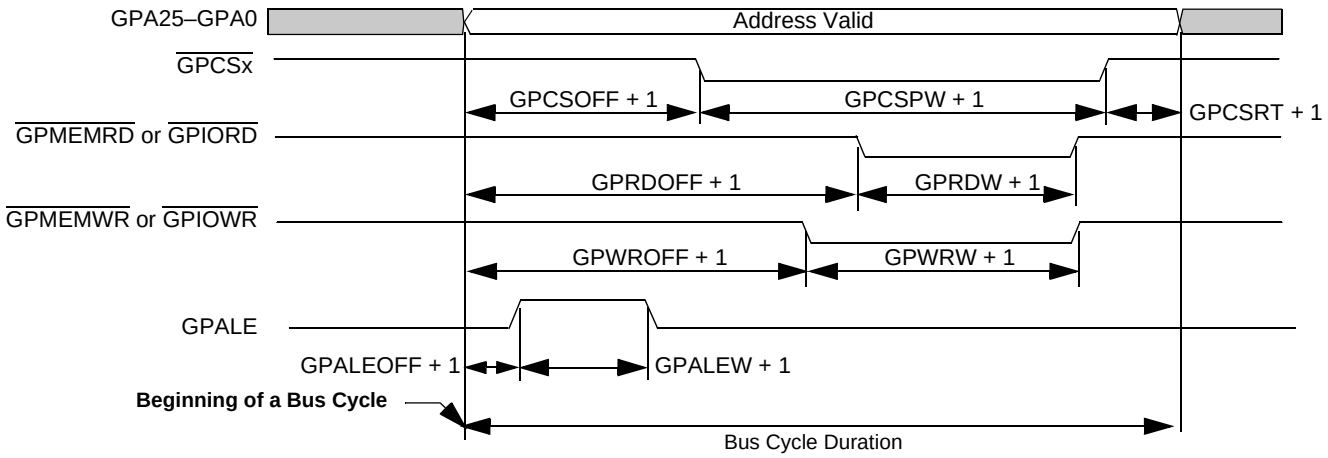
Bit	Name	Function
7–0	GPCS_ RECOVR[7–0]	**Chip Select Recovery Time** This field adjusts the recovery time for all the GP bus chip selects. The resolution of this parameter is one internal (33-MHz) clock period. The recovery time used is (GPCS_RECOVR + 1) internal clock periods; i.e., if GPCS_RECOVR is 0, the recovery time is one clock period.

Programming Notes

Before using one of the $\overline{\text{GPCS7}}$–$\overline{\text{GPCS1}}$ signals, software must set the corresponding GPCSx_SEL bit in the CSPFS register (see page 20-7). Before using the $\overline{\text{GPCS0}}$ signal, software must set the PIO27_SEL bit in the PIOPFS31–16 register (see page 20-5).

Figure 10-1 shows the relationships between the various adjustable GP bus timing parameters.

Figure 10-1 GP Bus Signal Timing Adjustment



Notes:

1. *Timing parameter values are in units of one internal (33-MHz) clock period.*
2. *Timing parameters in the diagram can be adjusted via the corresponding GP bus registers.*
3. *GPCSOFF + GPCSPW + GPCSRT must be greater than or equal to GPRDOFF + GPRDW, GPWROFF + GPWRW, and GPALEOFF + GPALEW.*
4. *Very long GP bus cycles can cause the PCI Host Bridge target controller to violate the 10 µs memory write maximum completion time limit set in the PCI Local Bus Specification, Revision 2.2. In PCI bus 2.2-compliant designs, software must limit the length of GP bus cycles.*

GP Chip Select Pulse Width (GPCSPW)

Memory-Mapped MMCR Offset C09h

	7	6	5	4	3	2	1	0
Bit	GPCS_WIDTH[7–0]							
Reset	0	0	0	0	0	0	0	0
R/W	R/W							

Register Description

This register is used to program the pulse width for all the GP bus chip selects.

Bit Definitions

Bit	Name	Function
7–0	GPCS_WIDTH [7–0]	**Signal Width for the GP Bus Chip Selects** This field adjusts the signal pulse width (time) of all the GP bus chip selects. The resolution of this parameter is one internal (33-MHz) clock period. The width used is (GPCS_WIDTH + 1) internal clock periods; i.e., if GPCS_WIDTH is 0, the pulse is one clock period wide.

Programming Notes

Before using one of the $\overline{\text{GPCS7}}$–$\overline{\text{GPCS1}}$ signals, software must set the corresponding GPCSx_SEL bit in the CSPFS register (see page 20-7). Before using the $\overline{\text{GPCS0}}$ signal, software must set the PIO27_SEL bit in the PIOPFS31–16 register (see page 20-5).

Figure 10-1 on page 10-7 shows the relationships between the various adjustable GP bus timing parameters.

GP Chip Select Offset (GPCSOFF)

Memory-Mapped
MMCR Offset C0Ah

	7	6	5	4	3	2	1	0
Bit	GPCS_OFF[7–0]							
Reset	0	0	0	0	0	0	0	0
R/W	R/W							

Register Description

This register is used to program the offset time from the beginning of a GP bus cycle for all the GP bus chip selects.

Bit Definitions

Bit	Name	Function
7–0	GPCS_OFF [7–0]	**Offset Time for the GP Bus Chip Select** This field adjusts the offset time of all the GP bus chip selects. The resolution of this parameter is one internal (33-MHz) clock period. The offset time used is (GPCS_OFF + 1) internal clock periods; i.e., if GPCS_OFF is 0, the offset time is one clock period.

Programming Notes

Before using one of the $\overline{\text{GPCS7}}$–$\overline{\text{GPCS1}}$ signals, software must set the corresponding GPCSx_SEL bit in the CSPFS register (see page 20-7). Before using the $\overline{\text{GPCS0}}$ signal, software must set the PIO27_SEL bit in the PIOPFS31–16 register (see page 20-5).

Figure 10-1 on page 10-7 shows the relationships between the various adjustable GP bus timing parameters.

GP Read Pulse Width (GPRDW)

Memory-Mapped
MMCR Offset C0Bh

	7	6	5	4	3	2	1	0
Bit	GP_RD_WIDTH[7–0]							
Reset	0	0	0	0	0	0	0	0
R/W	R/W							

Register Description

This register is used to program the signal width for the read strobes ($\overline{\text{GPIORD}}$ and $\overline{\text{GPMEMRD}}$).

Bit Definitions

Bit	Name	Function
7–0	GP_RD_ WIDTH[7–0]	**Signal Width for $\overline{\text{GPIORD}}$ and $\overline{\text{GPMEMRD}}$** This field adjusts the signal pulse width (time) of the GP bus read strobes. The resolution of this parameter is one internal (33-MHz) clock period. The width used is (GP_RD_WIDTH + 1) internal clock periods; i.e., if GP_RD_WIDTH is 0, the pulse is one clock period wide.

Programming Notes

Figure 10-1 on page 10-7 shows the relationships between the various adjustable GP bus timing parameters.

GP Read Offset (GPRDOFF)

Memory-Mapped
MMCR Offset C0Ch

	7	6	5	4	3	2	1	0
Bit	GP_RD_OFF[7–0]							
Reset	0	0	0	0	0	0	0	0
R/W	R/W							

Register Description

This register is used to program the offset time from beginning of a GP bus cycle for $\overline{\text{GPIORD}}$ and $\overline{\text{GPMEMRD}}$.

Bit Definitions

Bit	Name	Function
7–0	GP_RD_OFF [7–0]	**Offset Time for $\overline{\text{GPIORD}}$ and $\overline{\text{GPMEMRD}}$** This field adjusts the offset time of the GP bus read strobes. The resolution of this parameter is one internal (33-MHz) clock period. The offset time used is (GP_RD_OFF + 1) internal clock periods; i.e., if GP_RD_OFF is 0, the offset time is one clock period.

Programming Notes

Figure 10-1 on page 10-7 shows the relationships between the various adjustable GP bus timing parameters.

GP Write Pulse Width (GPWRW)

Memory-Mapped
MMCR Offset C0Dh

	7	6	5	4	3	2	1	0
Bit	GP_WR_WIDTH[7–0]							
Reset	0	0	0	0	0	0	0	0
R/W	R/W							

Register Description

This register is used to program the pulse width for the write strobes ($\overline{\text{GPIOWR}}$ and $\overline{\text{GPMEMWR}}$).

Bit Definitions

Bit	Name	Function
7–0	GP_WR_ WIDTH[7–0]	**Signal Width for ($\overline{\text{GPIOWR}}$ and $\overline{\text{GPMEMWR}}$)** This field adjusts the signal pulse width (time) of the GP bus write strobes. The resolution of this parameter is one internal (33-MHz) clock period. The width used is (GP_WR_WIDTH + 1) internal clock periods; i.e., if GP_WR_WIDTH is 0, the pulse is one clock period wide.

Programming Notes

Figure 10-1 on page 10-7 shows the relationships between the various adjustable GP bus timing parameters.

GP Write Offset (GPWROFF)

Memory-Mapped
MMCR Offset C0Eh

	7	6	5	4	3	2	1	0
Bit	GP_WR_OFF[7–0]							
Reset	0	0	0	0	0	0	0	0
R/W	R/W							

Register Description

This register is used to program the offset from beginning of a GP bus cycle for $\overline{\text{GPIOWR}}$ and $\overline{\text{GPMEMWR}}$.

Bit Definitions

Bit	Name	Function
7–0	GP_WR_OFF [7–0]	**Offset Time for $\overline{\text{GPIOWR}}$ and $\overline{\text{GPMEMWR}}$** This field adjusts the offset time of the GP bus write strobes. The resolution of this parameter is one internal (33-MHz) clock period. The offset time used is (GP_WR_OFF + 1) internal clock periods; i.e., if GP_WR_OFF is 0, the offset time is one clock period.

Programming Notes

Figure 10-1 on page 10-7 shows the relationships between the various adjustable GP bus timing parameters.

GPALE Pulse Width (GPALEW)

Memory-Mapped
MMCR Offset C0Fh

	7	6	5	4	3	2	1	0
Bit	GP_ALE_WIDTH[7–0]							
Reset	0	0	0	0	0	0	0	0
R/W	R/W							

Register Description

This register is used to program the signal width for GPALE.

Bit Definitions

Bit	Name	Function
7–0	GP_ALE_ WIDTH[7–0]	**Signal Width for GPALE** This field adjusts the signal pulse width (time) of the GPALE signal. The resolution of this parameter is one internal (33-MHz) clock period. The width used is (GP_ALE_WIDTH + 1) internal clock periods; i.e., if GP_ALE_WIDTH is 0, the pulse is one clock period wide.

Programming Notes

Figure 10-1 on page 10-7 shows the relationships between the various adjustable GP bus timing parameters.

GPALE Offset (GPALEOFF)

Memory-Mapped
MMCR Offset C10h

	7	6	5	4	3	2	1	0
Bit	GP_ALE_OFF[7–0]							
Reset	0	0	0	0	0	0	0	0
R/W	R/W							

Register Description

This register is used to program the offset from the beginning of a GP bus cycle for GPALE.

Bit Definitions

Bit	Name	Function
7–0	GP_ALE_OFF [7–0]	**Offset Time for GPALE** This field adjusts the offset time of the GPALE signal. The resolution of this parameter is one internal (33-MHz) clock period. The offset time used is (GP_ALE_OFF + 1) internal clock periods; i.e., if GP_ALE_OFF is 0, the offset time is one clock period.

Programming Notes

Figure 10-1 on page 10-7 shows the relationships between the various adjustable GP bus timing parameters.

CHAPTER

11 GP DMA CONTROLLER REGISTERS

11.1 OVERVIEW

This chapter describes the general-purpose bus direct-memory access (GP bus DMA) registers of the ÉlanSC520 microcontroller.

The ÉlanSC520 microcontroller's GP DMA controller consists of two cascaded DMA controller devices that provide a total of seven independent channels. Channels 0–3 are on the slave controller device, and Channels 5–7 are on the master controller device. (Channel 4 is used for cascading the two devices.)

The GP bus DMA register set includes two groups of registers:

- 35 memory-mapped configuration region (MMCR) registers are used to specify the extended page address and the operation of each channel in the enhanced mode.
- 52 direct-mapped I/O registers are used to configure the operation of each DMA controller device (master and slave) within the DMA controller, and to set up the memory address and transfer count for each channel.

See the *Élan™SC520 Microcontroller User's Manual*, order #22004, for details about the GP bus DMA controller.

Table 11-1 and Table 11-2 list each type of GP bus DMA register in offset order, with the corresponding description's page number.

11.2 REGISTERS

Table 11-1 GP-DMA MMCR Registers

Register Name	Mnemonic	MMCR Offset	Page Number
GP-DMA Control	GPDMACTL	D80h	page 11-4
GP-DMA Memory-Mapped I/O	GPDMAMMIO	D81h	page 11-5
GP-DMA Resource Channel Map A	GPDMAEXTCHMAPA	D82h	page 11-6
GP-DMA Resource Channel Map B	GPDMAEXTCHMAPB	D84h	page 11-8
GP-DMA Channel 0 Extended Page	GPDMAEXTPG0	D86h	page 11-10
GP-DMA Channel 1 Extended Page	GPDMAEXTPG1	D87h	page 11-11
GP-DMA Channel 2 Extended Page	GPDMAEXTPG2	D88h	page 11-12
GP-DMA Channel 3 Extended Page	GPDMAEXTPG3	D89h	page 11-13
GP-DMA Channel 5 Extended Page	GPDMAEXTPG5	D8Ah	page 11-14
GP-DMA Channel 6 Extended Page	GPDMAEXTPG6	D8Bh	page 11-15
GP-DMA Channel 7 Extended Page	GPDMAEXTPG7	D8Ch	page 11-16
GP-DMA Channel 3 Extended Transfer Count	GPDMAEXTTC3	D90h	page 11-17
GP-DMA Channel 5 Extended Transfer Count	GPDMAEXTTC5	D91h	page 11-18

Table 11-1 GP-DMA MMCR Registers (Continued)

Register Name	Mnemonic	MMCR Offset	Page Number
GP-DMA Channel 6 Extended Transfer Count	GPDMAEXTTC6	D92h	page 11-19
GP-DMA Channel 7 Extended Transfer Count	GPDMAEXTTC7	D93h	page 11-20
Buffer Chaining Control	GPDMABCCTL	D98h	page 11-21
Buffer Chaining Status	GPDMABCSTA	D99h	page 11-22
Buffer Chaining Interrupt Enable	GPDMABSINTENB	D9Ah	page 11-24
Buffer Chaining Valid	GPDMABCVAL	D9Bh	page 11-25
GP-DMA Channel 3 Next Address Low	GPDMANXTADDL3	DA0h	page 11-26
GP-DMA Channel 3 Next Address High	GPDMANXTADDH3	DA2h	page 11-27
GP-DMA Channel 5 Next Address Low	GPDMANXTADDL5	DA4h	page 11-28
GP-DMA Channel 5 Next Address High	GPDMANXTADDH5	DA6h	page 11-29
GP-DMA Channel 6 Next Address Low	GPDMANXTADDL6	DA8h	page 11-30
GP-DMA Channel 6 Next Address High	GPDMANXTADDH6	DAAh	page 11-31
GP-DMA Channel 7 Next Address Low	GPDMANXTADDL7	DACh	page 11-32
GP-DMA Channel 7 Next Address High	GPDMANXTADDH7	DAEh	page 11-33
GP-DMA Channel 3 Next Transfer Count Low	GPDMANXTTCL3	DB0h	page 11-34
GP-DMA Channel 3 Next Transfer Count High	GPDMANXTTCH3	DB2h	page 11-35
GP-DMA Channel 5 Next Transfer Count Low	GPDMANXTTCL5	DB4h	page 11-36
GP-DMA Channel 5 Next Transfer Count High	GPDMANXTTCH5	DB6h	page 11-37
GP-DMA Channel 6 Next Transfer Count Low	GPDMANXTTCL6	DB8h	page 11-38
GP-DMA Channel 6 Next Transfer Count High	GPDMANXTTCH6	DBAh	page 11-39
GP-DMA Channel 7 Next Transfer Count Low	GPDMANXTTCL7	DBCh	page 11-40
GP-DMA Channel 7 Next Transfer Count High	GPDMANXTTCH7	DBEh	page 11-41

Table 11-2 GP-DMA Direct-Mapped Registers

Register Name	Mnemonic	I/O Address	Page Number
Slave DMA Channel 0 Memory Address	GPDMA0MAR	0000h	page 11-42
Slave DMA Channel 0 Transfer Count	GPDMA0TC	0001h	page 11-43
Slave DMA Channel 1 Memory Address	GPDMA1MAR	0002h	page 11-44
Slave DMA Channel 1 Transfer Count	GPDMA1TC	0003h	page 11-45
Slave DMA Channel 2 Memory Address	GPDMA2MAR	0004h	page 11-46
Slave DMA Channel 2 Transfer Count	GPDMA2TC	0005h	page 11-47
Slave DMA Channel 3 Memory Address	GPDMA3MAR	0006h	page 11-48
Slave DMA Channel 3 Transfer Count	GPDMA3TC	0007h	page 11-49
Slave DMA Channel 0–3 Status	SLDMASTA	0008h	page 11-50
Slave DMA Channel 0–3 Control	SLDMACTL	0008h	page 11-51
Slave Software DRQ(n) Request	SLDMASWREQ	0009h	page 11-53
Slave DMA Channel 0–3 Mask	SLDMAMSK	000Ah	page 11-54
Slave DMA Channel 0–3 Mode	SLDMAMODE	000Bh	page 11-55
Slave DMA Clear Byte Pointer	SLDMACBP	000Ch	page 11-57

Table 11-2 GP-DMA Direct-Mapped Registers (Continued)

Register Name	Mnemonic	I/O Address	Page Number
Slave DMA Controller Reset	SLDMARST	000Dh	page 11-58
Slave DMA Controller Temporary	SLDMATMP	000Dh	page 11-59
Slave DMA Mask Reset	SLDMAMSKRST	000Eh	page 11-60
Slave DMA General Mask	SLDMAGENMSK	000Fh	page 11-61
General 0	GPDMAGR0	0080h	page 11-62
Slave DMA Channel 2 Page	GPDMA2PG	0081h	page 11-63
Slave DMA Channel 3 Page	GPDMA3PG	0082h	page 11-64
Slave DMA Channel 1 Page	GPDMA1PG	0083h	page 11-65
General 1	GPDMAGR1	0084h	page 11-66
General 2	GPDMAGR2	0085h	page 11-67
General 3	GPDMAGR3	0086h	page 11-68
Slave DMA Channel 0 Page	GPDMA0PG	0087h	page 11-69
General 4	GPDMAGR4	0088h	page 11-70
Master DMA Channel 6 Page	GPDMA6PG	0089h	page 11-71
Master DMA Channel 7 Page	GPDMA7PG	008Ah	page 11-72
Master DMA Channel 5 Page	GPDMA5PG	008Bh	page 11-73
General 5	GPDMAGR5	008Ch	page 11-74
General 6	GPDMAGR6	008Dh	page 11-75
General 7	GPDMAGR7	008Eh	page 11-76
General 8	GPDMAGR8	008Fh	page 11-77
Master DMA Channel 4 Memory Address	GPDMA4MAR	00C0h	page 11-78
Master DMA Channel 4 Transfer Count	GPDMA4TC	00C2h	page 11-79
Master DMA Channel 5 Memory Address	GPDMA5MAR	00C4h	page 11-80
Master DMA Channel 5 Transfer Count	GPDMA5TC	00C6h	page 11-81
Master DMA Channel 6 Memory Address	GPDMA6MAR	00C8h	page 11-82
Master DMA Channel 6 Transfer Count	GPDMA6TC	00CAh	page 11-83
Master DMA Channel 7 Memory Address	GPDMA7MAR	00CCh	page 11-84
Master DMA Channel 7 Transfer Count	GPDMA7TC	00CEh	page 11-85
Master DMA Channel 4–7 Status	MSTDMASTA	00D0h	page 11-86
Master DMA Channel 4–7 Control	MSTDMACTL	00D0h	page 11-87
Master Software DRQ(n) Request	MSTDMASWREQ	00D2h	page 11-89
Master DMA Channel 4–7 Mask	MSTDMAMSK	00D4h	page 11-90
Master DMA Channel 4–7 Mode	MSTDMAMODE	00D6h	page 11-91
Master DMA Clear Byte Pointer	MSTDMACBP	00D8h	page 11-93
Master DMA Controller Reset	MSTDMARST	00DAh	page 11-94
Master DMA Controller Temporary	MSTDMATMP	00DAh	page 11-95
Master DMA Mask Reset	MSTDMAMSKRST	00DCh	page 11-96
Master DMA General Mask	MSTDMAGENMSK	00DEh	page 11-97

GP-DMA Control (GPDMACTL)

Memory-Mapped
MMCR Offset D80h

	7	6	5	4	3	2	1	0
Bit	CH7_ALT_SIZE	CH6_ALT_SIZE	CH5_ALT_SIZE	CH3_ALT_SIZE	CLK_MODE		Reserved	ENH_MODE_ENB
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W		RSV	R/W

Register Description

This register provides control for the enhanced mode and selects the clock frequency.

Bit Definitions

Bit	Name	Function
7	CH7_ALT_SIZE	**Alternate Size for Channel 7** 0 = Channel 7 is 16 bits wide. 1 = Channel 7 is 8 bits wide. This bit field is ignored if the ENH_MODE_ENB bit is 0.
6	CH6_ALT_SIZE	**Alternate Size for Channel 6** 0 = Channel 6 is 16 bits wide. 1 = Channel 6 is 8 bits wide. This bit field is ignored if the ENH_MODE_ENB bit is 0.
5	CH5_ALT_SIZE	**Alternate Size for Channel 5** 0 = Channel 5 is 16 bits wide. 1 = Channel 5 is 8 bits wide. This bit field is ignored if the ENH_MODE_ENB bit is 0.
4	CH3_ALT_SIZE	**Alternate Size for Channel 3** 0 = Channel 3 is 8 bits wide. 1 = Channel 3 is 16 bits wide. This bit field is ignored if the ENH_MODE_ENB bit is 0. Note that the Channel 3 default size differs from the Channel 5–7 default size. Do not set the CH3_ALT_SIZE bit if an internal UART is mapped to Channel 3 via the GPDMAEXTCHMAPB register (see page 11-8).
3–2	CLK_MODE	**Clock Mode** 00 = Operate the GP bus DMA controller at 4 Mhz. 01 = Operate the GP bus DMA controller at 8 Mhz. 10 = Operate the GP bus DMA controller at 16 Mhz. 11 = Reserved. The frequencies shown here are nominal. The exact GP bus DMA frequency is derived by dividing the microcontroller's 33-MHz crystal input, which can be 33.000 or 33.333 MHz.
1	Reserved	**Reserved** This bit field should be written to 0 for normal system operation.
0	ENH_MODE_ENB	**Enhanced Mode Enable** 0 = Disable enhanced GP-DMA mode (use normal GP-DMA mode). 1 = Enable enhanced GP-DMA mode.

Programming Notes

GP-DMA Memory-Mapped I/O (GPDMAMMIO)

Memory-Mapped MMCR Offset D81h

	7	6	5	4	3	2	1	0
Bit	DMA7_ MMAP	DMA6_ MMAP	DMA5_ MMAP	Reserved	DMA3_ MMAP	DMA2_ MMAP	DMA1_ MMAP	DMA0_ MMAP
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	RSV	R/W	R/W	R/W	R/W

Register Description

This register provides the selection of an I/O device (enabled by the $\overline{\text{GPIORD}}$ and $\overline{\text{GPIOWR}}$ signals) or a memory-mapped I/O device (enabled by the $\overline{\text{GPMEMRD}}$ or $\overline{\text{GPMEMWR}}$ signals) for each channel.

Bit Definitions

Bit	Name	Function
7	DMA7_MMAP	**Memory-Mapped Device for DMA Channel 7** 0 = DMA Channel 7 connects to a direct-mapped I/O device. 1 = DMA Channel 7 connects to a memory-mapped device.
6	DMA6_MMAP	**Memory-Mapped Device for DMA Channel 6** 0 = DMA Channel 6 connects to a direct-mapped I/O device. 1 = DMA Channel 6 connects to a memory-mapped device.
5	DMA5_MMAP	**Memory-Mapped Device for DMA Channel 5** 0 = DMA Channel 5 connects to a direct-mapped I/O device. 1 = DMA Channel 5 connects to a memory-mapped device.
4	Reserved	**Reserved** This bit field should be written to 0 for normal system operation.
3	DMA3_MMAP	**Memory-Mapped Device for DMA Channel 3** 0 = DMA Channel 3 connects to a direct-mapped I/O device. 1 = DMA Channel 3 connects to a memory-mapped device.
2	DMA2_MMAP	**Memory-Mapped Device for DMA Channel 2** 0 = DMA Channel 2 connects to a direct-mapped I/O device. 1 = DMA Channel 2 connects to a memory-mapped device.
1	DMA1_MMAP	**Memory-Mapped Device for DMA Channel 1** 0 = DMA Channel 1 connects to a direct-mapped I/O device. 1 = DMA Channel 1 connects to a memory-mapped device.
0	DMA0_MMAP	**Memory-Mapped Device for DMA Channel 0** 0 = DMA Channel 0 connects to a direct-mapped I/O device. 1 = DMA Channel 0 connects to a memory-mapped device.

Programming Notes

A bit field in this register (GPDMAMMIO) is ignored if the corresponding channel is mapped to an internal UART via the GPDMAEXTCHMAPB register (see page 11-8).

GP-DMA Resource Channel Map A (GPDMAEXTCHMAPA) Memory-Mapped MMCR Offset D82h

	15	14	13	12	11	10	9	8
Bit	GPDRQ3_CHSEL[3–0]				GPDRQ2_CHSEL[3–0]			
Reset	1	1	1	1	1	1	1	1
R/W	R/W				R/W			

	7	6	5	4	3	2	1	0
Bit	GPDRQ1_CHSEL[3–0]				GPDRQ0_CHSEL[3–0]			
Reset	1	1	1	1	1	1	1	1
R/W	R/W				R/W			

Register Description

This register indicates the channel mapping for the GPDRQ3–GPDRQ0 and $\overline{\text{GPDACK3}}$–$\overline{\text{GPDACK0}}$ pins.

Bit Definitions

Bit	Name	Function
15–12	GPDRQ3_ CHSEL[3–0]	**GPDRQ3 Channel Mapping** Map the GPDRQ3 and $\overline{\text{GPDACK3}}$ pins to a GP-DMA channel: 0000 = Channel 0 0001 = Channel 1 0010 = Channel 2 0011 = Channel 3 0101 = Channel 5 0110 = Channel 6 0111 = Channel 7 All other values are treated as unconnected.
11–8	GPDRQ2_ CHSEL[3–0]	**GPDRQ2 Channel Mapping** Map the GPDRQ2 and $\overline{\text{GPDACK2}}$ pins to a GP-DMA channel: 0000 = Channel 0 0001 = Channel 1 0010 = Channel 2 0011 = Channel 3 0101 = Channel 5 0110 = Channel 6 0111 = Channel 7 All other values are treated as unconnected.

Bit	Name	Function
7–4	GPDRQ1_ CHSEL[3–0]	**GPDRQ1 Channel Mapping** Map the GPDRQ1 and $\overline{\text{GPDACK1}}$ pins to a GP-DMA channel: 0000 = Channel 0 0001 = Channel 1 0010 = Channel 2 0011 = Channel 3 0101 = Channel 5 0110 = Channel 6 0111 = Channel 7 All other values are treated as unconnected.
3–0	GPDRQ0_ CHSEL[3–0]	**GPDRQ0 Channel Mapping** Map the GPDRQ0 and $\overline{\text{GPDACK0}}$ pins to a GP-DMA channel: 0000 = Channel 0 0001 = Channel 1 0010 = Channel 2 0011 = Channel 3 0101 = Channel 5 0110 = Channel 6 0111 = Channel 7 All other values are treated as unconnected.

Programming Notes

GP-DMA Resource Channel Map B (GPDMAEXTCHMAPB) Memory-Mapped MMCR Offset D84h

	15	14	13	12	11	10	9	8
Bit	TXDRQ2_CHSEL[3–0]				TXDRQ1_CHSEL[3–0]			
Reset	1	1	1	1	1	1	1	1
R/W	R/W				R/W			

	7	6	5	4	3	2	1	0
Bit	RXDRQ2_CHSEL[3–0]				RXDRQ1_CHSEL[3–0]			
Reset	1	1	1	1	1	1	1	1
R/W	R/W				R/W			

Register Description

This register indicates the channel mapping for the internal serial port request signals: txdrq2–txdrq1 and rxdrq2–rxdrq1.

Bit Definitions

Bit	Name	Function
15–12	TXDRQ2_ CHSEL[3–0]	**TXDRQ2 Channel Mapping** Map the txdrq2 and txdack2 internal signals to a GP-DMA channel: 0000 = Channel 0 0001 = Channel 1 0010 = Channel 2 0011 = Channel 3 All other values are treated as unconnected.
11–8	TXDRQ1_ CHSEL[3–0]	**TXDRQ1 Channel Mapping** Map the txdrq1 and txdack1 internal signals to a GP-DMA channel: 0000 = Channel 0 0001 = Channel 1 0010 = Channel 2 0011 = Channel 3 All other values are treated as unconnected.
7–4	RXDRQ2_ CHSEL[3–0]	**RXDRQ2 Channel Mapping** Map rxdrq2 and rxdack2 internal signals to a GP-DMA channel: 0000 = Channel 0 0001 = Channel 1 0010 = Channel 2 0011 = Channel 3 All other values are treated as unconnected.

Bit	Name	Function
3–0	RXDRQ1_ CHSEL[3–0]	**RXDRQ1 Channel Mapping** Map the rxdrq1 and rxdack1 internal signals to a GP-DMA channel: 0000 = Channel 0 0001 = Channel 1 0010 = Channel 2 0011 = Channel 3 All other values are treated as unconnected.

Programming Notes

GP-DMA Channel 0 Extended Page (GPDMAEXTPG0)

Memory-Mapped
MMCR Offset D86h

	7	6	5	4	3	2	1	0
Bit	Reserved				DMA0ADR[27–24]			
Reset	0	0	0	0	0	0	0	0
R/W	RSV				R/W			

Register Description

This register provides the extended page address for Channel 0.

Bit Definitions

Bit	Name	Function
7–4	Reserved	**Reserved** This bit field should be written to 0 for normal system operation.
3–0	DMA0ADR [27–24]	**DMA Channel 0 Extended Page Address** This bit field specifies the highest four memory address bits (A27–A24) for Channel 0.

Programming Notes

The extended page address is used in conjunction with the memory address and the page address registers for the associated channel to make up a 28-bit address (A27–A0).

The Channel 0 extended page address bit field (DMA0ADR[27–24]) does not increment or decrement during DMA because this channel does not support enhanced mode.

GP-DMA Channel 1 Extended Page (GPDMAEXTPG1)

Memory-Mapped
MMCR Offset D87h

	7	6	5	4	3	2	1	0
Bit	Reserved				DMA1ADR[27–24]			
Reset	0	0	0	0	0	0	0	0
R/W	RSV				R/W			

Register Description

This register provides the extended page address for Channel 1.

Bit Definitions

Bit	Name	Function
7–4	Reserved	**Reserved** This bit field should be written to 0 for normal system operation.
3–0	DMA1ADR [27–24]	**DMA Channel 1 Extended Page Address** This bit field specifies the highest four memory address bits (A27–A24) for Channel 1.

Programming Notes

The extended page address is used in conjunction with the memory address and the page address registers for the associated channel to make up a 28-bit address (A27–A0).

The Channel 1 extended page address bit field (DMA1ADR[27–24]) does not increment or decrement during DMA because this channel does not support enhanced mode.

GP-DMA Channel 2 Extended Page (GPDMAEXTPG2)

Memory-Mapped
MMCR Offset D88h

	7	6	5	4	3	2	1	0
Bit	Reserved				DMA2ADR[27–24]			
Reset	0	0	0	0	0	0	0	0
R/W	RSV				R/W			

Register Description

This register provides the extended page address for Channel 2.

Bit Definitions

Bit	Name	Function
7–4	Reserved	**Reserved** This bit field should be written to 0 for normal system operation.
3–0	DMA2ADR [27–24]	**DMA Channel 2 Extended Page Address** This bit field specifies the highest four memory address bits (A27–A24) for Channel 2.

Programming Notes

The extended page address is used in conjunction with the memory address and the page address registers for the associated channel to make up a 28-bit address (A27–A0).

The Channel 2 extended page address bit field (DMA2ADR[27–24]) does not increment or decrement during DMA because this channel does not support enhanced mode.

GP-DMA Channel 3 Extended Page (GPDMAEXTPG3)

Memory-Mapped
MMCR Offset D89h

	7	6	5	4	3	2	1	0
Bit	Reserved				DMA3ADR[27–24]			
Reset	0	0	0	0	0	0	0	0
R/W	RSV				R/W!			

Register Description

This register provides the extended page address for Channel 3.

Bit Definitions

Bit	Name	Function
7–4	Reserved	**Reserved** This bit field should be written to 0 for normal system operation.
3–0	DMA3ADR [27–24]	**DMA Channel 3 Extended Page Address** This bit field specifies the highest four memory address bits (A27–A24) for Channel 3.

Programming Notes

The extended page address is used in conjunction with the memory address and the page address registers for the associated channel to make up a 28-bit address (A27–A0).

In enhanced mode, the Channel 3 extended page address bit field (DMA3ADR[27–24]) increments or decrements if the memory address crosses the 16-Mbyte boundary. In normal mode, these bits remain constant during the DMA transfer.

Enhanced mode is enabled by setting the ENH_MODE_ENB bit in the GPDMACTL register (see page 11-4).

GP-DMA Channel 5 Extended Page (GPDMAEXTPG5)

Memory-Mapped
MMCR Offset D8Ah

	7	6	5	4	3	2	1	0
Bit	Reserved				DMA5ADR[27–24]			
Reset	0	0	0	0	0	0	0	0
R/W	RSV				R/W!			

Register Description

This register provides the extended page address for Channel 5.

Bit Definitions

Bit	Name	Function
7–4	Reserved	**Reserved** This bit field should be written to 0 for normal system operation.
3–0	DMA5ADR [27–24]	**DMA Channel 5 Extended Page Address** This bit field specifies the highest four memory address bits (A27–A24) for Channel 5.

Programming Notes

The extended page address is used in conjunction with the memory address and the page address registers for the associated channel to make up a 28-bit address (A27–A0).

In enhanced mode, the Channel 5 extended page address bit field (DMA5ADR[27–24]) increments or decrements if the memory address crosses the 16-Mbyte boundary. In normal mode, these bits remain constant during the DMA transfer.

Enhanced mode is enabled by setting the ENH_MODE_ENB bit in the GPDMACTL register (see page 11-4).

GP-DMA Channel 6 Extended Page (GPDMAEXTPG6)

Memory-Mapped
MMCR Offset D8Bh

	7	6	5	4	3	2	1	0
Bit	Reserved				DMA6ADR[27–24]			
Reset	0	0	0	0	0	0	0	0
R/W	RSV				R/W!			

Register Description

This register provides the extended page address for Channel 6.

Bit Definitions

Bit	Name	Function
7–4	Reserved	**Reserved** This bit field should be written to 0 for normal system operation.
3–0	DMA6ADR [27–24]	**DMA Channel 6 Extended Page Address** This bit field specifies the highest four memory address bits (A27–A24) for Channel 6.

Programming Notes

The extended page address is used in conjunction with the memory address and the page address registers for the associated channel to make up a 28-bit address (A27–A0).

In enhanced mode, the Channel 6 extended page address bit field (DMA6ADR[27–24]) increments or decrements if the memory address crosses the 16-Mbyte boundary. In normal mode, these bits remain constant during the DMA transfer.

Enhanced mode is enabled by setting the ENH_MODE_ENB bit in the GPDMACTL register (see page 11-4).

GP-DMA Channel 7 Extended Page (GPDMAEXTPG7)

Memory-Mapped
MMCR Offset D8Ch

	7	6	5	4	3	2	1	0
Bit	Reserved				DMA7ADR[27–24]			
Reset	0	0	0	0	0	0	0	0
R/W	RSV				R/W!			

Register Description

This register provides the extended page address for Channel 7.

Bit Definitions

Bit	Name	Function
7–4	Reserved	**Reserved** This bit field should be written to 0 for normal system operation.
3–0	DMA7ADR [27–24]	**DMA Channel 7 Extended Page Address** This bit field specifies the highest four memory address bits (A27–A24) for Channel 7.

Programming Notes

The extended page address is used in conjunction with the memory address and the page address registers for the associated channel to make up a 28-bit address (A27–A0).

In enhanced mode, the Channel 7 extended page address bit field (DMA7ADR[27–24]) increments or decrements if the memory address crosses the 16-Mbyte boundary. In normal mode, these bits remain constant during the DMA transfer.

Enhanced mode is enabled by setting the ENH_MODE_ENB bit in the GPDMACTL register (see page 11-4).

GP-DMA Channel 3 Extended Transfer Count (GPDMAEXTTC3) Memory-Mapped MMCR Offset D90h

	7	6	5	4	3	2	1	0
Bit	DMA3TC[23–16]							
Reset	0	0	0	0	0	0	0	0
R/W	R/W!							

Register Description

This register provides extended transfer count bits for Channel 3.

Bit Definitions

Bit	Name	Function
7–0	DMA3TC [23–16]	**DMA Channel 3 Transfer Count Extension** This bit field provides the higher 8 bits of the transfer count for DMA Channel 3. In enhanced mode, this bit field is used with the DMA3TC[15–0] bit field in the GPDMA3TC register (see page 11-49) to allow counts up to 16 M (16,777,216) transfers. In normal mode, the value of this bit field (DMA3TC[23–16]) is ignored.

Programming Notes

Enhanced mode is enabled by setting the ENH_MODE_ENB bit in the GPDMACTL register (see page 11-4).

In PCI bus 2.2-compliant designs, software must limit the length of GP bus DMA demand- or block-mode transfers. Very large transfers could cause the PCI Host Bridge target controller to violate the 10-µs memory write maximum completion time limit set in the *PCI Local Bus Specification,* Revision 2.2.

GP-DMA Channel 5 Extended Transfer Count (GPDMAEXTTC5) Memory-Mapped MMCR Offset D91h

	7	6	5	4	3	2	1	0
Bit	DMA5TC[23–16]							
Reset	0	0	0	0	0	0	0	0
R/W	R/W!							

Register Description

This register provides the extended transfer count bits for Channel 5.

Bit Definitions

Bit	Name	Function
7–0	DMA5TC [23–16]	**DMA Channel 5 Transfer Count Extension** This bit field provides the higher 8 bits of the transfer count for DMA Channel 5. In enhanced mode, this bit field is used with the DMA5TC[15–0] bit field in the GPDMA5TC register (see page 11-81) to allow counts up to 16 M (16,777,216) transfers. In normal mode, the value of this bit field (DMA5TC[23–16]) is ignored.

Programming Notes

Enhanced mode is enabled by setting the ENH_MODE_ENB bit in the GPDMACTL register (see page 11-4).

In PCI bus 2.2-compliant designs, software must limit the length of GP bus DMA demand- or block-mode transfers. Very large transfers could cause the PCI Host Bridge target controller to violate the 10-µs memory write maximum completion time limit set in the *PCI Local Bus Specification,* Revision 2.2.

GP-DMA Channel 6 Extended Transfer Count (GPDMAEXTTC6) Memory-Mapped MMCR Offset D92h

	7	6	5	4	3	2	1	0
Bit	DMA6TC[23–16]							
Reset	0	0	0	0	0	0	0	0
R/W	R/W!							

Register Description

This register provides the extended transfer count bits for Channel 6.

Bit Definitions

Bit	Name	Function
7–0	DMA6TC [23–16]	**DMA Channel 6 Transfer Count Extension** This bit field provides the higher 8 bits of the transfer count for DMA Channel 6. In enhanced mode, this bit field is used with the DMA6TC[15–0] bit field in the GPDMA6TC register (see page 11-83) to allow counts up to 16 M (16,777,216) transfers. In normal mode, the value of this bit field (DMA6TC[23–16]) is ignored.

Programming Notes

Enhanced mode is enabled by setting the ENH_MODE_ENB bit in the GPDMACTL register (see page 11-4).

In PCI bus 2.2-compliant designs, software must limit the length of GP bus DMA demand- or block-mode transfers. Very large transfers could cause the PCI Host Bridge target controller to violate the 10-µs memory write maximum completion time limit set in the *PCI Local Bus Specification,* Revision 2.2.

GP-DMA Channel 7 Extended Transfer Count (GPDMAEXTTC7) Memory-Mapped MMCR Offset D93h

	7	6	5	4	3	2	1	0
Bit	DMA7TC[23–16]							
Reset	0	0	0	0	0	0	0	0
R/W	R/W!							

Register Description

This register provides the extended transfer count bits for Channel 7.

Bit Definitions

Bit	Name	Function
7–0	DMA7TC [23–16]	**DMA Channel 7 Transfer Count Extension** This bit field provides the higher 8 bits of the transfer count for DMA Channel 7. In enhanced mode, this bit field is used with the DMA7TC[15–0] bit field in the GPDMA7TC register (see page 11-85) to allow counts up to 16 M (16,777,216) transfers. In normal mode, the value of this bit field (DMA7TC[23–16]) is ignored.

Programming Notes

Enhanced mode is enabled by setting the ENH_MODE_ENB bit in the GPDMACTL register (see page 11-4).

In PCI bus 2.2-compliant designs, software must limit the length of GP bus DMA demand- or block-mode transfers. Very large transfers could cause the PCI Host Bridge target controller to violate the 10-µs memory write maximum completion time limit set in the *PCI Local Bus Specification,* Revision 2.2.

Buffer Chaining Control (GPDMABCCTL)

Memory-Mapped
MMCR Offset D98h

	7	6	5	4	3	2	1	0
Bit	Reserved				CH7_ BCHN_ENB	CH6_ BCHN_ENB	CH5_ BCHN_ENB	CH3_ BCHN_ENB
Reset	0	0	0	0	0	0	0	0
R/W	RSV				R/W	R/W	R/W	R/W

Register Description

This register controls the buffer chaining feature in enhanced mode.

Bit Definitions

Bit	Name	Function
7–4	Reserved	**Reserved** This bit field should be written to 0 for normal system operation.
3	CH7_BCHN_ ENB	**Buffer Chaining Enable for Channel 7** This bit enables buffer chaining via the Channel 7 Next Address and Channel 7 Next Transfer Count registers. See the register descriptions beginning on page 11-32 and page 11-40. When this bit is set, the CH7_CBUF_VAL bit in the in the GPDMABCVAL register becomes effective (see page 11-25). 0 = Disable buffer chaining. 1 = Enable buffer chaining (enhanced mode only).
2	CH6_BCHN_ ENB	**Buffer Chaining Enable for Channel 6** This bit enables buffer chaining via the Channel 6 Next Address and Channel 6 Next Transfer Count registers. See the register descriptions beginning on page 11-30 and page 11-38. When this bit is set, the CH6_CBUF_VAL bit in the in the GPDMABCVAL register becomes effective (see page 11-25). 0 = Disable buffer chaining. 1 = Enable buffer chaining (enhanced mode only).
1	CH5_BCHN_ ENB	**Buffer Chaining Enable for Channel 5** This bit enables buffer chaining via the Channel 5 Next Address and Channel 5 Next Transfer Count registers. See the register descriptions beginning on page 11-28 and page 11-36. When this bit is set, the CH5_CBUF_VAL bit in the in the GPDMABCVAL register becomes effective (see page 11-25). 0 = Disable buffer chaining. 1 = Enable buffer chaining (enhanced mode only).
0	CH3_BCHN_ ENB	**Buffer Chaining Enable for Channel 3** This bit enables buffer chaining via the Channel 3 Next Address and Channel 3 Next Transfer Count registers. See the register descriptions beginning on page 11-26 and page 11-34. When this bit is set, the CH3_CBUF_VAL bit in the in the GPDMABCVAL register becomes effective (see page 11-25). 0 = Disable buffer chaining. 1 = Enable buffer chaining (enhanced mode only).

Programming Notes

This register is ignored unless enhanced mode is enabled. Enhanced mode is enabled by setting the ENH_MODE_ENB bit in the GPDMACTL register (see page 11-4).

Buffer Chaining Status (GPDMABCSTA)

Memory-Mapped
MMCR Offset D99h

	7	6	5	4	3	2	1	0
Bit	Reserved				CH7_EOB_STA	CH6_EOB_STA	CH5_EOB_STA	CH3_EOB_STA
Reset	0	0	0	0	0	0	0	0
R/W	RSV				R/W!	R/W!	R/W!	R/W!

Register Description

This register provides the status of channels 3, 5, 6, and 7 when buffer chaining is enabled.

Bit Definitions

Bit	Name	Function
7–4	Reserved	**Reserved** This bit field should be written to 0 for normal system operation.
3	CH7_EOB_STA	**End of Current Buffer in Channel 7** This bit is set by the GP bus DMA controller when the current buffer transfer is completed. 0 = The event has not occurred, or software cleared this bit by writing a 1. 1= The current buffer transfer is completed. In addition, an interrupt is generated if bit CH7_INT_ENB is 1 in the GPDMABSINTENB register (see page 11-24). Software should write a 1 to this bit (CH7_EOB_STA) to acknowledge the transfer completion and clear the interrupt condition. This acknowledgment is usually done in the interrupt handling routine if the interrupt is enabled. Writing 0 to this bit has no effect. If this bit is 0, writing a 1 to it has no effect. This bit's value (when read) is meaningful only if buffer chaining is enabled.
2	CH6_EOB_STA	**End of Current Buffer in Channel 6** This bit is set by the GP bus DMA controller when the current buffer transfer is completed. 0 = The event has not occurred, or software cleared this bit by writing a 1. 1= The current buffer transfer is completed. In addition, an interrupt is generated if bit CH6_INT_ENB is 1 in the GPDMABSINTENB register (see page 11-24). Software should write a 1 to this bit (CH6_EOB_STA) to acknowledge the transfer completion and clear the interrupt condition. This acknowledgment is usually done in the interrupt handling routine if the interrupt is enabled. Writing 0 to this bit has no effect. If this bit is 0, writing a 1 to it has no effect. This bit's value (when read) is meaningful only if buffer chaining is enabled.
1	CH5_EOB_STA	**End of Current Buffer in Channel 5** This bit is set by the GP bus DMA controller when the current buffer transfer is completed. 0 = The event has not occurred, or software cleared this bit by writing a 1. 1= The current buffer transfer is completed. In addition, an interrupt is generated if bit CH5_INT_ENB is 1 in the GPDMABSINTENB register (see page 11-24). Software should write a 1 to this bit (CH5_EOB_STA) to acknowledge the transfer completion and clear the interrupt condition. This acknowledgment is usually done in the interrupt handling routine if the interrupt is enabled. Writing 0 to this bit has no effect. If this bit is 0, writing a 1 to it has no effect. This bit's value (when read) is meaningful only if buffer chaining is enabled.

Bit	Name	Function
0	CH3_EOB_STA	**End of Current Buffer in Channel 3** This bit is set by the GP bus DMA controller when the current buffer transfer is completed. 0 = The event has not occurred, or software cleared this bit by writing a 1. 1= The current buffer transfer is completed. In addition, an interrupt is generated if bit CH3_INT_ENB is 1 in the GPDMABSINTENB register (see page 11-24). Software should write a 1 to this bit (CH3_EOB_STA) to acknowledge the transfer completion and clear the interrupt condition. This acknowledgment is usually done in the interrupt handling routine if the interrupt is enabled. Writing 0 to this bit has no effect. If this bit is 0, writing a 1 to it has no effect. This bit's value (when read) is meaningful only if buffer chaining is enabled.

Programming Notes

The interrupt output is shared between the four channels 3, 5, 6, and 7. If more than one interrupt is asserted and software acknowledges one of the interrupts (by setting the corresponding CHx_EOB_STA bit), any other pending interrupts remain asserted. Software has the option of acknowledging all pending CHx_EOB_STA bits and handling all of the interrupts.

Buffer chaining is enabled separately for each channel in the GPDMABCCTL register (see page 11-21).

Buffer Chaining Interrupt Enable (GPDMABSINTENB)

Memory-Mapped
MMCR Offset D9Ah

	7	6	5	4	3	2	1	0
Bit	Reserved				CH7_INT_ ENB	CH6_INT_ ENB	CH5_INT_ ENB	CH3_INT_ ENB
Reset	0	0	0	0	0	0	0	0
R/W	RSV				R/W	R/W	R/W	R/W

Register Description

This register provides the interrupt enable bits for the buffer chaining interrupts. An interrupt is generated when the GP bus DMA controller completes the transfer that is in progress. Software can use this interrupt to queue up a subsequent transfer.

Bit Definitions

Bit	Name	Function
7–4	Reserved	**Reserved** This bit field should be written to 0 for normal system operation.
3	CH7_INT_ ENB	**Interrupt Enable for Channel 7** This bit is applicable only if the Buffer Chaining feature is enabled for this channel. 0 = End-of-buffer interrupt for this channel is disabled. 1 = End-of-buffer interrupt for this channel is enabled.
2	CH6_INT_ ENB	**Interrupt Enable for Channel 6** This bit is applicable only if the Buffer Chaining feature is enabled for this channel. 0 = End-of-buffer interrupt for this channel is disabled. 1 = End-of-buffer interrupt for this channel is enabled.
1	CH5_INT_ ENB	**Interrupt Enable for Channel 5** This bit is applicable only if the Buffer Chaining feature is enabled for this channel. 0 = End-of-buffer interrupt for this channel is disabled. 1 = End-of-buffer interrupt for this channel is enabled.
0	CH3_INT_ ENB	**Interrupt Enable for Channel 3** This bit is applicable only if the Buffer Chaining feature is enabled for this channel. 0 = End-of-buffer interrupt for this channel is disabled. 1 = End-of-buffer interrupt for this channel is enabled.

Programming Notes

Before buffer chaining interrupts are enabled, the DMABCINTMAP register (see page 12-21) must be configured to route the interrupt to the appropriate interrupt request level and priority.

Buffer chaining is enabled separately for each channel in the GPDMABCCTL register (see page 11-21).

Buffer Chaining Valid (GPDMABCVAL)

Memory-Mapped
MMCR Offset D9Bh

	7	6	5	4	3	2	1	0
Bit	Reserved				CH7_ CBUF_VAL	CH6_ CBUF_VAL	CH5_ CBUF_VAL	CH3_ CBUF_VAL
Reset	0	0	0	0	0	0	0	0
R/W	RSV				R/W!	R/W!	R/W!	R/W!

Register Description

This register provides the operating interface with the buffer chaining feature.

Bit Definitions

Bit	Name	Function
7–4	Reserved	**Reserved** This bit field should be written to 0 for normal system operation.
3	CH7_CBUF_ VAL	**Chaining Buffer Valid for Channel 7** 0 = The channel's Next Address registers and Next Transfer Count registers are not valid. Only hardware can clear this bit. Writing a 0 has no effect. 1 = Software sets this bit to indicate that the values of the channel's Next Address registers and Next Transfer Count registers are valid.
2	CH6_CBUF_ VAL	**Chaining Buffer Valid for Channel 6** 0 = The channel's Next Address registers and Next Transfer Count registers are not valid. Only hardware can clear this bit. Writing a 0 has no effect. 1 = Software sets this bit to indicate that the values of the channel's Next Address registers and Next Transfer Count registers are valid.
1	CH5_CBUF_ VAL	**Chaining Buffer Valid for Channel 5** 0 = The channel's Next Address registers and Next Transfer Count registers are not valid. Only hardware can clear this bit. Writing a 0 has no effect. 1 = Software sets this bit to indicate that the values of the channel's Next Address registers and Next Transfer Count registers are valid.
0	CH3_CBUF_ VAL	**Chaining Buffer Valid for Channel 3** 0 = The channel's Next Address registers and Next Transfer Count registers are not valid. Only hardware can clear this bit. Writing a 0 has no effect. 1 = Software sets this bit to indicate that the values of the channel's Next Address registers and Next Transfer Count registers are valid.

Programming Notes

A channel's CHx_CBUF_VAL bit is ignored if buffer chaining is not enabled for the channel. Buffer chaining is enabled separately for each channel in the GPDMABCCTL register (see page 11-21).

If buffer chaining is enabled for a channel, and if the channel's CHx_CBUF_VAL bit is set when the end of the buffer is reached (i.e., when the channel's Transfer Count register reaches 0), then the channel's Memory Address and Transfer Count registers are loaded with the values in the channel's Next Address and Next Count registers, respectively. Then the channel's CHx_CBUF_VAL bit is cleared by the GP bus DMA controller. An interrupt is also generated if the CHx_INT_ENB bit is set in the GPDMABSINTENB register (see page 11-24).

If buffer chaining is enabled and software has not set the channel's CHx_CBUF_VAL bit before the end of the buffer is reached, then the GPTC signal is asserted.

Note that software can only set the CHx_CBUF_VAL bit to indicate to the hardware that the Next Address and Transfer Count registers contain valid information. The CHx_CBUF_VAL bit can be cleared only by hardware.

GP-DMA Channel 3 Next Address Low (GPDMANXTADDL3) Memory-Mapped MMCR Offset DA0h

	15	14	13	12	11	10	9	8
Bit	DMA3_NXT_ADR[15–8]							
Reset	0	0	0	0	0	0	0	0
R/W	R/W							

	7	6	5	4	3	2	1	0
Bit	DMA3_NXT_ADR[7–0]							
Reset	0	0	0	0	0	0	0	0
R/W	R/W							

Register Description

This register provides address bits A15–A0 of the next memory address for Channel 3 in buffer chaining mode.

Bit Definitions

Bit	Name	Function
15–0	DMA3_NXT_ ADR[15–0]	**DMA Channel 3 Next Address Low** This bit field provides address bits A15–A0 of the next memory buffer to be used by Channel 3 in the buffer chaining mode.

Programming Notes

The value of this register is ignored if buffer chaining mode is not enabled for this channel in the GPDMABCCTL register (see page 11-21).

Bit 0 of this register (GPDMANXTADDL3) is ignored for 16-bit mode transfers, so the buffer address used is always even in 16-bit mode. Software should ensure that 16-bit mode buffers always begin on an even word boundary.

GP-DMA Channel 3 Next Address High (GPDMANXTADDH3) Memory-Mapped MMCR Offset DA2h

	15	14	13	12	11	10	9	8
Bit	Reserved				DMA3_NXT_ADR[27–24]			
Reset	0	0	0	0	0	0	0	0
R/W	RSV				R/W			

	7	6	5	4	3	2	1	0
Bit	DMA3_NXT_ADR[23–16]							
Reset	0	0	0	0	0	0	0	0
R/W	R/W							

Register Description

This register provides address bits A27–A16 of the next memory address for Channel 3 in buffer chaining mode.

Bit Definitions

Bit	Name	Function
15–12	Reserved	**Reserved** This bit field should be written to 0 for normal system operation.
11–0	DMA3_NXT_ ADR[27–16]	**DMA Channel 3 Next Address High** This bit field provides address bits A27–A16 of the next memory buffer to be used by Channel 3 in the buffer chaining mode.

Programming Notes

The value of this register is ignored if buffer chaining mode is not enabled for this channel in the GPDMABCCTL register (see page 11-21).

GP-DMA Channel 5 Next Address Low (GPDMANXTADDL5) Memory-Mapped MMCR Offset DA4h

	15	14	13	12	11	10	9	8
Bit	DMA5_NXT_ADR[15–8]							
Reset	0	0	0	0	0	0	0	0
R/W	R/W							

	7	6	5	4	3	2	1	0
Bit	DMA5_NXT_ADR[7–0]							
Reset	0	0	0	0	0	0	0	0
R/W	R/W							

Register Description

This register provides address bits A15–A0 of the next memory address for Channel 5 in buffer chaining mode.

Bit Definitions

Bit	Name	Function
15–0	DMA5_NXT_ ADR[15–0]	**DMA Channel 5 Next Address Low** This bit field provides address bits A15–A0 of the next memory buffer to be used by Channel 5 in the buffer chaining mode.

Programming Notes

The value of this register is ignored if buffer chaining mode is not enabled for this channel in the GPDMABCCTL register (see page 11-21).

Bit 0 of this register (GPDMANXTADDL5) is ignored for 16-bit mode transfers, so the buffer address used is always even in 16-bit mode. Software should ensure that 16-bit mode buffers always begin on an even word boundary.

GP-DMA Channel 5 Next Address High (GPDMANXTADDH5) Memory-Mapped MMCR Offset DA6h

	15	14	13	12	11	10	9	8
Bit	Reserved				DMA5_NXT_ADR[27–24]			
Reset	0	0	0	0	0	0	0	0
R/W	RSV				R/W			

	7	6	5	4	3	2	1	0
Bit	DMA5_NXT_ADR[23–16]							
Reset	0	0	0	0	0	0	0	0
R/W	R/W							

Register Description

This register provides address bits A27–A16 of the next memory address for Channel 5 in buffer chaining mode.

Bit Definitions

Bit	Name	Function
15–12	Reserved	**Reserved** This bit field should be written to 0 for normal system operation.
11–0	DMA5_NXT_ ADR[27–16]	**DMA Channel 5 Next Address High** This bit field provides address bits A27–A16 of the next memory buffer to be used by Channel 5 in the buffer chaining mode.

Programming Notes

The value of this register is ignored if buffer chaining mode is not enabled for this channel in the GPDMABCCTL register (see page 11-21).

GP-DMA Channel 6 Next Address Low (GPDMANXTADDL6) Memory-Mapped MMCR Offset DA8h

	15	14	13	12	11	10	9	8
Bit	DMA6_NXT_ADR[15–8]							
Reset	0	0	0	0	0	0	0	0
R/W	R/W							

	7	6	5	4	3	2	1	0
Bit	DMA6_NXT_ADR[7–0]							
Reset	0	0	0	0	0	0	0	0
R/W	R/W							

Register Description

This register provides address bits A15–A0 of the next memory address for Channel 6 in buffer chaining mode.

Bit Definitions

Bit	Name	Function
15–0	DMA6_NXT_ ADR[15–0]	**DMA Channel 6 Next Address Low** This bit field provides address bits A15–A0 of the next memory buffer to be used by Channel 6 in the buffer chaining mode.

Programming Notes

The value of this register is ignored if buffer chaining mode is not enabled for this channel in the GPDMABCCTL register (see page 11-21).

Bit 0 of this register (GPDMANXTADDL6) is ignored for 16-bit mode transfers, so the buffer address used is always even in 16-bit mode. Software should ensure that 16-bit mode buffers always begin on an even word boundary.

GP-DMA Channel 6 Next Address High (GPDMANXTADDH6) Memory-Mapped MMCR Offset DAAh

	15	14	13	12	11	10	9	8
Bit	Reserved				DMA6_NXT_ADR[27–24]			
Reset	0	0	0	0	0	0	0	0
R/W	RSV				R/W			

	7	6	5	4	3	2	1	0
Bit	DMA6_NXT_ADR[23–16]							
Reset	0	0	0	0	0	0	0	0
R/W	R/W							

Register Description

This register provides address bits A27–A16 of the next memory address for Channel 6 in buffer chaining mode.

Bit Definitions

Bit	Name	Function
15–12	Reserved	**Reserved** This bit field should be written to 0 for normal system operation.
11–0	DMA6_NXT_ ADR[27–16]	**DMA Channel 6 Next Address High** This bit field provides address bits A27–A16 of the next memory buffer to be used by Channel 6 in the buffer chaining mode.

Programming Notes

The value of this register is ignored if buffer chaining mode is not enabled for this channel in the GPDMABCCTL register (see page 11-21).

GP-DMA Channel 7 Next Address Low (GPDMANXTADDL7) Memory-Mapped MMCR Offset DACh

	15	14	13	12	11	10	9	8
Bit	DMA7_NXT_ADR[15–8]							
Reset	0	0	0	0	0	0	0	0
R/W	R/W							

	7	6	5	4	3	2	1	0
Bit	DMA7_NXT_ADR[7–0]							
Reset	0	0	0	0	0	0	0	0
R/W	R/W							

Register Description

This register provides address bits A15–A0 of the next memory address for Channel 7 in buffer chaining mode.

Bit Definitions

Bit	Name	Function
15–0	DMA7_NXT_ ADR[15–0]	**DMA Channel 7 Next Address Low** This bit field provides address bits A15–A0 of the next memory buffer to be used by Channel 7 in the buffer chaining mode.

Programming Notes

The value of this register is ignored if buffer chaining mode is not enabled for this channel in the GPDMABCCTL register (see page 11-21).

Bit 0 of this register (GPDMANXTADDL7) is ignored for 16-bit mode transfers, so the buffer address used is always even in 16-bit mode. Software should ensure that 16-bit mode buffers always begin on an even word boundary.

GP-DMA Channel 7 Next Address High (GPDMANXTADDH7) Memory-Mapped MMCR Offset DAEh

	15	14	13	12	11	10	9	8
Bit	Reserved				DMA7_NXT_ADR[27–24]			
Reset	0	0	0	0	0	0	0	0
R/W	RSV				R/W			

	7	6	5	4	3	2	1	0
Bit	DMA7_NXT_ADR[23–16]							
Reset	0	0	0	0	0	0	0	0
R/W	R/W							

Register Description

This register provides address bits A27–A16 of the next memory address for Channel 7 in buffer chaining mode.

Bit Definitions

Bit	Name	Function
15–12	Reserved	**Reserved** This bit field should be written to 0 for normal system operation.
11–0	DMA7_NXT_ ADR[27–16]	**DMA Channel 7 Next Address High** This bit field provides address bits A27–A16 of the next memory buffer to be used by Channel 7 in the buffer chaining mode.

Programming Notes

The value of this register is ignored if buffer chaining mode is not enabled for this channel in the GPDMABCCTL register (see page 11-21).

GP-DMA Channel 3 Next Transfer Count Low (GPDMANXTTCL3) Memory-Mapped MMCR Offset DB0h

	15	14	13	12	11	10	9	8
Bit	DMA3_NXT_TC[15–8]							
Reset	0	0	0	0	0	0	0	0
R/W	R/W							

	7	6	5	4	3	2	1	0
Bit	DMA3_NXT_TC[7–0]							
Reset	0	0	0	0	0	0	0	0
R/W	R/W							

Register Description

This register provides bits 15–0 of the next transfer count for Channel 3 in buffer chaining mode.

Bit Definitions

Bit	Name	Function
15–0	DMA3_NXT_ TC[15–0]	**DMA Channel 3 Next Transfer Count Low** This bit field provides bits 15–0 of the next transfer count to be used by Channel 3 in the buffer chaining mode.

Programming Notes

The value of this register is ignored if buffer chaining mode is not enabled for this channel in the GPDMABCCTL register (see page 11-21).

In PCI bus 2.2-compliant designs, software must limit the length of GP bus DMA demand- or block-mode transfers. Very large transfers could cause the PCI Host Bridge target controller to violate the 10-µs memory write maximum completion time limit set in the *PCI Local Bus Specification,* Revision 2.2.

GP-DMA Channel 3 Next Transfer Count High (GPDMANXTTCH3) Memory-Mapped MMCR Offset DB2h

	7	6	5	4	3	2	1	0
Bit	DMA3_NXT_TC[23–16]							
Reset	0	0	0	0	0	0	0	0
R/W	R/W							

Register Description

This register provides bits 23–16 of the next transfer count for Channel 3 in buffer chaining mode.

Bit Definitions

Bit	Name	Function
7–0	DMA3_NXT_ TC[23–16]	**DMA Channel 3 Next Transfer Count High** This bit field provides bits 23–16 of the next transfer count to be used by Channel 3 in the buffer chaining mode.

Programming Notes

The value of this register is ignored if buffer chaining mode is not enabled for this channel in the GPDMABCCTL register (see page 11-21).

In PCI bus 2.2-compliant designs, software must limit the length of GP bus DMA demand- or block-mode transfers. Very large transfers could cause the PCI Host Bridge target controller to violate the 10-µs memory write maximum completion time limit set in the *PCI Local Bus Specification,* Revision 2.2.

GP-DMA Channel 5 Next Transfer Count Low (GPDMANXTTCL5) Memory-Mapped MMCR Offset DB4h

	15	14	13	12	11	10	9	8
Bit	DMA5_NXT_TC[15–8]							
Reset	0	0	0	0	0	0	0	0
R/W	R/W							

	7	6	5	4	3	2	1	0
Bit	DMA5_NXT_TC[7–0]							
Reset	0	0	0	0	0	0	0	0
R/W	R/W							

Register Description

This register provides bits 15–0 of the next transfer count for Channel 5 in buffer chaining mode.

Bit Definitions

Bit	Name	Function
15–0	DMA5_NXT_ TC[15–0]	**DMA Channel 5 Next Transfer Count Low** This bit field provides bits 15–0 of the next transfer count to be used by Channel 5 in the buffer chaining mode.

Programming Notes

The value of this register is ignored if buffer chaining mode is not enabled for this channel in the GPDMABCCTL register (see page 11-21).

In PCI bus 2.2-compliant designs, software must limit the length of GP bus DMA demand- or block-mode transfers. Very large transfers could cause the PCI Host Bridge target controller to violate the 10-µs memory write maximum completion time limit set in the *PCI Local Bus Specification,* Revision 2.2.

GP-DMA Channel 5 Next Transfer Count High (GPDMANXTTCH5) Memory-Mapped MMCR Offset DB6h

	7	6	5	4	3	2	1	0
Bit	DMA5_NXT_TC[23–16]							
Reset	0	0	0	0	0	0	0	0
R/W	R/W							

Register Description

This register provides bits 23–16 of the next transfer count for Channel 5 in buffer chaining mode.

Bit Definitions

Bit	Name	Function
7–0	DMA5_NXT_ TC[23–16]	**DMA Channel 5 Next Transfer Count High** This bit field provides bits 23–16 of the next transfer count to be used by Channel 5 in the buffer chaining mode.

Programming Notes

The value of this register is ignored if buffer chaining mode is not enabled for this channel in the GPDMABCCTL register (see page 11-21).

In PCI bus 2.2-compliant designs, software must limit the length of GP bus DMA demand- or block-mode transfers. Very large transfers could cause the PCI Host Bridge target controller to violate the 10-µs memory write maximum completion time limit set in the *PCI Local Bus Specification,* Revision 2.2.

GP-DMA Channel 6 Next Transfer Count Low (GPDMANXTTCL6) Memory-Mapped MMCR Offset DB8h

	15	14	13	12	11	10	9	8
Bit	DMA6_NXT_TC[15–8]							
Reset	0	0	0	0	0	0	0	0
R/W	R/W							

	7	6	5	4	3	2	1	0
Bit	DMA6_NXT_TC[7–0]							
Reset	0	0	0	0	0	0	0	0
R/W	R/W							

Register Description

This register provides bits 15–0 of the next transfer count for Channel 6 in buffer chaining mode.

Bit Definitions

Bit	Name	Function
15–0	DMA6_NXT_TC[15–0]	**DMA Channel 6 Next Transfer Count Low** This bit field provides bits 15–0 of the next transfer count to be used by Channel 6 in the buffer chaining mode.

Programming Notes

The value of this register is ignored if buffer chaining mode is not enabled for this channel in the GPDMABCCTL register (see page 11-21).

In PCI bus 2.2-compliant designs, software must limit the length of GP bus DMA demand- or block-mode transfers. Very large transfers could cause the PCI Host Bridge target controller to violate the 10-µs memory write maximum completion time limit set in the *PCI Local Bus Specification,* Revision 2.2.

GP-DMA Channel 6 Next Transfer Count High (GPDMANXTTCH6) Memory-Mapped MMCR Offset DBAh

	7	6	5	4	3	2	1	0
Bit	DMA6_NXT_TC[23–16]							
Reset	0	0	0	0	0	0	0	0
R/W	R/W							

Register Description

This register provides bits 23–16 of the next transfer count for Channel 6 in buffer chaining mode.

Bit Definitions

Bit	Name	Function
7–0	DMA6_NXT_ TC[23–16]	**DMA Channel 6 Next Transfer Count High** This bit field provides bits 23–16 of the next transfer count to be used by Channel 6 in the buffer chaining mode.

Programming Notes

The value of this register is ignored if buffer chaining mode is not enabled for this channel in the GPDMABCCTL register (see page 11-21).

In PCI bus 2.2-compliant designs, software must limit the length of GP bus DMA demand- or block-mode transfers. Very large transfers could cause the PCI Host Bridge target controller to violate the 10-µs memory write maximum completion time limit set in the *PCI Local Bus Specification,* Revision 2.2.

GP-DMA Channel 7 Next Transfer Count Low (GPDMANXTTCL7) Memory-Mapped MMCR Offset DBCh

	15	14	13	12	11	10	9	8
Bit	DMA7_NXT_TC[15–8]							
Reset	0	0	0	0	0	0	0	0
R/W	R/W							

	7	6	5	4	3	2	1	0
Bit	DMA7_NXT_TC[7–0]							
Reset	0	0	0	0	0	0	0	0
R/W	R/W							

Register Description

This register provides bits 15–0 of the next transfer count for Channel 7 in buffer chaining mode.

Bit Definitions

Bit	Name	Function
15–0	DMA7_NXT_ TC[15–0]	**DMA Channel 7 Next Transfer Count Low** This bit field provides bits 15–0 of the next transfer count to be used by Channel 7 in the buffer chaining mode.

Programming Notes

The value of this register is ignored if buffer chaining mode is not enabled for this channel in the GPDMABCCTL register (see page 11-21).

In PCI bus 2.2-compliant designs, software must limit the length of GP bus DMA demand- or block-mode transfers. Very large transfers could cause the PCI Host Bridge target controller to violate the 10-µs memory write maximum completion time limit set in the *PCI Local Bus Specification,* Revision 2.2.

GP-DMA Channel 7 Next Transfer Count High (GPDMANXTTCH7) Memory-Mapped MMCR Offset DBEh

	7	6	5	4	3	2	1	0
Bit	DMA7_NXT_TC[23–16]							
Reset	0	0	0	0	0	0	0	0
R/W	R/W							

Register Description

This register provides bits 23–16 of the next transfer count for Channel 7 in buffer chaining mode.

Bit Definitions

Bit	Name	Function
7–0	DMA7_NXT_TC[23–16]	**DMA Channel 7 Next Transfer Count High** This bit field provides bits 23–16 of the next transfer count to be used by Channel 7 in the buffer chaining mode.

Programming Notes

The value of this register is ignored if buffer chaining mode is not enabled for this channel in the GPDMABCCTL register (see page 11-21).

In PCI bus 2.2-compliant designs, software must limit the length of GP bus DMA demand- or block-mode transfers. Very large transfers could cause the PCI Host Bridge target controller to violate the 10-µs memory write maximum completion time limit set in the *PCI Local Bus Specification,* Revision 2.2.

Slave DMA Channel 0 Memory Address (GPDMA0MAR)

Direct-Mapped
I/O Address 0000h

	7	6	5	4	3	2	1	0
Bit	DMA0MAR[15–0]							
Reset	x	x	x	x	x	x	x	x
R/W	R/W!							

Register Description

This register contains bits 15–0 of the memory address for Channel 0 during DMA operation.

Bit Definitions

Bit	Name	Function
7–0	DMA0MAR [15–0]	**Lower 16 Bits of DMA Channel 0 Memory Address** This 8-bit field is used in two successive I/O accesses to read or write the channel's memory address bits 15–0. Bits 7–0 of the channel's memory address can be read from or written to this bit field immediately after a write to the SLDMACBP register (see page 11-57). Bits 15–8 of the channel's memory address can be read from or written to this bit field immediately after memory address bits 7–0 are read from or written to this bit field.

Programming Notes

To ensure that the lower byte of this register (GPDMA0MAR) is always accessed first, software should precede any access to this register with a write to the SLDMACBP register (see page 11-57) to clear the slave DMA byte pointer.

The value in this register (GPDMA0MAR) is used with the values in the GPDMA0PG register (see page 11-69) and the GPDMAEXTPG0 register (see page 11-10) to generate DMA address bits 27–0.

Slave DMA Channel 0 Transfer Count (GPDMA0TC)

Direct-Mapped I/O Address 0001h

	7	6	5	4	3	2	1	0
Bit	DMA0TC[15–0]							
Reset	x	x	x	x	x	x	x	x
R/W	R/W!							

Register Description

This register contains bits 15–0 of the transfer count for Channel 0 during DMA operation.

Bit Definitions

Bit	Name	Function
7–0	DMA0TC [15–0]	**DMA Channel 0 Transfer Count (16-Bit Register)** This 8-bit field is used in two successive I/O accesses to read or write the channel's transfer count bits 15–0. Bits 7–0 of the channel's transfer count can be read from or written to this bit field immediately after a write to the SLDMACBP register (see page 11-57). Bits 15–8 of the channel's transfer count can be read from or written to this bit field immediately after transfer count bits 7–0 are read from or written to this bit field. The actual number of transfers is one more than the programmed transfer count value.

Programming Notes

To ensure that the lower byte of this register (GPDMA0TC) is always accessed first, software should precede any access to this register with a write to the SLDMACBP register (see page 11-57) to clear the slave DMA byte pointer.

In PCI bus 2.2-compliant designs, software must limit the length of GP bus DMA demand- or block-mode transfers. Very large transfers could cause the PCI Host Bridge target controller to violate the 10-µs memory write maximum completion time limit set in the *PCI Local Bus Specification,* Revision 2.2.

Slave DMA Channel 1 Memory Address (GPDMA1MAR)

Direct-Mapped
I/O Address 0002h

	7	6	5	4	3	2	1	0
Bit	DMA1MAR[15–0]							
Reset	x	x	x	x	x	x	x	x
R/W	R/W!							

Register Description

This register contains bits 15–0 of the memory address for Channel 1 during DMA operation.

Bit Definitions

Bit	Name	Function
7–0	DMA1MAR [15–0]	**Lower 16 Bits of DMA Channel 1 Memory Address** This 8-bit field is used in two successive I/O accesses to read or write the channel's memory address bits 15–0. Bits 7–0 of the channel's memory address can be read from or written to this bit field immediately after a write to the SLDMACBP register (see page 11-57). Bits 15–8 of the channel's memory address can be read from or written to this bit field immediately after memory address bits 7–0 are read from or written to this bit field.

Programming Notes

To ensure that the lower byte of this register (GPDMA1MAR) is always accessed first, software should precede any access to this register with a write to the SLDMACBP register (see page 11-57) to clear the slave DMA byte pointer.

The value in this register (GPDMA1MAR) is used with the values in the GPDMA1PG register (see page 11-65) and the GPDMAEXTPG1 register (see page 11-11) to generate DMA address bits 27–0.

Slave DMA Channel 1 Transfer Count (GPDMA1TC)

Direct-Mapped I/O Address 0003h

	7	6	5	4	3	2	1	0
Bit	DMA1TC[15–0]							
Reset	x	x	x	x	x	x	x	x
R/W	R/W!							

Register Description

This register contains bits 15–0 of the transfer count for Channel 1 during DMA operation.

Bit Definitions

Bit	Name	Function
7–0	DMA1TC [15–0]	**DMA Channel 1 Transfer Count (16-Bit Register)** This 8-bit field is used in two successive I/O accesses to read or write the channel's transfer count bits 15–0. Bits 7–0 of the channel's transfer count can be read from or written to this bit field immediately after a write to the SLDMACBP register (see page 11-57). Bits 15–8 of the channel's transfer count can be read from or written to this bit field immediately after transfer count bits 7–0 are read from or written to this bit field. The actual number of transfers is one more than the programmed transfer count value.

Programming Notes

To ensure that the lower byte of this register (GPDMA1TC) is always accessed first, software should precede any access to this register with a write to the SLDMACBP register (see page 11-57) to clear the slave DMA byte pointer.

In PCI bus 2.2-compliant designs, software must limit the length of GP bus DMA demand- or block-mode transfers. Very large transfers could cause the PCI Host Bridge target controller to violate the 10-µs memory write maximum completion time limit set in the *PCI Local Bus Specification,* Revision 2.2.

Slave DMA Channel 2 Memory Address (GPDMA2MAR)

Direct-Mapped I/O Address 0004h

	7	6	5	4	3	2	1	0
Bit	DMA2MAR[15–0]							
Reset	x	x	x	x	x	x	x	x
R/W	R/W!							

Register Description

This register contains bits 15–0 of the memory address for Channel 2 during DMA operation.

Bit Definitions

Bit	Name	Function
7–0	DMA2MAR [15–0]	**Lower 16 Bits of DMA Channel 2 Memory Address** This 8-bit field is used in two successive I/O accesses to read or write the channel's memory address bits 15–0. Bits 7–0 of the channel's memory address can be read from or written to this bit field immediately after a write to the SLDMACBP register (see page 11-57). Bits 15–8 of the channel's memory address can be read from or written to this bit field immediately after memory address bits 7–0 are read from or written to this bit field.

Programming Notes

To ensure that the lower byte of this register (GPDMA2MAR) is always accessed first, software should precede any access to this register with a write to the SLDMACBP register (see page 11-57) to clear the slave DMA byte pointer.

The value in this register (GPDMA2MAR) is used with the values in the GPDMA2PG register (see page 11-63) and the GPDMAEXTPG2 register (see page 11-12) to generate DMA address bits 27–0.

Slave DMA Channel 2 Transfer Count (GPDMA2TC)

Direct-Mapped I/O Address 0005h

	7	6	5	4	3	2	1	0
Bit	DMA2TC[15–0]							
Reset	x	x	x	x	x	x	x	x
R/W	R/W!							

Register Description

This register contains bits 15–0 of the transfer count for Channel 2 during DMA operation.

Bit Definitions

Bit	Name	Function
7–0	DMA2TC [15–0]	**DMA Channel 2 Transfer Count (16-Bit Register)** This 8-bit field is used in two successive I/O accesses to read or write the channel's transfer count bits 15–0. Bits 7–0 of the channel's transfer count can be read from or written to this bit field immediately after a write to the SLDMACBP register (see page 11-57). Bits 15–8 of the channel's transfer count can be read from or written to this bit field immediately after transfer count bits 7–0 are read from or written to this bit field. The actual number of transfers is one more than the programmed transfer count value.

Programming Notes

To ensure that the lower byte of this register (GPDMA2TC) is always accessed first, software should precede any access to this register with a write to the SLDMACBP register (see page 11-57) to clear the slave DMA byte pointer.

In PCI bus 2.2-compliant designs, software must limit the length of GP bus DMA demand- or block-mode transfers. Very large transfers could cause the PCI Host Bridge target controller to violate the 10-µs memory write maximum completion time limit set in the *PCI Local Bus Specification,* Revision 2.2.

Slave DMA Channel 3 Memory Address (GPDMA3MAR)

Direct-Mapped
I/O Address 0006h

	7	6	5	4	3	2	1	0
Bit	DMA3MAR[15–0]							
Reset	x	x	x	x	x	x	x	x
R/W	R/W!							

Register Description

This register contains bits 15–0 of the memory address for Channel 3 during DMA operation.

Bit Definitions

Bit	Name	Function
7–0	DMA3MAR [15–0]	**Lower 16 Bits of DMA Channel 3 Memory Address** This 8-bit field is used in two successive I/O accesses to read or write the channel's memory address bits 15–0. Bits 7–0 of the channel's memory address can be read from or written to this bit field immediately after a write to the SLDMACBP register (see page 11-57). Bits 15–8 of the channel's memory address can be read from or written to this bit field immediately after memory address bits 7–0 are read from or written to this bit field.

Programming Notes

To ensure that the lower byte of this register (GPDMA3MAR) is always accessed first, software should precede any access to this register with a write to the SLDMACBP register (see page 11-57) to clear the slave DMA byte pointer.

The value in this register (GPDMA3MAR) is used with the values in the GPDMA3PG register (see page 11-64) and the GPDMAEXTPG3 register (see page 11-13) to generate DMA address bits 27–0.

In enhanced mode, this channel can be programmed for 16-bit DMA transfers (see the descriptions for GPDMACTL register bits CH3_ALT_SIZE and ENH_MODE_ENB, on page 11-4). For 16-bit transfers, this register (GPDMA3MAR) holds address bits 16–1 and address bit 0 is always 0 (i.e., the 16-bit transfers are word-aligned). Because of this, software must load this register with the desired memory address divided by 2 for 16-bit transfers.

Slave DMA Channel 3 Transfer Count (GPDMA3TC)

Direct-Mapped I/O Address 0007h

	7	6	5	4	3	2	1	0
Bit	DMA3TC[15–0]							
Reset	x	x	x	x	x	x	x	x
R/W	R/W!							

Register Description

This register contains bits 15–0 of the transfer count for Channel 3 during DMA operation.

Bit Definitions

Bit	Name	Function
7–0	DMA3TC [15–0]	**DMA Channel 3 Transfer Count (16-Bit Register)** This 8-bit field is used in two successive I/O accesses to read or write the channel's transfer count bits 15–0. Bits 7–0 of the channel's transfer count can be read from or written to this bit field immediately after a write to the SLDMACBP register (see page 11-57). Bits 15–8 of the channel's transfer count can be read from or written to this bit field immediately after transfer count bits 7–0 are read from or written to this bit field. The actual number of transfers is one more than the programmed transfer count value.

Programming Notes

To ensure that the lower byte of this register (GPDMA3TC) is always accessed first, software should precede any access to this register with a write to the SLDMACBP register (see page 11-57) to clear the slave DMA byte pointer.

The value in this register (GPDMA3TC) can be used with the value in the GPDMAEXTTC3 register (see page 11-17) to allow counts of up to 16 M (16,777,216) transfers.

In PCI bus 2.2-compliant designs, software must limit the length of GP bus DMA demand- or block-mode transfers. Very large transfers could cause the PCI Host Bridge target controller to violate the 10-µs memory write maximum completion time limit set in the *PCI Local Bus Specification,* Revision 2.2.

Slave DMA Channel 0–3 Status (SLDMASTA)

Direct-Mapped I/O Address 0008h

	7	6	5	4	3	2	1	0
Bit	DMAR3	DMAR2	DMAR1	DMAR0	TC3	TC2	TC1	TC0
Reset	?	?	?	?	0	0	0	0
R/W	R	R	R	R	R!	R!	R!	R!

Register Description

This register indicates the request status and terminal count status for Channels 0–3.

Bit Definitions

Bit	Name	Function
7	DMAR3	**Channel 3 DMA Request** 0 = Channel 3 DMA request not pending 1 = Channel 3 DMA request pending
6	DMAR2	**Channel 2 DMA Request** 0 = Channel 2 DMA request not pending 1 = Channel 2 DMA request pending
5	DMAR1	**Channel 1 DMA Request** 0 = Channel 1 DMA request not pending 1 = Channel 1 DMA request pending
4	DMAR0	**Channel 0 DMA Request** 0 = Channel 0 DMA request not pending 1 = Channel 0 DMA request pending
3	TC3	**Channel 3 Terminal Count** 0 = Channel 3 terminal count not detected 1 = Channel 3 terminal count detected
2	TC2	**Channel 2 Terminal Count** 0 = Channel 2 terminal count not detected 1 = Channel 2 terminal count detected
1	TC1	**Channel 1 Terminal Count** 0 = Channel 1 terminal count not detected 1 = Channel 1 terminal count detected
0	TC0	**Channel 0 Terminal Count** 0 = Channel 0 terminal count not detected 1 = Channel 0 terminal count detected

Programming Notes

Bits 3–0 of this register are reset when read. Any read from this register (SLDMASTA) clears bits 3–0.

Slave DMA Channel 0–3 Control (SLDMACTL)

Direct-Mapped I/O Address 0008h

	7	6	5	4	3	2	1	0
Bit	DAKSEN	DRQSEN	WRTSEL	PRITYPE	COMPTIM	DMA_DIS	Reserved	
Reset	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	RSV	

Register Description

This register provides the control function for Channels 0–3.

Bit Definitions

Bit	Name	Function
7	DAKSEN	**Internal $\overline{\text{dackx}}$ Sense** In a discrete DMA controller, this bit controls the polarity of all $\overline{\text{dackx}}$ outputs from the slave DMA controller: 0 = Asserted Low 1 = Asserted High System logic external to the DMA controller expects the DMA controller to drive active Low $\overline{\text{dackx}}$ outputs. This bit must be written to 0 for proper system operation.
6	DRQSEN	**Internal drqx Sense** In a discrete DMA controller, this bit controls the polarity of all drqx inputs to the slave DMA controller: 0 = Asserted High 1 = Asserted Low System logic external to the DMA controller expects the DMA controller to respond to active High drqx inputs. This bit must be written to 0 for proper system operation.
5	WRTSEL	**Write Selection Control** 0 = Late write selection. 1 = Extended (early) write selection. Write command signals ($\overline{\text{GPIOWR}}$ and $\overline{\text{GPMEMWR}}$) are asserted one clock early. Enabling this feature results in timing changes on the GP bus that can violate the ISA specification. This bit has no effect when the COMPTIM bit is 1.
4	PRITYPE	**Priority Type** 0 = Fixed priority 1 = Rotating priority
3	COMPTIM	**Compressed Timing** 0 = Normal timing. 1 = Compressed timing. Read command signals ($\overline{\text{GPIORD}}$ and $\overline{\text{GPMEMRD}}$) have a one clock pulse width. Enabling this feature results in timing changes on the GP bus that can violate the ISA specification.
2	DMA_DIS	**Disable DMA Controller** 0 = DMA requests are enabled. 1 = DMA requests are ignored but DMA registers are available to the CPU. The DMA controller should be disabled prior to programming it in order to prevent unintended transfers from occurring during the DMA controller programming operation. Care should also be taken to ensure that the DMA controller is idle before disabling it. If the DMA controller is performing a transfer when software disables it, abnormal system operation can occur.

Bit	Name	Function
1–0	Reserved	**Reserved** This bit field should be written to 0 for normal system operation.

Programming Notes

If DMA Channel 0, 1, 2, or 3 is used by an internal UART, the DAKSEN, DRQSEN, WRTSEL, and COMPTIM bit fields must be 0 (the default).

Slave Software DRQ(n) Request (SLDMASWREQ)

Direct-Mapped I/O Address 0009h

	7	6	5	4	3	2	1	0
Bit	Reserved					REQDMA	REQSEL[1–0]	
Reset	0	0	0	0	0	x	x	x
R/W	RSV					W	W	

Register Description

This register is used to initiate a software DMA request for one of Channels 0–3.

Bit Definitions

Bit	Name	Function
7–3	Reserved	**Reserved** This bit field should be written to 0 for normal system operation.
2	REQDMA	**Software DMA Request** 0 = Clear the request bit for the channel selected by the REQSEL bit field. 1 = Set the request bit for the channel selected by the REQSEL bit field. The request bit is cleared internally after the corresponding channel has reached the end of its transfer count.
1–0	REQSEL[1–0]	**DMA Channel Select** This bit field selects the DMA channel that is to latch the REQDMA bit internally to assert or deassert a DMA request via software: 00 = Select Channel 0 for internal DMA request per the REQDMA bit. 01 = Select Channel 1 for internal DMA request per the REQDMA bit. 10 = Select Channel 2 for internal DMA request per the REQDMA bit. 11 = Select Channel 3 for internal DMA request per the REQDMA bit.

Programming Notes

Slave DMA Channel 0–3 Mask (SLDMAMSK)

Direct-Mapped I/O Address 000Ah

	7	6	5	4	3	2	1	0
Bit	Reserved					CHMASK	MSKSEL[1–0]	
Reset	0	0	0	0	0	x	x	x
R/W	RSV					W	W	

Register Description

This register provides a mask or unmask capability for the DMA request signal to each of Channels 0–3.

Bit Definitions

Bit	Name	Function
7–3	Reserved	**Reserved** This bit field should be written to 0 for normal system operation.
2	CHMASK	**DMA Channel Mask** 0 = Clear the mask bit for the channel selected by the MSKSEL bit field. 1 = Set the mask bit for the channel selected by the MSKSEL bit field. If the AINIT bit is 0 in the SLDMAMODE register (see page 11-55), the mask bit is set internally when the corresponding channel reaches the end of its transfer count. If the AINIT bit is 1, the mask bit remains clear when the transfer ends.
1–0	MSKSEL[1–0]	**DMA Channel Mask Select** This bit field selects the DMA channel that is to latch the CHMASK bit internally to mask or unmask the DMA request signal into the specified DMA channel: 00 = Mask or unmask DMA Channel 0 per the CHMASK bit. 01 = Mask or unmask DMA Channel 1 per the CHMASK bit. 10 = Mask or unmask DMA Channel 2 per the CHMASK bit. 11 = Mask or unmask DMA Channel 3 per the CHMASK bit.

Programming Notes

The same DMA channel masks can be controlled via DMA registers SLDMAMSK, SLDMAMSKRST (see page 11-60), and SLDMAGENMSK (see page 11-61).

Before masking an active DMA channel, software must ensure that the DMA request is deasserted. Masking an active channel while it is being granted might cause the system to hang.

Slave DMA Channel 0–3 Mode (SLDMAMODE)

Direct-Mapped I/O Address 000Bh

	7	6	5	4	3	2	1	0
Bit	TRNMOD[1–0]		ADDDEC	AINIT	OPSEL[1–0]		MODSEL[1–0]	
Reset	x	x	x	x	x	x	x	x
R/W	W		W	W	W		W	

Register Description

This register indicates the transfer mode for Channels 0–3.

Bit Definitions

Bit	Name	Function
7–6	TRNMOD[1–0]	**Transfer Mode** This bit field selects the transfer mode for the channel selected by the MODSEL bit field. 00 = Demand transfer mode 01 = Single transfer mode 10 = Block transfer mode 11 = Cascade mode. (Only Channel 4 should be programmed for cascade mode. All other channels should be programmed for one of the other modes).
5	ADDDEC	**Address Decrement** This bit field selects increment or decrement counting for the channel selected by the MODSEL bit field. 0 = Increment the DMA memory address after each transfer. 1 = Decrement the DMA memory address after each transfer.
4	AINIT	**Automatic Initialization Control** This bit field enables automatic initialization for the channel selected by the MODSEL bit field. 0 = Automatic initialization is disabled. 1 = Automatic initialization is enabled. Automatic initialization causes the channel's base address and transfer count registers to be restored to the values they contained prior to the DMA transfer when the transfer count ends. The channel is then ready to perform another DMA transfer without processor intervention as soon as the next DMA request is detected. Automatic initialization must be disabled when buffer chaining mode is used; otherwise unexpected results may occur.
3–2	OPSEL[1–0]	**Operation Select** This bit field selects the DMA operation for the channel selected by the MODSEL bit field. 00 = Verify mode. The DMA controller acts normally except that no I/O or memory commands are generated, and no data is transferred. 01 = Write transfer. Data is transferred from a DMA-capable I/O or memory-mapped device into system memory. 10 = Read transfer. Data is transferred from system memory to a DMA-capable I/O or memory-mapped device. 11 = Reserved.

Bit	Name	Function
1–0	MODSEL[1–0]	**DMA Channel Select** This bit field selects the channel that is to internally latch the other bits written to this register: 00 = Select Channel 0 01 = Select Channel 1 10 = Select Channel 2 11 = Select Channel 3

Programming Notes

Slave DMA Clear Byte Pointer (SLDMACBP)

Direct-Mapped I/O Address 000Ch

	7	6	5	4	3	2	1	0
Bit	SLAVE_CBP[7–0]							
Reset	x	x	x	x	x	x	x	x
R/W	W!							

Register Description

This register channel provides a mechanism to adjust the byte pointer to the low byte of the memory address and transfer count registers for Channels 0–3.

Bit Definitions

Bit	Name	Function
7–0	SLAVE_CBP [7–0]	**Slave DMA Clear Byte Pointer** The DMA controller's 16-bit memory address and terminal count registers are accessed by writing or reading consecutive 8-bit values to the same direct-mapped I/O address. A single byte pointer is used across the slave DMA controller to determine which byte is accessed if any of these 16-bit registers is read or written. Any access to one of these registers toggles the byte pointer between the low and high bytes. A write of any data to this bit field (SLAVE_CBP) clears the byte pointer so that the next memory address or transfer count register access is to the low byte.

Programming Notes

Slave DMA Controller Reset (SLDMARST)

Direct-Mapped I/O Address 000Dh

	7	6	5	4	3	2	1	0
Bit	SLAVE_RST[7–0]							
Reset	0	0	0	0	0	0	0	0
R/W	W!							

Register Description

This register provides a reset mechanism for Channels 0–3.

Bit Definitions

Bit	Name	Function
7–0	SLAVE_RST [7–0]	**Slave DMA Controller Reset** A write of any data to this address resets the DMA controller to the same state as a system reset.

Programming Notes

Slave DMA Controller Temporary (SLDMATMP)

Direct-Mapped I/O Address 000Dh

	7	6	5	4	3	2	1	0
Bit	SLAVE_TMP[7–0]							
Reset	0	0	0	0	0	0	0	0
R/W	R							

Register Description

This register has no real use in the ÉlanSC520 microcontroller. It is included for compatibility only.

Bit Definitions

Bit	Name	Function
7–0	SLAVE_TMP [7–0]	**Slave DMA Controller Temporary Register** In a discrete DMA controller, this bit field is used as a temporary storage buffer when doing memory-to-memory DMA. Memory-to-memory DMA transfers are not supported in the ÉlanSC520 microcontroller, so this register is included for compatibility reasons only.

Programming Notes

Slave DMA Mask Reset (SLDMAMSKRST)

Direct-Mapped I/O Address 000Eh

	7	6	5	4	3	2	1	0
Bit	SLAVE_MSK_RST[7–0]							
Reset	x	x	x	x	x	x	x	x
R/W	W!							

Register Description

This register provides a mechanism to reset the SLDMAGENMSK register (see page 11-61).

Bit Definitions

Bit	Name	Function
7–0	SLAVE_MSK_RST[7–0]	**Slave DMA Reset Mask** Writing any data to this I/O address resets the SLDMAGENMSK register (see page 11-61), thereby activating the four slave DMA channels.

Programming Notes

The same DMA channel masks can be controlled via DMA registers SLDMAMSK (see page 11-54), SLDMAMSKRST, and SLDMAGENMSK (see page 11-61).

Before masking an active DMA channel, software must ensure that the DMA request is deasserted. Masking an active channel while it is being granted might cause the system to hang.

Slave DMA General Mask (SLDMAGENMSK)

Direct-Mapped I/O Address 000Fh

	7	6	5	4	3	2	1	0
Bit	Reserved				CH3_DIS	CH2_DIS	CH1_DIS	CH0_DIS
Reset	0	0	0	0	1	1	1	1
R/W	RSV				W	W	W	W

Register Description

This register provides a mechanism to mask or unmask the DMA request signal to each of Channels 0–3.

Bit Definitions

Bit	Name	Function
7–4	Reserved	**Reserved** This bit field should be written to 0 for normal system operation.
3	CH3_DIS	**DMA Channel 3 Mask** 0 = Enable DMA Channel 3 for servicing DMA requests. 1 = Disable DMA Channel 3 from servicing DMA requests.
2	CH2_DIS	**DMA Channel 2 Mask** 0 = Enable DMA Channel 2 for servicing DMA requests. 1 = Disable DMA Channel 2 from servicing DMA requests.
1	CH1_DIS	**DMA Channel 1 Mask** 0 = Enable DMA Channel 1 for servicing DMA requests. 1 = Disable DMA Channel 1 from servicing DMA requests.
0	CH0_DIS	**DMA Channel 0 Mask** 0 = Enable DMA Channel 0 for servicing DMA requests. 1 = Disable DMA Channel 0 from servicing DMA requests.

Programming Notes

The same DMA channel masks can be controlled via DMA registers SLDMAMSK (see page 11-54), SLDMAMSKRST (see page 11-60), and SLDMAGENMSK.

Before masking an active DMA channel, software must ensure that the DMA request is deasserted. Masking an active channel while it is being granted might cause the system to hang.

General 0 (GPDMAGR0)

Direct-Mapped I/O Address 0080h

	7	6	5	4	3	2	1	0
Bit	PORT80[7–0]							
Reset	0	0	0	0	0	0	0	0
R/W	R/W							

Register Description

This is a general-purpose register.

Bit Definitions

Bit	Name	Function
7–0	PORT80[7–0]	**General-Purpose R/W Register** Writes to this bit field are stored internally and propagated to the GP bus. Reads from this bit field return the internal register value, but are not propagated to the GP bus.

Programming Notes

In a discrete DMA controller, this register would be the DMA Channel 4 Page register, but in PC/AT-compatible systems DMA Channel 4 is used for the cascade function, so this register is not used by the DMA subsystem.

In a PC/AT-compatible system, this I/O address (Port 0080h) is typically used to send BIOS Power-on Self Test (POST) codes to the ISA bus where special hardware can be used to decode the address and display the POST codes. In the ÉlanSC520 microcontroller, writes to this register are propagated to the GP bus to allow PC/AT-compatible operation. Reads from this register return the internally-stored value only.

Slave DMA Channel 2 Page (GPDMA2PG)

Direct-Mapped I/O Address 0081h

	7	6	5	4	3	2	1	0
Bit	DMA2MAR[23–16]							
Reset	0	0	0	0	0	0	0	0
R/W	R/W							

Register Description

This register provides bits 23–16 of the memory address for Channel 2.

Bit Definitions

Bit	Name	Function
7–0	DMA2MAR [23–16]	**DMA Channel 2 Memory Address Bits [23–16]** This bit field is used with the values in the GPDMA2MAR register (see page 11-46) and the GPDMAEXTPG2 register (see page 11-12) to generate DMA address bits 27–0.

Programming Notes

Slave DMA Channel 3 Page (GPDMA3PG)

Direct-Mapped I/O Address 0082h

	7	6	5	4	3	2	1	0
Bit	DMA3MAR[23–16]							
Reset	0	0	0	0	0	0	0	0
R/W	R/W!							

Register Description

This register provides bits 23–16 of the memory address for Channel 3.

Bit Definitions

Bit	Name	Function
7–0	DMA3MAR [23–16]	**DMA Channel 3 Memory Address Bits [23–16]** For 8-bit transfers, the value in this bit field is used with the values in the GPDMA3MAR register (see page 11-48) and the GPDMAEXTPG3 register (see page 11-13) to generate DMA address bits 27–0. For 16-bit transfers (enhanced mode only), bits 7–1 of this bit field hold address bits 23–17, and address bit 16 is located in the GPDMA3MAR register (see page 11-48). Bit 0 of this bit field is not used in 16-bit DMA operation. The bit 0 value read back is always the last value written, but the bit is not applied to the system address unless 8-bit operation is selected.

Programming Notes

In enhanced mode, this channel can be programmed for 16-bit DMA transfers (see the descriptions for GPDMACTL register bits CH3_ALT_SIZE and ENH_MODE_ENB, on page 11-4).

In the enhanced mode, this register is updated during DMA cycles if the DMA addresses cross the 64-Kbyte boundary for 8-bit transfers, or cross the 128-Kbyte boundary for 16-bit transfers.

Slave DMA Channel 1 Page (GPDMA1PG)

Direct-Mapped I/O Address 0083h

	7	6	5	4	3	2	1	0
Bit	DMA1MAR[23–16]							
Reset	0	0	0	0	0	0	0	0
R/W	R/W							

Register Description

This register provides bits 23–16 of the memory address for Channel 1.

Bit Definitions

Bit	Name	Function
7–0	DMA1MAR [23–16]	**DMA Channel 1 Memory Address Bits [23–16]** This bit field is used with the values in the GPDMA1MAR register (see page 11-44) and the GPDMAEXTPG1 register (see page 11-11) to generate DMA address bits 27–0.

Programming Notes

General 1 (GPDMAGR1)

Direct-Mapped I/O Address 0084h

	7	6	5	4	3	2	1	0
Bit	PORT84[7–0]							
Reset	0	0	0	0	0	0	0	0
R/W	R/W							

Register Description

This is a general-purpose register.

Bit Definitions

Bit	Name	Function
7–0	PORT84[7–0]	**General-Purpose R/W Register** Writes to this bit field are stored internally and propagated to the GP bus. Reads from this bit field return the internal register value, but are not propagated to the GP bus.

Programming Notes

General 2 (GPDMAGR2)

Direct-Mapped I/O Address 0085h

	7	6	5	4	3	2	1	0
Bit	PORT85[7–0]							
Reset	0	0	0	0	0	0	0	0
R/W	R/W							

Register Description

This is a general-purpose register.

Bit Definitions

Bit	Name	Function
7–0	PORT85[7–0]	**General-Purpose R/W Register** Writes to this bit field are stored internally and propagated to the GP bus. Reads from this bit field return the internal register value, but are not propagated to the GP bus.

Programming Notes

General 3 (GPDMAGR3)

Direct-Mapped I/O Address 0086h

	7	6	5	4	3	2	1	0
Bit	PORT86[7–0]							
Reset	0	0	0	0	0	0	0	0
R/W	R/W							

Register Description

This is a general-purpose register.

Bit Definitions

Bit	Name	Function
7–0	PORT86[7–0]	**General-Purpose R/W Register** Writes to this bit field are stored internally and propagated to the GP bus. Reads from this bit field return the internal register value, but are not propagated to the GP bus.

Programming Notes

Slave DMA Channel 0 Page (GPDMA0PG)

Direct-Mapped I/O Address 0087h

	7	6	5	4	3	2	1	0
Bit	DMA0MAR[23–16]							
Reset	0	0	0	0	0	0	0	0
R/W	R/W							

Register Description

This register provides bits 23–16 of the memory address for Channel 0.

Bit Definitions

Bit	Name	Function
7–0	DMA0MAR [23–16]	**DMA Channel 0 Memory Address Bits [23–16]** This bit field is used with the values in the GPDMA0MAR register (see page 11-42) and the GPDMAEXTPG0 register (see page 11-10) to generate DMA address bits 27–0.

Programming Notes

General 4 (GPDMAGR4)

Direct-Mapped I/O Address 0088h

	7	6	5	4	3	2	1	0
Bit	PORT88[7–0]							
Reset	0	0	0	0	0	0	0	0
R/W	R/W							

Register Description

This is a general-purpose register.

Bit Definitions

Bit	Name	Function
7–0	PORT88[7–0]	**General-Purpose R/W Register** Writes to this bit field are stored internally and propagated to the GP bus. Reads from this bit field return the internal register value, but are not propagated to the GP bus.

Programming Notes

Master DMA Channel 6 Page (GPDMA6PG)

Direct-Mapped I/O Address 0089h

	7	6	5	4	3	2	1	0
Bit	DMA6MAR[23–16]							
Reset	0	0	0	0	0	0	0	0
R/W	R/W!							

Register Description

This register provides bits 23–16 of the memory address for Channel 6.

Bit Definitions

Bit	Name	Function
7–1	DMA6MAR [23–16]	**DMA Channel 6 Memory Address Bits [23–16]** For 8-bit transfers (enhanced mode only), the value in this bit field is used with the values in the GPDMA6MAR register (see page 11-82) and the GPDMAEXTPG6 register (see page 11-15) to generate DMA address bits 27–0. For 16-bit transfers, bits 7–1 of this bit field hold address bits 23–17, and address bit 16 is located in the GPDMA6MAR register (see page 11-82). Bit 0 of this bit field is not used in 16-bit DMA operation. The bit 0 value read back is always the last value written, but the bit is not applied to the system address unless 8-bit operation is selected.

Programming Notes

In enhanced mode, this channel can be programmed for 8-bit DMA transfers (see the descriptions for GPDMACTL register bits CH6_ALT_SIZE and ENH_MODE_ENB, on page 11-4).

In enhanced mode, this register is updated during DMA cycles if the DMA addresses cross the 64-Kbyte boundary for 8-bit transfers, or cross the 128-Kbyte boundary for 16-bit transfers.

Master DMA Channel 7 Page (GPDMA7PG)

Direct-Mapped I/O Address 008Ah

	7	6	5	4	3	2	1	0
Bit	DMA7MAR[23–16]							
Reset	0	0	0	0	0	0	0	0
R/W	R/W!							

Register Description

This register provides bits 23–16 of the memory address for Channel 7.

Bit Definitions

Bit	Name	Function
7–1	DMA7MAR [23–16]	**DMA Channel 7 Memory Address Bits [23–17]** For 8-bit transfers (enhanced mode only), the value in this bit field is used with the values in the GPDMA7MAR register (see page 11-84) and the GPDMAEXTPG7 register (see page 11-16) to generate DMA address bits 27–0. For 16-bit transfers, bits 7–1 of this bit field hold address bits 23–17, and address bit 16 is located in the GPDMA7MAR register (see page 11-84). Bit 0 of this bit field is not used in 16-bit DMA operation. The bit 0 value read back is always the last value written, but the bit is not applied to the system address unless 8-bit operation is selected.

Programming Notes

In enhanced mode, this channel can be programmed for 8-bit DMA transfers (see the descriptions for GPDMACTL register bits CH7_ALT_SIZE and ENH_MODE_ENB, on page 11-4).

In enhanced mode, this register is updated during DMA cycles if the DMA addresses cross the 64-Kbyte boundary for 8-bit transfers, or cross the 128-Kbyte boundary for 16-bit transfers.

Master DMA Channel 5 Page (GPDMA5PG)

Direct-Mapped I/O Address 008Bh

	7	6	5	4	3	2	1	0
Bit	DMA5MAR[23–16]							
Reset	0	0	0	0	0	0	0	0
R/W	R/W!							

Register Description

This register provides bits 23–16 of the memory address for Channel 5.

Bit Definitions

Bit	Name	Function
7–1	DMA5MAR [23–16]	**DMA Channel 5 Memory Address Bits [23–16]** For 8-bit transfers (enhanced mode only), the value in this bit field is used with the values in the GPDMA5MAR register (see page 11-80) and the GPDMAEXTPG5 register (see page 11-14) to generate DMA address bits 27–0. For 16-bit transfers, bits 7–1 of this bit field hold address bits 23–17, and address bit 16 is located in the GPDMA5MAR register (see page 11-80). Bit 0 of this bit field is not used in 16-bit DMA operation. The bit 0 value read back is always the last value written, but the bit is not applied to the system address unless 8-bit operation is selected.

Programming Notes

In enhanced mode, this channel can be programmed for 8-bit DMA transfers (see the descriptions for GPDMACTL register bits CH5_ALT_SIZE and ENH_MODE_ENB, on page 11-4).

In enhanced mode, this register is updated during DMA cycles if the DMA addresses cross the 64-Kbyte boundary for 8-bit transfers, or cross the 128-Kbyte boundary for 16-bit transfers.

General 5 (GPDMAGR5)

Direct-Mapped I/O Address 008Ch

	7	6	5	4	3	2	1	0
Bit	PORT8C[7–0]							
Reset	0	0	0	0	0	0	0	0
R/W	R/W							

Register Description

This is a general-purpose register.

Bit Definitions

Bit	Name	Function
7–0	PORT8C[7–0]	**General-Purpose R/W Register** Writes to this bit field are stored internally and propagated to the GP bus. Reads from this bit field return the internal register value, but are not propagated to the GP bus.

Programming Notes

General 6 (GPDMAGR6)

Direct-Mapped I/O Address 008Dh

	7	6	5	4	3	2	1	0
Bit	PORT8D[7–0]							
Reset	0	0	0	0	0	0	0	0
R/W	R/W							

Register Description

This is a general-purpose register.

Bit Definitions

Bit	Name	Function
7–0	PORT8D[7–0]	**General-Purpose R/W Register** Writes to this bit field are stored internally and propagated to the GP bus. Reads from this bit field return the internal register value, but are not propagated to the GP bus.

Programming Notes

General 7 (GPDMAGR7)

Direct-Mapped I/O Address 008Eh

	7	6	5	4	3	2	1	0
Bit	PORT8E[7–0]							
Reset	0	0	0	0	0	0	0	0
R/W	R/W							

Register Description

This is a general-purpose register.

Bit Definitions

Bit	Name	Function
7–0	PORT8E[7–0]	**General-Purpose R/W Register** Writes to this bit field are stored internally and propagated to the GP bus. Reads from this bit field return the internal register value, but are not propagated to the GP bus.

Programming Notes

General 8 (GPDMAGR8)

Direct-Mapped I/O Address 008Fh

	7	6	5	4	3	2	1	0
Bit	PORT8F[7–0]							
Reset	0	0	0	0	0	0	0	0
R/W	R/W							

Register Description

This is a general-purpose register.

Bit Definitions

Bit	Name	Function
7–0	PORT8F[7–0]	**General-Purpose R/W Register** Writes to this bit field are stored internally. Reads from this bit field return the internal register value.

Programming Notes

This general register is slightly different from the other direct-mapped general registers. No GP bus cycles are generated on writes to this I/O address (Port 008Fh).

Master DMA Channel 4 Memory Address (GPDMA4MAR)

Direct-Mapped
I/O Address 00C0h

	7	6	5	4	3	2	1	0
Bit	DMA4MAR[16–1]							
Reset	x	x	x	x	x	x	x	x
R/W	R/W!							

Register Description

Because Channel 4 is used to cascade the slave DMA controller to the master DMA controller, this register has no real use.

Bit Definitions

Bit	Name	Function
7–0	DMA4MAR [16–1]	**DMA Channel 4 Memory Address** In a discrete DMA controller, this bit field holds the lower 16 bits of memory address for DMA Channel 4. Because the drq and dack internal signals for this DMA channel are used to cascade to the slave DMA controller in a PC/AT-compatible system, DMA Channel 4 is not used directly for DMA transfers. So this register has no real function; it is documented only for completeness. For the same reason, there is no corresponding page register for DMA Channel 4.

Programming Notes

Master DMA Channel 4 Transfer Count (GPDMA4TC)

Direct-Mapped I/O Address 00C2h

	7	6	5	4	3	2	1	0
Bit	DMA4TC[15–0]							
Reset	x	x	x	x	x	x	x	x
R/W	R/W!							

Register Description

Because Channel 4 is used to cascade the slave DMA controller to the master DMA controller, this register has no real use.

Bit Definitions

Bit	Name	Function
7–0	DMA4TC [15–0]	**DMA Channel 4 Transfer Count** In a discrete DMA controller, this bit field holds the 16-bit transfer count for DMA Channel 4. Because the drq and $\overline{\text{dack}}$ internal signals for this DMA channel are used to cascade to the slave DMA controller in a PC/AT-compatible system, DMA Channel 4 is not used directly for DMA transfers. So this register has no real function; it is documented only for completeness.

Programming Notes

Master DMA Channel 5 Memory Address (GPDMA5MAR) — Direct-Mapped I/O Address 00C4h

	7	6	5	4	3	2	1	0
Bit	DMA5MAR[16–1]							
Reset	x	x	x	x	x	x	x	x
R/W	R/W!							

Register Description

This register contains bits 16–1 of the memory address for Channel 5 during DMA operation.

Bit Definitions

Bit	Name	Function
7–0	DMA5MAR [16–1]	**Lower 16 Bits of DMA Channel 5 Memory Address** This 8-bit field is used in two successive I/O accesses to read or write the channel's memory address bits 16–1 for 16-bit DMA transfers. Bits 8–1 of the channel's memory address can be read from or written to this bit field immediately after a write to the MSTDMACBP register (see page 11-93). Bits 16–9 of the channel's memory address can be read from or written to this bit field immediately after memory address bits 8–1 are read from or written to this bit field.

Programming Notes

To ensure that the lower byte of this register (GPDMA5MAR) is always accessed first, software should precede any access to this register with a write to the MSTDMACBP register (see page 11-93) to clear the master DMA byte pointer.

The value in this register (GPDMA5MAR) is used with the values in the GPDMA5PG register (see page 11-73) and the GPDMAEXTPG5 register (see page 11-14) to generate DMA address bits 27–0.

By default, this channel is set up for PC/AT compatibility (16-bit DMA transfers on the master DMA controller). For 16-bit transfers, this register (GPDMA5MAR) holds address bits 16–1 and address bit 0 is always 0 (i.e., the 16-bit transfers are word-aligned). Because of this, software must load this register with the desired memory address divided by 2 for 16-bit transfers.

In enhanced mode, this channel can be programmed for 8-bit DMA transfers (see the descriptions for GPDMACTL register bits CH5_ALT_SIZE and ENH_MODE_ENB, on page 11-4). For 8-bit transfers, this register (GPDMA5MAR) holds address bits 15–0, and address bit 16 is controlled via the GPDMA5PG register (see page 11-73).

Master DMA Channel 5 Transfer Count (GPDMA5TC)

Direct-Mapped I/O Address 00C6h

	7	6	5	4	3	2	1	0
Bit	DMA5TC[15–0]							
Reset	x	x	x	x	x	x	x	x
R/W	R/W!							

Register Description

This register contains bits 15–0 of the transfer count for Channel 5 during DMA operation.

Bit Definitions

Bit	Name	Function
7–0	DMA5TC [15–0]	**DMA Channel 5 Transfer Count (16-Bit Register)** This 8-bit field is used in two successive I/O accesses to read or write the channel's transfer count bits 15–0. Bits 7–0 of the channel's transfer count can be read from or written to this bit field immediately after a write to the MSTDMACBP register (see page 11-93). Bits 15–8 of the channel's transfer count can be read from or written to this bit field immediately after transfer count bits 7–0 are read from or written to this bit field. The actual number of transfers is one more than the programmed transfer count value.

Programming Notes

To ensure that the lower byte of this register (GPDMA5TC) is always accessed first, software should precede any access to this register with a write to the MSTDMACBP register (see page 11-93) to clear the master DMA byte pointer.

By default, this channel is set up for PC/AT compatibility (16-bit DMA transfers on the master DMA controller). For 16-bit transfers, each transfer is two bytes, so a transfer count of FFFFh results in a transfer of 128 Kbytes.

The value in this register (GPDMA5TC) can be used with the value in the GPDMAEXTTC5 register (see page 11-18) to allow counts of up to 16 M (16,777,216) transfers.

In PCI bus 2.2-compliant designs, software must limit the length of GP bus DMA demand- or block-mode transfers. Very large transfers could cause the PCI Host Bridge target controller to violate the 10-µs memory write maximum completion time limit set in the *PCI Local Bus Specification,* Revision 2.2.

Master DMA Channel 6 Memory Address (GPDMA6MAR)

Direct-Mapped
I/O Address 00C8h

	7	6	5	4	3	2	1	0
Bit	DMA6MAR[16–1]							
Reset	x	x	x	x	x	x	x	x
R/W	R/W!							

Register Description

This register contains bits 16–1 of the memory address for Channel 6 during DMA operation.

Bit Definitions

Bit	Name	Function
7–0	DMA6MAR [16–1]	**Lower 16 Bits of DMA Channel 6 Memory Address** This 8-bit field is used in two successive I/O accesses to read or write the channel's memory address bits 16–1 for 16-bit DMA transfers. Bits 8–1 of the channel's memory address can be read from or written to this bit field immediately after a write to the MSTDMACBP register (see page 11-93). Bits 16–9 of the channel's memory address can be read from or written to this bit field immediately after memory address bits 8–1 are read from or written to this bit field.

Programming Notes

To ensure that the lower byte of this register (GPDMA6MAR) is always accessed first, software should precede any access to this register with a write to the MSTDMACBP register (see page 11-93) to clear the master DMA byte pointer.

The value in this register (GPDMA6MAR) is used with the values in the GPDMA6PG register (see page 11-71) and the GPDMAEXTPG6 register (see page 11-15) to generate DMA address bits 27–0.

By default, this channel is set up for PC/AT compatibility (16-bit DMA transfers on the master DMA controller). For 16-bit transfers, this register (GPDMA6MAR) holds address bits 16–1 and address bit 0 is always 0 (i.e., the 16-bit transfers are word-aligned). Because of this, software must load this register with the desired memory address divided by 2 for 16-bit transfers.

In enhanced mode, this channel can be programmed for 8-bit DMA transfers (see the descriptions for GPDMACTL register bits CH6_ALT_SIZE and ENH_MODE_ENB, on page 11-4). For 8-bit transfers, this register (GPDMA6MAR) holds address bits 15–0, and address bit 16 is controlled via the GPDMA6PG register (see page 11-71).

Master DMA Channel 6 Transfer Count (GPDMA6TC)

Direct-Mapped I/O Address 00CAh

	7	6	5	4	3	2	1	0
Bit	DMA6TC[15–0]							
Reset	x	x	x	x	x	x	x	x
R/W	R/W!							

Register Description

This register contains bits 15–0 of the transfer count for Channel 6 during DMA operation.

Bit Definitions

Bit	Name	Function
7–0	DMA6TC [15–0]	**DMA Channel 6 Transfer Count (16-Bit Register)** This 8-bit field is used in two successive I/O accesses to read or write the channel's transfer count bits 15–0. Bits 7–0 of the channel's transfer count can be read from or written to this bit field immediately after a write to the MSTDMACBP register (see page 11-93). Bits 15–8 of the channel's transfer count can be read from or written to this bit field immediately after transfer count bits 7–0 are read from or written to this bit field. The actual number of transfers is one more than the programmed transfer count value.

Programming Notes

To ensure that the lower byte of this register (GPDMA6TC) is always accessed first, software should precede any access to this register with a write to the MSTDMACBP register (see page 11-93) to clear the master DMA byte pointer.

By default, this channel is set up for PC/AT compatibility (16-bit DMA transfers on the master DMA controller). For 16-bit transfers, each transfer is two bytes, so a transfer count of FFFFh results in a transfer of 128 Kbytes.

The value in this register (GPDMA6TC) can be used with the value in the GPDMAEXTTC6 register (see page 11-19) to allow counts of up to 16 M (16,777,216) transfers.

In PCI bus 2.2-compliant designs, software must limit the length of GP bus DMA demand- or block-mode transfers. Very large transfers could cause the PCI Host Bridge target controller to violate the 10-µs memory write maximum completion time limit set in the *PCI Local Bus Specification,* Revision 2.2.

Master DMA Channel 7 Memory Address (GPDMA7MAR)

Direct-Mapped I/O Address 00CCh

	7	6	5	4	3	2	1	0
Bit	DMA7MAR[16–1]							
Reset	x	x	x	x	x	x	x	x
R/W	R/W!							

Register Description

This register contains bits 16–1 of the memory address for Channel 7 during DMA operation.

Bit Definitions

Bit	Name	Function
7–0	DMA7MAR [16–1]	**Lower 16 Bits of DMA Channel 7 Memory Address** This 8-bit field is used in two successive I/O accesses to read or write the channel's memory address bits 16–1 for 16-bit DMA transfers. Bits 8–1 of the channel's memory address can be read from or written to this bit field immediately after a write to the MSTDMACBP register (see page 11-93). Bits 16–9 of the channel's memory address can be read from or written to this bit field immediately after memory address bits 8–1 are read from or written to this bit field.

Programming Notes

To ensure that the lower byte of this register (GPDMA7MAR) is always accessed first, software should precede any access to this register with a write to the MSTDMACBP register (see page 11-93) to clear the master DMA byte pointer.

The value in this register (GPDMA7MAR) is used with the values in the GPDMA7PG register (see page 11-72) and the GPDMAEXTPG7 register (see page 11-16) to generate DMA address bits 27–0.

By default, this channel is set up for PC/AT compatibility (16-bit DMA transfers on the master DMA controller). For 16-bit transfers, this register (GPDMA7MAR) holds address bits 16–1 and address bit 0 is always 0 (i.e., the 16-bit transfers are word-aligned). Because of this, software must load this register with the desired memory address divided by 2 for 16-bit transfers.

In enhanced mode, this channel can be programmed for 8-bit DMA transfers (see the descriptions for GPDMACTL register bits CH7_ALT_SIZE and ENH_MODE_ENB, on page 11-4). For 8-bit transfers, this register (GPDMA7MAR) holds address bits 15–0, and address bit 16 is controlled via the GPDMA7PG register (see page 11-72).

Master DMA Channel 7 Transfer Count (GPDMA7TC)

Direct-Mapped I/O Address 00CEh

	7	6	5	4	3	2	1	0
Bit	DMA7TC[15–0]							
Reset	x	x	x	x	x	x	x	x
R/W	R/W!							

Register Description

This register contains bits 15–0 of the transfer count for Channel 7 during DMA operation.

Bit Definitions

Bit	Name	Function
7–0	DMA7TC [15–0]	**DMA Channel 7 Transfer Count (16-Bit Register)** This 8-bit field is used in two successive I/O accesses to read or write the channel's transfer count bits 15–0. Bits 7–0 of the channel's transfer count can be read from or written to this bit field immediately after a write to the MSTDMACBP register (see page 11-93). Bits 15–8 of the channel's transfer count can be read from or written to this bit field immediately after transfer count bits 7–0 are read from or written to this bit field. The actual number of transfers is one more than the programmed transfer count value.

Programming Notes

To ensure that the lower byte of this register (GPDMA7TC) is always accessed first, software should precede any access to this register with a write to the MSTDMACBP register (see page 11-93) to clear the master DMA byte pointer.

By default, this channel is set up for PC/AT compatibility (16-bit DMA transfers on the master DMA controller). For 16-bit transfers, each transfer is two bytes, so a transfer count of FFFFh results in a transfer of 128 Kbytes.

The value in this register (GPDMA7TC) can be used with the value in the GPDMAEXTTC7 register (see page 11-20) to allow counts of up to 16 M (16,777,216) transfers.

In PCI bus 2.2-compliant designs, software must limit the length of GP bus DMA demand- or block-mode transfers. Very large transfers could cause the PCI Host Bridge target controller to violate the 10-µs memory write maximum completion time limit set in the *PCI Local Bus Specification,* Revision 2.2.

Master DMA Channel 4–7 Status (MSTDMASTA)

Direct-Mapped
I/O Address 00D0h

	7	6	5	4	3	2	1	0
Bit	DMAR7	DMAR6	DMAR5	DMAR4	TC7	TC6	TC5	TC4
Reset	?	?	?	?	0	0	0	0
R/W	R	R	R	R	R!	R!	R!	R!

Register Description

This register indicates the request status and terminal count status for Channels 4–7.

Bit Definitions

Bit	Name	Function
7	DMAR7	**Channel 7 DMA Request** 0 = Channel 7 DMA request not pending 1 = Channel 7 DMA request pending
6	DMAR6	**Channel 6 DMA Request** 0 = Channel 6 DMA request not pending 1 = Channel 6 DMA request pending
5	DMAR5	**Channel 5 DMA Request** 0 = Channel 5 DMA request not pending 1 = Channel 5 DMA request pending
4	DMAR4	**Channel 4 DMA Request** 0 = Channel 4 DMA request not pending 1 = Channel 4 DMA request pending
3	TC7	**Channel 7 Terminal Count** 0 = Channel 7 terminal count not detected 1 = Channel 7 terminal count detected
2	TC6	**Channel 6 Terminal Count** 0 = Channel 6 terminal count not detected 1 = Channel 6 terminal count detected
1	TC5	**Channel 5 Terminal Count** 0 = Channel 5 terminal count not detected 1 = Channel 5 terminal count detected
0	TC4	**Channel 4 Terminal Count** 0 = Channel 4 terminal count not detected 1 = Channel 4 terminal count detected

Programming Notes

Bits 3–0 of this register are reset when read. Any read from this register (MSTDMASTA) clears bits 3–0.

Master DMA Channel 4–7 Control (MSTDMACTL)

Direct-Mapped I/O Address 00D0h

	7	6	5	4	3	2	1	0
Bit	DAKSEN	DRQSEN	WRTSEL	PRITYPE	COMPTIM	DMA_DIS	Reserved	
Reset	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	RSV	

Register Description

This register provides the control function for Channels 4–7.

Bit Definitions

Bit	Name	Function
7	DAKSEN	**Internal $\overline{\text{dackx}}$ Sense** In a discrete DMA controller, this bit controls the polarity of all $\overline{\text{dackx}}$ outputs from the slave DMA controller: 0 = Asserted Low 1 = Asserted High System logic external to the DMA controller expects the DMA controller to drive active Low $\overline{\text{dackx}}$ outputs. This bit must be written to '0b' for proper system operation.
6	DRQSEN	**Internal drqx Sense** In a discrete DMA controller, this bit controls the polarity of all drqx inputs to the slave DMA controller: 0 = Asserted High 1 = Asserted Low System logic external to the DMA controller expects the DMA controller to respond to active High drqx inputs. This bit must be written to '0b' for proper system operation.
5	WRTSEL	**Write Selection Control** 0 = Late write selection 1 = Extended (early) write selection. Write command signals ($\overline{\text{GPIOWR}}$ and $\overline{\text{GPMEMWR}}$) are asserted one clock early. Enabling this feature results in timing changes on the GP bus that can violate the ISA specification. This bit has no effect when the COMPTIM bit is 1.
4	PRITYPE	**Priority Type** 0 = Fixed priority 1 = Rotating priority
3	COMPTIM	**Compressed Timing** 0 = Normal timing. 1 = Compressed timing. Read command signals ($\overline{\text{GPIORD}}$ and $\overline{\text{GPMEMRD}}$) have a one clock pulse width. Enabling this feature results in timing changes on the GP bus that can violate the ISA specification.
2	DMA_DIS	**Disable DMA Controller** 0 = DMA requests are enabled. 1 = DMA requests are ignored but DMA registers are available to the CPU. The DMA controller should be disabled prior to programming it in order to prevent unintended transfers from occurring during the DMA controller programming operation. Care should also be taken to ensure that the DMA controller is idle before disabling it. If the DMA controller is performing a transfer when software disables it, abnormal system operation can occur.

Bit	Name	Function
1–0	Reserved	**Reserved** This bit field should be written to 0 for normal system operation.

Programming Notes

Master Software DRQ(n) Request (MSTDMASWREQ)

Direct-Mapped
I/O Address 00D2h

	7	6	5	4	3	2	1	0
Bit	Reserved					REQDMA	REQSEL[1–0]	
Reset	0	0	0	0	0	x	x	x
R/W	RSV					W	W	

Register Description

This register is used to initiate a software DMA request for one of Channels 4–7.

Bit Definitions

Bit	Name	Function
7–3	Reserved	**Reserved** This bit field should be written to 0 for normal system operation.
2	REQDMA	**Software DMA Request** 0 = Clear the request bit for the channel selected by the REQSEL bit field. 1 = Set the request bit for the channel selected by the REQSEL bit field. The request bit is cleared internally after the corresponding channel has reached the end of its transfer count.
1–0	REQSEL[1–0]	**DMA Channel Select** This bit field selects DMA channel that is to latch the REQDMA bit internally to assert or deassert a DMA request via software: 00 = Select Channel 4 for internal DMA request per the REQDMA bit. 01 = Select Channel 5 for internal DMA request per the REQDMA bit. 10 = Select Channel 6 for internal DMA request per the REQDMA bit. 11 = Select Channel 7 for internal DMA request per the REQDMA bit.

Programming Notes

Master DMA Channel 4–7 Mask (MSTDMAMSK)

Direct-Mapped I/O Address 00D4h

	7	6	5	4	3	2	1	0
Bit	Reserved					CHMASK	MSKSEL[1–0]	
Reset	0	0	0	0	0	x	x	x
R/W	RSV					W	W	

Register Description

This register provides a mask or unmask capability for the DMA request signal to each of Channels 4–7.

Bit Definitions

Bit	Name	Function
7–3	Reserved	**Reserved** This bit field should be written to 0 for normal system operation.
2	CHMASK	**DMA Channel Mask** 0 = Clear the mask bit for the channel selected by the MSKSEL bit field. 1 = Set the mask bit for the channel selected by the MSKSEL bit field. If the AINIT bit is 0 in the MSTDMAMODE register (see page 11-91), the mask bit is set internally when the corresponding channel reaches the end of its transfer count. If the AINIT bit is 1, the mask bit remains clear when the transfer ends.
1–0	MSKSEL[1–0]	**DMA Channel Mask Select** This bit field selects the DMA channel that is to latch the CHMASK bit internally to mask or unmask the DMA request signal into the specified DMA channel: 00 = Mask or unmask DMA Channel 4 per the CHMASK bit. 01 = Mask or unmask DMA Channel 5 per the CHMASK bit. 10 = Mask or unmask DMA Channel 6 per the CHMASK bit. 11 = Mask or unmask DMA Channel 7 per the CHMASK bit.

Programming Notes

The same DMA channel masks can be controlled via DMA registers MSTDMAMSK, MSTDMAMSKRST (see page 11-96), and MSTDMAGENMSK (see page 11-97).

Before masking an active DMA channel, software must ensure that the DMA request is deasserted. Masking an active channel while it is being granted might cause the system to hang.

Master DMA Channel 4–7 Mode (MSTDMAMODE)

Direct-Mapped I/O Address 00D6h

	7	6	5	4	3	2	1	0
Bit	TRNMOD[1–0]		ADDDEC	AINIT	OPSEL[1–0]		MODSEL[1–0]	
Reset	x	x	x	x	x	x	x	x
R/W	W		W	W	W		W	

Register Description

This register indicates the transfer mode for Channels 4–7.

Bit Definitions

Bit	Name	Function
7–6	TRNMOD[1–0]	**Transfer Mode** This bit field selects the transfer mode for the channel selected by the MODSEL bit field. 00 = Demand transfer mode 01 = Single transfer mode 10 = Block transfer mode 11 = Cascade mode. (Only Channel 4 should be programmed for cascade mode. All other channels should be programmed for one of the other modes).
5	ADDDEC	**Address Decrement** This bit field selects increment or decrement counting for the channel selected by the MODSEL bit field. 0 = Increment the DMA memory address after each transfer. 1 = Decrement the DMA memory address after each transfer.
4	AINIT	**Automatic Initialization Control** This bit field enables automatic initialization for the channel selected by the MODSEL bit field. 0 = Automatic initialization is disabled. 1 = Automatic initialization is enabled. Automatic initialization causes the channel's base address and transfer count registers to be restored to the values they contained prior to the DMA transfer when the transfer count ends. The channel is then ready to perform another DMA transfer without processor intervention as soon as the next DMA request is detected. Automatic initialization must be disabled when buffer chaining mode is used; otherwise unexpected results may occur.
3–2	OPSEL[1–0]	**Operation Select** This bit field selects the DMA operation for the channel selected by the MODSEL bit field. 00 = Verify mode. The DMA controller acts normally except that no I/O or memory commands are generated, and no data is transferred. 01 = Write transfer. Data is transferred from a DMA-capable I/O or memory-mapped device into system memory. 10 = Read transfer. Data is transferred from system memory to a DMA-capable I/O or memory-mapped device. 11 = Reserved.

Bit	Name	Function
1–0	MODSEL[1–0]	**DMA Channel Select** This bit field selects the channel that is to internally latch the other bits written to this register: 00 = Select Channel 4 01 = Select Channel 5 10 = Select Channel 6 11 = Select Channel 7

Programming Notes

Master DMA Clear Byte Pointer (MSTDMACBP)

Direct-Mapped I/O Address 00D8h

	7	6	5	4	3	2	1	0
Bit	MASTR_CBP[7–0]							
Reset	x	x	x	x	x	x	x	x
R/W	W!							

Register Description

This register channel provides a mechanism to adjust the byte pointer to the low byte of the memory address and transfer count registers for Channels 4–7.

Bit Definitions

Bit	Name	Function
7–0	MASTR_CBP [7–0]	**Master DMA Clear Byte Pointer** The DMA controller's 16-bit memory address and terminal count registers are accessed by writing or reading consecutive 8-bit values to the same direct-mapped I/O address. A single byte pointer is used across the master DMA controller to determine which byte is accessed if any of these 16-bit registers is read or written. Any access to one of these registers toggles the byte pointer between the low and high bytes. A write of any data to this bit field (MASTR_CBP) clears the byte pointer so that the next memory address or transfer count register access is to the low byte.

Programming Notes

Master DMA Controller Reset (MSTDMARST)

Direct-Mapped I/O Address 00DAh

	7	6	5	4	3	2	1	0
Bit	MASTR_RST[7–0]							
Reset	0	0	0	0	0	0	0	0
R/W	W!							

Register Description

This register provides a reset mechanism for Channels 4–7.

Bit Definitions

Bit	Name	Function
7–0	MASTR_RST [7–0]	**Master DMA Controller Reset** A write of any data to this address resets the DMA controller to the same state as a system reset.

Programming Notes

Master DMA Controller Temporary (MSTDMATMP)

Direct-Mapped
I/O Address 00DAh

	7	6	5	4	3	2	1	0
Bit	MASTR_TMP[7–0]							
Reset	0	0	0	0	0	0	0	0
R/W	R							

Register Description

This register has no real use in the ÉlanSC520 microcontroller. It is included for compatibility only.

Bit Definitions

Bit	Name	Function
7–0	MASTR_TMP [7–0]	**Master DMA Controller Temporary Register** In a discrete DMA controller, this bit field is used as a temporary storage buffer when doing memory-to-memory DMA. Memory-to-memory DMA transfers are not supported in the ÉlanSC520 microcontroller, so this register is included for compatibility reasons only.

Programming Notes

Master DMA Mask Reset (MSTDMAMSKRST)

Direct-Mapped I/O Address 00DCh

	7	6	5	4	3	2	1	0
Bit	MASTR_MSK_RST[7–0]							
Reset	x	x	x	x	x	x	x	x
R/W	W!							

Register Description

This register provides a mechanism to reset the MSTDMAGENMSK register (see page 11-97).

Bit Definitions

Bit	Name	Function
7–0	MASTR_MSK_RST[7–0]	**Master DMA Reset Mask** Writing any data to this I/O address resets the MSTDMAGENMSK register (see page 11-97), thereby activating the four slave DMA channels.

Programming Notes

The same DMA channel masks can be controlled via DMA registers MSTDMAMSK (see page 11-90), MSTDMAMSKRST, and MSTDMAGENMSK (see page 11-97).

Before masking an active DMA channel, software must ensure that the DMA request is deasserted. Masking an active channel while it is being granted might cause the system to hang.

Master DMA General Mask (MSTDMAGENMSK)

Direct-Mapped I/O Address 00DEh

	7	6	5	4	3	2	1	0
Bit	Reserved				CH7_DIS	CH6_DIS	CH5_DIS	CH4_DIS
Reset	0	0	0	0	1	1	1	1
R/W	RSV				W	W	W	W

Register Description

This register provides a mechanism to mask or unmask the DMA request signal to each of Channels 4–7.

Bit Definitions

Bit	Name	Function
7–4	Reserved	**Reserved** This bit field should be written to 0 for normal system operation.
3	CH7_DIS	**DMA Channel 7 Mask** 0 = Enable DMA Channel 7 for servicing DMA requests. 1 = Disable DMA Channel 7 from servicing DMA requests.
2	CH6_DIS	**DMA Channel 6 Mask** 0 = Enable DMA Channel 6 for servicing DMA requests. 1 = Disable DMA Channel 6 from servicing DMA requests.
1	CH5_DIS	**DMA Channel 5 Mask** 0 = Enable DMA Channel 5 for servicing DMA requests. 1 = Disable DMA Channel 5 from servicing DMA requests.
0	CH4_DIS	**DMA Channel 4 Mask** 0 = Enable DMA Channel 4 for servicing DMA requests. 1 = Disable DMA Channel 4 from servicing DMA requests.

Programming Notes

The same DMA channel masks can be controlled via DMA registers MSTDMAMSK (see page 11-90), MSTDMAMSKRST (see page 11-96), and MSTDMAGENMSK.

Before masking an active DMA channel, software must ensure that the DMA request is deasserted. Masking an active channel while it is being granted might cause the system to hang.

CHAPTER

12 PROGRAMMABLE INTERRUPT CONTROLLER REGISTERS

12.1 OVERVIEW

This chapter describes the programmable interrupt controller (PIC) registers of the ÉlanSC520 microcontroller.

The ÉlanSC520 microcontroller's programmable interrupt controller (PIC) consists of three industry-standard controllers, integrated with a highly programmable interrupt router.

The PIC register set includes two groups of registers:

- 39 memory-mapped configuration region (MMCR) registers are used to configure and control PIC functions specific to the ÉlanSC520 microcontroller.
- 28 direct-mapped I/O registers are used for industry-standard PIC configuration, control, and status functions.

See the *Élan™SC520 Microcontroller User's Manual*, order #22004, for details about the interrupt controller.

Table 12-1 and Table 12-2 on page 12-2 list each type of PIC register in offset order, with the corresponding description's page number.

12.2 REGISTERS

Table 12-1 Programmable Interrupt Controller MMCR Registers

Register Name	Mnemonic	MMCR Offset	Page Number
Interrupt Control	PICICR	D00h	page 12-4
Master PIC Interrupt Mode	MPICMODE	D02h	page 12-6
Slave 1 PIC Interrupt Mode	SL1PICMODE	D03h	page 12-8
Slave 2 PIC Interrupt Mode	SL2PICMODE	D04h	page 12-9
Software Interrupt 16–1 Control	SWINT16_1	D08h	page 12-10
Software Interrupt 22–17/NMI Control	SWINT22_17	D0Ah	page 12-13
Interrupt Pin Polarity	INTPINPOL	D10h	page 12-15
PCI Host Bridge Interrupt Mapping	PCIHOSTMAP	D14h	page 12-17
ECC Interrupt Mapping	ECCMAP	D18h	page 12-19
GP Timer 0 Interrupt Mapping	GPTMR0MAP	D1Ah	page 12-21
GP Timer 1 Interrupt Mapping	GPTMR1MAP	D1Bh	page 12-21
GP Timer 2 Interrupt Mapping	GPTMR2MAP	D1Ch	page 12-21
PIT 0 Interrupt Mapping	PIT0MAP	D20h	page 12-21
PIT 1 Interrupt Mapping	PIT1MAP	D21h	page 12-21
PIT 2 Interrupt Mapping	PIT2MAP	D22h	page 12-21
UART 1 Interrupt Mapping	UART1MAP	D28h	page 12-21
UART 2 Interrupt Mapping	UART2MAP	D29h	page 12-21
PCI Interrupt A Mapping	PCIINTAMAP	D30h	page 12-21

Table 12-1 Programmable Interrupt Controller MMCR Registers (Continued)

Register Name	Mnemonic	MMCR Offset	Page Number
PCI Interrupt B Mapping	PCIINTBMAP	D31h	page 12-21
PCI Interrupt C Mapping	PCIINTCMAP	D32h	page 12-21
PCI Interrupt D Mapping	PCIINTDMAP	D33h	page 12-21
DMA Buffer Chaining Interrupt Mapping	DMABCINTMAP	D40h	page 12-21
SSI Interrupt Mapping	SSIMAP	D41h	page 12-21
Watchdog Timer Interrupt Mapping	WDTMAP	D42h	page 12-21
RTC Interrupt Mapping	RTCMAP	D43h	page 12-21
Write-Protect Violation Interrupt Mapping	WPVMAP	D44h	page 12-21
AMDebug™ Technology RX/TX Interrupt Mapping	ICEMAP	D45h	page 12-21
Floating Point Error Interrupt Mapping	FERRMAP	D46h	page 12-21
GPIRQ0 Interrupt Mapping	GP0IMAP	D50h	page 12-21
GPIRQ1 Interrupt Mapping	GP1IMAP	D51h	page 12-21
GPIRQ2 Interrupt Mapping	GP2IMAP	D52h	page 12-21
GPIRQ3 Interrupt Mapping	GP3IMAP	D53h	page 12-21
GPIRQ4 Interrupt Mapping	GP4IMAP	D54h	page 12-21
GPIRQ5 Interrupt Mapping	GP5IMAP	D55h	page 12-21
GPIRQ6 Interrupt Mapping	GP6IMAP	D56h	page 12-21
GPIRQ7 Interrupt Mapping	GP7IMAP	D57h	page 12-21
GPIRQ8 Interrupt Mapping	GP8IMAP	D58h	page 12-21
GPIRQ9 Interrupt Mapping	GP9IMAP	D59h	page 12-21
GPIRQ10 Interrupt Mapping	GP10IMAP	D5Ah	page 12-21

Table 12-2 Programmable Interrupt Controller Direct-Mapped Registers

Register Name	Mnemonic	I/O Address	Page Number
Master PIC Interrupt Request	MPICIR	0020h	page 12-24
Master PIC In-Service	MPICISR	0020h	page 12-25
Master PIC Initialization Control Word 1	MPICICW1	0020h	page 12-26
Master PIC Operation Control Word 2	MPICOCW2	0020h	page 12-28
Master PIC Operation Control Word 3	MPICOCW3	0020h	page 12-30
Master PIC Initialization Control Word 2	MPICICW2	0021h	page 12-32
Master PIC Initialization Control Word 3	MPICICW3	0021h	page 12-33
Master PIC Initialization Control Word 4	MPICICW4	0021h	page 12-35
Master PIC Interrupt Mask	MPICINTMSK	0021h	page 12-36
Slave 2 PIC Interrupt Request	S2PICIR	0024h	page 12-37
Slave 2 PIC In-Service	S2PICISR	0024h	page 12-38
Slave 2 PIC Initialization Control Word 1	S2PICICW1	0024h	page 12-39
Slave 2 PIC Operation Control Word 2	S2PICOCW2	0024h	page 12-41
Slave 2 PIC Operation Control Word 3	S2PICOCW3	0024h	page 12-43
Slave 2 PIC Initialization Control Word 2	S2PICICW2	0025h	page 12-45
Slave 2 PIC Initialization Control Word 3	S2PICICW3	0025h	page 12-46
Slave 2 PIC Initialization Control Word 4	S2PICICW4	0025h	page 12-47

Table 12-2 Programmable Interrupt Controller Direct-Mapped Registers (Continued)

Register Name	Mnemonic	I/O Address	Page Number
Slave 2 PIC Interrupt Mask	S2PICINTMSK	0025h	page 12-48
Slave 1 PIC Interrupt Request	S1PICIR	00A0h	page 12-49
Slave 1 PIC In-Service	S1PICISR	00A0h	page 12-50
Slave 1 PIC Initialization Control Word 1	S1PICICW1	00A0h	page 12-51
Slave 1 PIC Operation Control Word 2	S1PICOCW2	00A0h	page 12-53
Slave 1 PIC Operation Control Word 3	S1PICOCW3	00A0h	page 12-55
Slave 1 PIC Initialization Control Word 2	S1PICICW2	00A1h	page 12-57
Slave 1 PIC Initialization Control Word 3	S1PICICW3	00A1h	page 12-58
Slave 1 PIC Initialization Control Word 4	S1PICICW4	00A1h	page 12-59
Slave 1 PIC Interrupt Mask	S1PICINTMSK	00A1h	page 12-60
FPU Error Interrupt Clear	FPUERRCLR	00F0h	page 12-61

Interrupt Control (PICICR)

Memory-Mapped
MMCR Offset D00h

	7	6	5	4	3	2	1	0
Bit	NMI_DONE	NMI_ENB	Reserved			S2_GINT_ MODE	S1_GINT_ MODE	M_GINT_ MODE
Reset	0	0	0	0	0	1	1	1
R/W	R/W!	R/W	RSV			R/W	R/W	R/W

Register Description

This register controls the global interrupt mode for the Master, Slave 1 and Slave 2 programmable interrupt controllers, slave controller bypass, and the global nonmaskable interrupt (NMI) mask bit.

Bit Definitions

Bit	Name	Function
7	NMI_DONE	**NMI Routine Done** 0 = After this bit is set, it is cleared automatically by interrupt controller logic. 1 = Software sets this bit to indicate that the NMI routine is completed. Subsequent NMI events are blocked by the interrupt controller until the NMI_DONE bit is set by software, but if an NMI source is active at the time when the handler for another NMI sets this bit, then the interrupt controller generates an NMI for the second source shortly after returning from the first handler.
6	NMI_ENB	**Master NMI Enable** This bit is a read/write version of the NMI enable bit that typically resides at direct-mapped Port 0070h, bit 7, on a PC/AT-compatible system. It has been moved here to facilitate internal design integration with the ÉlanSC520 microcontroller. 0 = NMI is gated off from reaching the CPU core. 1 = NMI propagates to the CPU core. If any NMI interrupt sources are active when this bit is set, an NMI is generated immediately. The NMI_ENB bit is cleared by a CPU soft reset event. This allows software to initialize the stack pointer before setting the NMI_ENB bit again after a soft reset. See Table 3-3 on page 3-6 for a summary of ÉlanSC520 microcontroller reset sources.
5–3	Reserved	**Reserved** This bit field should be written to 0 for normal system operation.
2	S2_GINT_ MODE	**Slave 2 PIC Global Interrupt Mode Enable** This bit provides a global or individual channel interrupt mode for the Slave 2 PIC. 0 = Slave 2 PIC global interrupt mode disabled. 1 = Slave 2 PIC global interrupt mode enabled. If the S2_GINT_MODE bit is set, bit LTIM of the S2PICICW1 register (see page 12-39) determines the interrupt mode for the Slave 2 PIC channels. If the S2_GINT_MODE bit and the LTIM bit are set, all the Slave 2 PIC interrupt channels recognize level-sensitive interrupt requests. If the S2_GINT_MODE bit is set and the LTIM bit is cleared, all the Slave 2 PIC interrupt channels recognize edge-sensitive interrupt requests. If the S2_GINT_MODE bit is cleared, the Slave 2 LTIM bit has no meaning, and the Slave 2 PIC channels can be programmed individually via the SL2PICMODE register (see page 12-9) to select either edge- or level-sensitive interrupt recognition.

Bit	Name	Function
1	S1_GINT_MODE	**Slave 1 PIC Global Interrupt Mode Enable** This bit provides a global or individual channel interrupt mode for the Slave 1 PIC. 0 = Slave 1 PIC global interrupt mode disabled. 1 = Slave 1 PIC global interrupt mode enabled. If the S1_GINT_MODE bit is set, bit LTIM of the S1PICICW1 register (see page 12-51) determines the interrupt mode for the Slave 1 PIC channels. If the S1_GINT_MODE bit and the LTIM bit are set, all the Slave 1 PIC interrupt channels recognize level-sensitive interrupt requests. If the S1_GINT_MODE bit is set and the LTIM bit is cleared, all the Slave 1 PIC interrupt channels recognize edge-sensitive interrupt requests. If the S1_GINT_MODE bit is cleared, the Slave 1 LTIM bit has no meaning, and the Slave 1 PIC channels can be programmed individually via the SL1PICMODE register (see page 12-8) to select either edge- or level-sensitive interrupt recognition.
0	M_GINT_MODE	**Master PIC Global Interrupt Mode Enable** This bit provides a global or individual channel interrupt mode for the Master PIC. 0 = Master PIC global interrupt mode disabled. 1 = Master PIC global interrupt mode enabled. If the M_GINT_MODE bit is set, bit LTIM of the MPICICW1 register (see page 12-26) determines the interrupt mode for the Master PIC channels. If the M_GINT_MODE bit and the LTIM bit are set, all the Master PIC interrupt channels recognize level-sensitive interrupt requests. If the M_GINT_MODE bit is set and the LTIM bit is cleared, all the Master PIC interrupt channels recognize edge-sensitive interrupt requests. If the M_GINT_MODE bit is cleared, the Master LTIM bit has no meaning, and the Master PIC channels can be programmed individually via the MPICMODE register (see page 12-6) to select either edge- or level-sensitive interrupt recognition.

Programming Notes

For PC/AT compatibility, bits M_GINT_MODE and S1_GINT_MODE in this register (PICICR) should be set, and bit S2 should be set and bit S5 cleared in the MPICICW3 register (see page 12-34). In this configuration, the Slave 2 controller is bypassed and any interrupt sources mapped to the Slave 2 controller have no effect; also, interrupt sources can be mapped directly to Master PIC interrupt channel 5 by mapping them to priority P13.

Master PIC Interrupt Mode (MPICMODE)

Memory-Mapped
MMCR Offset D02h

	7	6	5	4	3	2	1	0
Bit	CH7_INT_ MODE	CH6_INT_ MODE	CH5_INT_ MODE	CH4_INT_ MODE	CH3_INT_ MODE	CH2_INT_ MODE	CH1_INT_ MODE	CH0_INT_ MODE
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Register Description

This register controls the individual Master PIC channel interrupt mode.

Bit Definitions

Bit	Name	Function
7	CH7_INT_ MODE	**Master PIC Channel 7 Interrupt Mode** 0 = Edge-sensitive interrupt request detection 1 = Level-sensitive interrupt request detection
6	CH6_INT_ MODE	**Master PIC Channel 6 Interrupt Mode** 0 = Edge-sensitive interrupt request detection 1 = Level-sensitive interrupt request detection
5	CH5_INT_ MODE	**Master PIC Channel 5 Interrupt Mode** 0 = Edge-sensitive interrupt request detection 1 = Level-sensitive interrupt request detection If the SNGL bit in the MPICICW1 register is set (see page 12-26), and the M_GINT_MODE bit of the PICICR register is cleared (see page 12-5), setting the CH5_INT_MODE bit causes interrupts to be recognized as level-sensitive on channel 5. Clearing the CH5_INT_MODE bit under the same condition causes channel 5 interrupts to be recognized as edge-sensitive. However, if both the SNGL bit and the M_GINT_MODE bit are cleared, the CH5_INT_MODE bit value has no meaning because the channel is used for cascading with the Slave 2 controller.
4	CH4_INT_ MODE	**Master PIC Channel 4 Interrupt Mode** 0 = Edge-sensitive interrupt request detection 1 = Level-sensitive interrupt request detection
3	CH3_INT_ MODE	**Master PIC Channel 3 Interrupt Mode** 0 = Edge-sensitive interrupt request detection 1 = Level-sensitive interrupt request detection
2	CH2_INT_ MODE	**Master PIC Channel 2 Interrupt Mode** 0 = Edge-sensitive interrupt request detection 1 = Level-sensitive interrupt request detection If the SNGL bit in the MPICICW1 register is set (see page 12-26), and the M_GINT_MODE bit of the PICICR register is cleared (see page 12-5), setting the CH2_INT_MODE bit causes interrupts to be recognized as level-sensitive on channel 2. Clearing the CH2_INT_MODE bit under the same condition causes channel 2 interrupts to be recognized as edge-sensitive. However, if both the SNGL bit and the M_GINT_MODE bit are cleared, the CH2_INT_MODE bit value has no meaning because the channel is used for cascading with the Slave 1 controller.
1	CH1_INT_ MODE	**Master PIC Channel 1 Interrupt Mode** 0 = Edge-sensitive interrupt request detection 1 = Level-sensitive interrupt request detection

Bit	Name	Function
0	CH0_INT_MODE	**Master PIC Channel 0 Interrupt Mode** 0 = Edge-sensitive interrupt request detection 1 = Level-sensitive interrupt request detection

Programming Notes

There is a global overriding bit, M_GINT_MODE in the PICICR register (see page 12-5). When set, the M_GINT_MODE bit causes this register (MPICMODE) to have no effect on the interrupt mode programmed for each channel. If the M_GINT_MODE bit is set, the overriding global interrupt mode for the Master PIC channels is determined by the LTIM bit in the MPICICW1 register (see page 12-26). If the M_GINT_MODE bit is cleared, the LTIM bit has no meaning, and the MPICMODE register bits take effect for determining each Master PIC channel's interrupt mode.

Slave 1 PIC Interrupt Mode (SL1PICMODE)

Memory-Mapped
MMCR Offset D03h

	7	6	5	4	3	2	1	0
Bit	CH7_INT_ MODE	CH6_INT_ MODE	CH5_INT_ MODE	CH4_INT_ MODE	CH3_INT_ MODE	CH2_INT_ MODE	CH1_INT_ MODE	CH0_INT_ MODE
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Register Description

This register controls the individual Slave 1 PIC channel interrupt mode.

Bit Definitions

Bit	Name	Function
7	CH7_INT_ MODE	**Slave 1 PIC Channel 7 Interrupt Mode** 0 = Edge-sensitive interrupt request detection 1 = Level-sensitive interrupt request detection
6	CH6_INT_ MODE	**Slave 1 PIC Channel 6 Interrupt Mode** 0 = Edge-sensitive interrupt request detection 1 = Level-sensitive interrupt request detection
5	CH5_INT_ MODE	**Slave 1 PIC Channel 5 Interrupt Mode** 0 = Edge-sensitive interrupt request detection 1 = Level-sensitive interrupt request detection
4	CH4_INT_ MODE	**Slave 1 PIC Channel 4 Interrupt Mode** 0 = Edge-sensitive interrupt request detection 1 = Level-sensitive interrupt request detection
3	CH3_INT_ MODE	**Slave 1 PIC Channel 3 Interrupt Mode** 0 = Edge-sensitive interrupt request detection 1 = Level-sensitive interrupt request detection
2	CH2_INT_ MODE	**Slave 1 PIC Channel 2 Interrupt Mode** 0 = Edge-sensitive interrupt request detection 1 = Level-sensitive interrupt request detection
1	CH1_INT_ MODE	**Slave 1 PIC Channel 1 Interrupt Mode** 0 = Edge-sensitive interrupt request detection 1 = Level-sensitive interrupt request detection
0	CH0_INT_ MODE	**Slave 1 PIC Channel 0 Interrupt Mode** 0 = Edge-sensitive interrupt request detection 1 = Level-sensitive interrupt request detection

Programming Notes

There is a global overriding bit, S1_GINT_MODE in the PICICR register (see page 12-5). When set, the S1_GINT_MODE bit causes this register (SL1PICMODE) to have no effect on the interrupt mode programmed for each channel. If the S1_GINT_MODE bit is set, the overriding global interrupt mode for the Slave 1 PIC channels is determined by the LTIM bit in the S1PICICW1 register (see page 12-51). If the S1_GINT_MODE bit is cleared, the LTIM bit has no meaning, and the SL1PICMODE register bits take effect for determining each Slave 1 PIC channel's interrupt mode.

Slave 2 PIC Interrupt Mode (SL2PICMODE)

Memory-Mapped
MMCR Offset D04h

	7	6	5	4	3	2	1	0
Bit	CH7_INT_ MODE	CH6_INT_ MODE	CH5_INT_ MODE	CH4_INT_ MODE	CH3_INT_ MODE	CH2_INT_ MODE	CH1_INT_ MODE	CH0_INT_ MODE
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Register Description

This register controls the individual Slave 2 PIC channel interrupt mode.

Bit Definitions

Bit	Name	Function
7	CH7_INT_ MODE	**Slave 2 PIC Channel 7 Interrupt Mode** 0 = Edge-sensitive interrupt request detection 1 = Level-sensitive interrupt request detection
6	CH6_INT_ MODE	**Slave 2 PIC Channel 6 Interrupt Mode** 0 = Edge-sensitive interrupt request detection 1 = Level-sensitive interrupt request detection
5	CH5_INT_ MODE	**Slave 2 PIC Channel 5 Interrupt Mode** 0 = Edge-sensitive interrupt request detection 1 = Level-sensitive interrupt request detection
4	CH4_INT_ MODE	**Slave 2 PIC Channel 4 Interrupt Mode** 0 = Edge-sensitive interrupt request detection 1 = Level-sensitive interrupt request detection
3	CH3_INT_ MODE	**Slave 2 PIC Channel 3 Interrupt Mode** 0 = Edge-sensitive interrupt request detection 1 = Level-sensitive interrupt request detection
2	CH2_INT_ MODE	**Slave 2 PIC Channel 2 Interrupt Mode** 0 = Edge-sensitive interrupt request detection 1 = Level-sensitive interrupt request detection
1	CH1_INT_ MODE	**Slave 2 PIC Channel 1 Interrupt Mode** 0 = Edge-sensitive interrupt request detection 1 = Level-sensitive interrupt request detection
0	CH0_INT_ MODE	**Slave 2 PIC Channel 0 Interrupt Mode** 0 = Edge-sensitive interrupt request detection 1 = Level-sensitive interrupt request detection

Programming Notes

There is a global overriding bit, S2_GINT_MODE in the PICICR register (see page 12-5). When set, the S2_GINT_MODE bit causes this register (SL2PICMODE) to have no effect on the interrupt mode programmed for each channel. If the S2_GINT_MODE bit is set, the overriding global interrupt mode for the Slave 2 PIC channels is determined by the LTIM bit in the S2PICICW1 register (see page 12-51). If the S2_GINT_MODE bit is cleared, the LTIM bit has no meaning, and the SL2PICMODE register bits take effect for determining each Slave 2 PIC channel's interrupt mode.

Software Interrupt 16–1 Control (SWINT16_1)

Memory-Mapped
MMCR Offset D08h

	15	14	13	12	11	10	9	8
Bit	SW_P16_ TRIG	SW_P15_ TRIG	SW_P14_ TRIG	SW_P13_ TRIG	SW_P12_ TRIG	SW_P11_ TRIG	SW_P10_ TRIG	SW_P9_ TRIG
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

	7	6	5	4	3	2	1	0
Bit	SW_P8_ TRIG	SW_P7_ TRIG	SW_P6_ TRIG	SW_P5_ TRIG	SW_P4_ TRIG	SW_P3_ TRIG	SW_P2_ TRIG	SW_P1_ TRIG
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Register Description

This register allows software to generate interrupt priority levels P1–P16 to the CPU.

Bit Definitions

Bit	Name	Function
15	SW_P16_TRIG	**Directly Trigger Priority Level P16** Setting this bit directly asserts a maskable interrupt of priority level P16. Clearing this bit removes this direct interrupt assertion. 0 = Do not assert the interrupt. 1 = Assert the interrupt.
14	SW_P15_TRIG	**Directly Trigger Priority Level P15** Setting this bit directly asserts a maskable interrupt of priority level P15. Clearing this bit removes this direct interrupt assertion. 0 = Do not assert the interrupt. 1 = Assert the interrupt.
13	SW_P14_TRIG	**Directly Trigger Priority Level P14** Setting this bit directly asserts a maskable interrupt of priority level P14. Clearing this bit removes this direct interrupt assertion. 0 = Do not assert the interrupt. 1 = Assert the interrupt.
12	SW_P13_TRIG	**Directly Trigger Priority Level P13** Setting this bit directly asserts a maskable interrupt of priority level P13. Clearing this bit removes this direct interrupt assertion. 0 = Do not assert the interrupt. 1 = Assert the interrupt.
11	SW_P12_TRIG	**Directly Trigger Priority Level P12** Setting this bit directly asserts a maskable interrupt of priority level P12. Clearing this bit removes this direct interrupt assertion. 0 = Do not assert the interrupt. 1 = Assert the interrupt.

Bit	Name	Function
10	SW_P11_TRIG	**Directly Trigger Priority Level P11** Setting this bit directly asserts a maskable interrupt of priority level P11. Clearing this bit removes this direct interrupt assertion. 0 = Do not assert the interrupt. 1 = Assert the interrupt.
9	SW_P10_TRIG	**Directly Trigger Priority Level P10** Setting this bit directly asserts a maskable interrupt of priority level P10. Clearing this bit removes this direct interrupt assertion. 0 = Do not assert the interrupt. 1 = Assert the interrupt.
8	SW_P9_TRIG	**Directly Trigger Priority Level P9** Setting this bit directly asserts a maskable interrupt of priority level P9. Clearing this bit removes this direct interrupt assertion. 0 = Do not assert the interrupt. 1 = Assert the interrupt.
7	SW_P8_TRIG	**Directly Trigger Priority Level P8** Setting this bit directly asserts a maskable interrupt of priority level P8. Clearing this bit removes this direct interrupt assertion. 0 = Do not assert the interrupt. 1 = Assert the interrupt.
6	SW_P7_TRIG	**Directly Trigger Priority Level P7** Setting this bit directly asserts a maskable interrupt of priority level P7. Clearing this bit removes this direct interrupt assertion. 0 = Do not assert the interrupt. 1 = Assert the interrupt.
5	SW_P6_TRIG	**Directly Trigger Priority Level P6** Setting this bit directly asserts a maskable interrupt of priority level P6. Clearing this bit removes this direct interrupt assertion. 0 = Do not assert the interrupt. 1 = Assert the interrupt.
4	SW_P5_TRIG	**Directly Trigger Priority Level P5** Setting this bit directly asserts a maskable interrupt of priority level P5. Clearing this bit removes this direct interrupt assertion. 0 = Do not assert the interrupt. 1 = Assert the interrupt.
3	SW_P4_TRIG	**Directly Trigger Priority Level P4** Setting this bit directly asserts a maskable interrupt of priority level P4. Clearing this bit removes this direct interrupt assertion. 0 = Do not assert the interrupt. 1 = Assert the interrupt.
2	SW_P3_TRIG	**Directly Trigger Priority Level P3** Setting this bit directly asserts a maskable interrupt of priority level P3. Clearing this bit removes this direct interrupt assertion. 0 = Do not assert the interrupt. 1 = Assert the interrupt.
1	SW_P2_TRIG	**Directly Trigger Priority Level P2** Setting this bit directly asserts a maskable interrupt of priority level P2. Clearing this bit removes this direct interrupt assertion. 0 = Do not assert the interrupt. 1 = Assert the interrupt.

Bit	Name	Function
0	SW_P1_TRIG	**Directly Trigger Priority Level P1** Setting this bit directly asserts a maskable interrupt of priority level P1. Clearing this bit removes this direct interrupt assertion. This is the highest interrupt priority level. 0 = Do not assert the interrupt. 1 = Assert the interrupt.

Programming Notes

This register (SWINT16–1) and register SWINT22–17 provide access to all 22 maskable interrupt priority levels. Priority level P1 is the highest priority, and priority level P22 is the lowest.

Setting an interrupt trigger bit causes an interrupt to be asserted on the corresponding PIC's interrupt channel for as long as the bit is set. To use these bits effectively, the interrupt service routines should clear the interrupt trigger bit early in the routine.

Note that for the internal PC/AT-compatible peripherals, and for many PCI peripherals, existing drivers in off-the-shelf operating systems are not aware of these interrupt trigger bits. For peripherals with these kinds of interrupt service routines, care must be taken *not* to set the interrupt trigger bits. If this occurs, the interrupt gets stuck in the asserted state and the system loops in the interrupt service routine because the routine does not clear the trigger bit.

Software Interrupt 22–17/NMI Control (SWINT22_17)

Memory-Mapped
MMCR Offset D0Ah

	7	6	5	4	3	2	1	0
Bit	Reserved	NMI_TRIG	SW_P22_TRIG	SW_P21_TRIG	SW_P20_TRIG	SW_P19_TRIG	SW_P18_TRIG	SW_P17_TRIG
Reset	0	0	0	0	0	0	0	0
R/W	RSV	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Register Description

This register allows software to generate interrupt priority levels P17–P22 or the nonmaskable interrupt (NMI) to the CPU.

Bit Definitions

Bit	Name	Function
7	Reserved	**Reserved** This bit field should be written to 0 for normal system operation.
6	NMI_TRIG	**Software NMI Source** Setting this bit directly asserts an NMI to the CPU. This bit should be cleared by software prior to setting the NMI_DONE bit of the PICICR register (see page 12-4). 0 = NMI not directly asserted (other sources can still assert an NMI). 1 = Directly assert the NMI to the CPU.
5	SW_P22_TRIG	**Directly Trigger Priority Level P22** Setting this bit directly asserts a maskable interrupt of priority level P22. Clearing this bit removes this direct interrupt assertion. This is the lowest interrupt priority level. 0 = Do not assert the interrupt. 1 = Assert the interrupt.
4	SW_P21_TRIG	**Directly Trigger Priority Level P21** Setting this bit directly asserts a maskable interrupt of priority level P21. Clearing this bit removes this direct interrupt assertion. 0 = Do not assert the interrupt. 1 = Assert the interrupt.
3	SW_P20_TRIG	**Directly Trigger Priority Level P20** Setting this bit directly asserts a maskable interrupt of priority level P20. Clearing this bit removes this direct interrupt assertion. 0 = Do not assert the interrupt. 1 = Assert the interrupt.
2	SW_P19_TRIG	**Directly Trigger Priority Level P19** Setting this bit directly asserts a maskable interrupt of priority level P19. Clearing this bit removes this direct interrupt assertion. 0 = Do not assert the interrupt. 1 = Assert the interrupt.
1	SW_P18_TRIG	**Directly Trigger Priority Level P18** Setting this bit directly asserts a maskable interrupt of priority level P18. Clearing this bit removes this direct interrupt assertion. 0 = Do not assert the interrupt. 1 = Assert the interrupt.

Bit	Name	Function
0	SW_P17_TRIG	**Directly Trigger Priority Level P17** Setting this bit directly asserts a maskable interrupt of priority level P17. Clearing this bit removes this direct interrupt assertion. 0 = Do not assert the interrupt. 1 = Assert the interrupt.

Programming Notes

This register (SWINT22-17) and register SWINT16–1 provide access to all 22 maskable interrupt priority levels. Priority level P1 is the highest priority, and priority level P22 is the lowest.

Setting an interrupt trigger bit causes an interrupt to be asserted on the corresponding PIC's interrupt channel for as long as the bit is set. To use these bits effectively, the interrupt service routines should clear the interrupt trigger bit early in the routine.

Note that for the internal PC/AT-compatible peripherals, and for many PCI peripherals, existing drivers in off-the-shelf operating systems are not aware of these interrupt trigger bits. For peripherals with these kinds of interrupt service routines, care must be taken *not* to set the interrupt trigger bits. If this occurs, the interrupt gets stuck in the asserted state and the system loops in the interrupt service routine because the routine does not clear the trigger bit.

Interrupt Pin Polarity (INTPINPOL)

Memory-Mapped
MMCR Offset D10h

	15	14	13	12	11	10	9	8
Bit	INTD_POL	INTC_POL	INTB_POL	INTA_POL	Reserved	GPINT10 _POL	GPINT9_ POL	GPINT8_ POL
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	RSV	R/W	R/W	R/W

	7	6	5	4	3	2	1	0
Bit	GPINT7_ POL	GPINT6_ POL	GPINT5_ POL	GPINT4_ POL	GPINT3_ POL	GPINT2_ POL	GPINT1_ POL	GPINT0_ POL
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Register Description

This register determines the polarity to be used for each external interrupt source.

Bit Definitions

Bit	Name	Function
15	INTD_POL	**PCI Interrupt Request $\overline{\text{INTD}}$ Polarity** 0 = High-to-Low transition or Low-level-sensitive interrupt 1 = Low-to-High transition or High-level-sensitive interrupt
14	INTC_POL	**PCI Interrupt Request $\overline{\text{INTC}}$ Polarity** 0 = High-to-Low transition or Low-level-sensitive interrupt 1 = Low-to-High transition or High-level-sensitive interrupt
13	INTB_POL	**PCI Interrupt Request $\overline{\text{INTB}}$ Polarity** 0 = High-to-Low transition or Low-level-sensitive interrupt 1 = Low-to-High transition or High-level-sensitive interrupt
12	INTA_POL	**PCI Interrupt Request $\overline{\text{INTA}}$ Polarity** 0 = High-to-Low transition or Low-level-sensitive interrupt 1 = Low-to-High transition or High-level-sensitive interrupt
11	Reserved	**Reserved** This bit field should be written to 0 for normal system operation.
10	GPINT10_POL	**General-Purpose Interrupt Request GPIRQ10 Polarity** 0 = Low-to-High transition or High-level-sensitive interrupt 1 = High-to-Low transition or Low-level-sensitive interrupt
9	GPINT9_POL	**General-Purpose Interrupt Request GPIRQ9 Polarity** 0 = Low-to-High transition or High-level-sensitive interrupt 1 = High-to-Low transition or Low-level-sensitive interrupt
8	GPINT8_POL	**General-Purpose Interrupt Request GPIRQ8 Polarity** 0 = Low-to-High transition or High-level-sensitive interrupt 1 = High-to-Low transition or Low-level-sensitive interrupt
7	GPINT7_POL	**General-Purpose Interrupt Request GPIRQ7 Polarity** 0 = Low-to-High transition or High-level-sensitive interrupt 1 = High-to-Low transition or Low-level-sensitive interrupt

Bit	Name	Function
6	GPINT6_POL	**General-Purpose Interrupt Request GPIRQ6 Polarity** 0 = Low-to-High transition or High-level-sensitive interrupt 1 = High-to-Low transition or Low-level-sensitive interrupt
5	GPINT5_POL	**General-Purpose Interrupt Request GPIRQ5 Polarity** 0 = Low-to-High transition or High-level-sensitive interrupt 1 = High-to-Low transition or Low-level-sensitive interrupt
4	GPINT4_POL	**General-Purpose Interrupt Request GPIRQ4 Polarity** 0 = Low-to-High transition or High-level-sensitive interrupt 1 = High-to-Low transition or Low-level-sensitive interrupt
3	GPINT3_POL	**General-Purpose Interrupt Request GPIRQ3 Polarity** 0 = Low-to-High transition or High-level-sensitive interrupt 1 = High-to-Low transition or Low-level-sensitive interrupt
2	GPINT2_POL	**General-Purpose Interrupt Request GPIRQ2 Polarity** 0 = Low-to-High transition or High-level-sensitive interrupt 1 = High-to-Low transition or Low-level-sensitive interrupt
1	GPINT1_POL	**General-Purpose Interrupt Request GPIRQ1 Polarity** 0 = Low-to-High transition or High-level-sensitive interrupt 1 = High-to-Low transition or Low-level-sensitive interrupt
0	GPINT0_POL	**General-Purpose Interrupt Request GPIRQ0 Polarity** 0 = Low-to-High transition or High-level-sensitive interrupt 1 = High-to-Low transition or Low-level-sensitive interrupt

Programming Notes

This register should be programmed only when the corresponding interrupt channel mask bits are enabled.

At system reset, the INTx_POL bits (bits 15–12 of this register, INTPINPOL) are cleared to enable the active Low PCI interrupt to be used as a default so that no inversion is necessary for the PCI interrupt to be recognized correctly. If the $\overline{\text{INTx}}$ pins are to be configured for use as general-purpose interrupt sources (as opposed to PCI interrupt sources), then the correct polarity can be programmed as necessary using this register.

PCI Host Bridge Interrupt Mapping (PCIHOSTMAP)

Memory-Mapped
MMCR Offset D14h

	15	14	13	12	11	10	9	8
Bit	Reserved							PCI_NMI_ENB
Reset	0	0	0	0	0	0	0	0
R/W	RSV							R/W

	7	6	5	4	3	2	1	0
Bit	Reserved			PCI_IRQ_MAP[4–0]				
Reset	0	0	0	0	0	0	0	0
R/W	RSV			R/W				

Register Description

This register maps the internally-generated PCI host bridge interrupt to any of the desired interrupt channels or as an NMI. It also enables the PCI Host Bridge NMI request as an NMI source.

Bit Definitions

Bit	Name	Function
15–9	Reserved	**Reserved** This bit field should be written to 0 for normal system operation.
8	PCI_NMI_ENB	**PCI Host Bridge NMI Enable** This bit enables the PCI Host Bridge NMI request as an NMI source. 0 = PCI Host Bridge NMI request disabled as an NMI source 1 = PCI Host Bridge NMI request enabled as an NMI source This bit (PCI_NMI_ENB) has no effect on the PCI bus Host Bridge interrupt request, which can configured to generate an NMI by setting the PCI_IRQ_MAP bit field to 11111b.
7–5	Reserved	**Reserved** This bit field should be written to 0 for normal system operation.

Bit	Name	Function
4–0	PCI_IRQ_MAP [4–0]	**PCI Host Bridge Interrupt Mapping** The value in this 5-bit field maps the internally-generated PCI Host Bridge interrupt to one of the following interrupt priority channels on the microcontroller or as an NMI. 00000 = Disables the internally-generated PCI interrupt from reaching the PIC 00001 = Priority P1 (Master PIC IR0) (Highest priority) 00010 = Priority P2 (Master PIC IR1) 00011 = Priority P3 (Slave 1 PIC IR0/Master PIC IR2) 00100 = Priority P4 (Slave 1 PIC IR1) 00101 = Priority P5 (Slave 1 PIC IR2) 00110 = Priority P6 (Slave 1 PIC IR3) 00111 = Priority P7 (Slave 1 PIC IR4) 01000 = Priority P8 (Slave 1 PIC IR5) 01001 = Priority P9 (Slave 1 PIC IR6) 01010 = Priority P10 (Slave 1 PIC IR7) 01011 = Priority P11 (Master PIC IR3) 01100 = Priority P12 (Master PIC IR4) 01101 = Priority P13 (Slave 2 PIC IR0/Master PIC IR5) 01110 = Priority P14 (Slave 2 PIC IR1) 01111 = Priority P15 (Slave 2 PIC IR2) 10000 = Priority P16 (Slave 2 PIC IR3) 10001 = Priority P17 (Slave 2 PIC IR4) 10010 = Priority P18 (Slave 2 PIC IR5) 10011 = Priority P19 (Slave 2 PIC IR6) 10100 = Priority P20 (Slave 2 PIC IR7) 10101 = Priority P21 (Master PIC IR6) 10110 = Priority P22 (Master PIC IR7) (Lowest priority) 10111–11110 = Disables the internally-generated PCI interrupt from reaching the PIC 11111 = NMI source For example, if PCI_IRQ_MAP = 01101b, the PCI interrupt request is mapped to interrupt priority P13 in the microcontroller. If PCI_IRQ_MAP = 00000b or any binary value from 10111–11110b, the PCI interrupt request is disabled from reaching the microcontroller's PIC. If this field is set to 11111b, then the PCI interrupt is routed and enabled as an NMI source. If bit S2 in the MPICICW3 register is cleared (see page 12-34), the Slave 1 PIC is bypassed, so programming the PCI_IRQ_MAP bit field to a value in the range 00100–01010b does not pass the interrupt request to the CPU. However, if this bit field is programmed to 00011b with the S2 bit cleared, the PCI interrupt request is routed to the Master PIC IR2 input. If bit S5 in the MPICICW3 register is cleared (see page 12-33), the Slave 2 PIC is bypassed, so programming the PCI_IRQ_MAP bit field to a value in the range 01110–10100b does not pass the interrupt request to the CPU. However, if this field is programmed to 01101b with the S5 bit cleared, the PCI interrupt request is routed to Master PIC IR5 input.

Programming Notes

This register should be programmed only when the corresponding interrupt channel mask bits are set in the PIC.

For NMIs, this register should be programmed only when bit NMI_ENB is cleared in the PICICR register (see page 12-4). NMI_ENB can be set immediately after programming this register (PCIHOSTMAP) to allow NMIs to be passed to the CPU.

Programming more than one interrupt source to an interrupt channel results in interrupt sharing on that channel. Programming more than one interrupt source as an NMI source results in NMI sharing on the CPU's NMI input.

All interrupt and NMI sources can be found in the following register descriptions:

- Software Interrupt 16–1 Control, page 12-10
- Software Interrupt 22–17/NMI Control, page 12-13
- PCI Host Bridge Interrupt Mapping, page 12-17
- ECC Interrupt Mapping, page 12-19
- Other interrupt mapping registers (30), page 12-21

ECC Interrupt Mapping (ECCMAP)

Memory-Mapped
MMCR Offset D18h

	15	14	13	12	11	10	9	8
Bit	Reserved							ECC_NMI_ENB
Reset	0	0	0	0	0	0	0	0
R/W	RSV							R/W

	7	6	5	4	3	2	1	0
Bit	Reserved			ECC_IRQ_MAP[4–0]				
Reset	0	0	0	0	0	0	0	0
R/W	RSV			R/W				

Register Description

This register maps the internally-generated ECC single-bit interrupt to any of the desired interrupt channels. It also enables the ECC multi-bit error as an NMI source.

Bit Definitions

Bit	Name	Function
15–9	Reserved	**Reserved** This bit field should be written to 0 for normal system operation.
8	ECC_NMI_ENB	**ECC NMI Enable** This bit enables the ECC multiple-bit error as an NMI source. 0 = ECC multiple-bit error disabled as an NMI source. 1 = ECC multiple-bit error enabled as an NMI source.
7–5	Reserved	**Reserved** This bit field should be written to 0 for normal system operation.

Bit	Name	Function
4–0	ECC_IRQ_MAP [4–0]	**SDRAM ECC Interrupt Mapping** The value in this 5-bit field maps the internally-generated SDRAM ECC single-bit interrupt to one of the following interrupt priority channels on the microcontroller. 00000 = Disables the internally-generated ECC interrupt from reaching the PIC 00001 = Priority P1 (Master PIC IR0) (Highest priority) 00010 = Priority P2 (Master PIC IR1) 00011 = Priority P3 (Slave 1 PIC IR0/Master PIC IR2) 00100 = Priority P4 (Slave 1 PIC IR1) 00101 = Priority P5 (Slave 1 PIC IR2) 00110 = Priority P6 (Slave 1 PIC IR3) 00111 = Priority P7 (Slave 1 PIC IR4) 01000 = Priority P8 (Slave 1 PIC IR5) 01001 = Priority P9 (Slave 1 PIC IR6) 01010 = Priority P10 (Slave 1 PIC IR7) 01011 = Priority P11 (Master PIC IR3) 01100 = Priority P12 (Master PIC IR4) 01101 = Priority P13 (Slave 2 PIC IR0/Master PIC IR5) 01110 = Priority P14 (Slave 2 PIC IR1) 01111 = Priority P15 (Slave 2 PIC IR2) 10000 = Priority P16 (Slave 2 PIC IR3) 10001 = Priority P17 (Slave 2 PIC IR4) 10010 = Priority P18 (Slave 2 PIC IR5) 10011 = Priority P19 (Slave 2 PIC IR6) 10100 = Priority P20 (Slave 2 PIC IR7) 10101 = Priority P21 (Master PIC IR6) 10110 = Priority P22 (Master PIC IR7) (Lowest priority) 10111–11111 = Disables the internally-generated ECC interrupt from reaching the PIC For example, if ECC_IRQ_MAP = 01101b, the ECC interrupt request is mapped to interrupt priority P13 in the microcontroller. If ECC_IRQ_MAP = 00000b or any binary value from 10111–11111b, the ECC interrupt request is disabled from reaching the microcontroller's PIC. The single-bit ECC interrupt request cannot be routed as a source for generating an NMI. If bit S2 in the MPICICW3 register is cleared (see page 12-34), the Slave 1 PIC is bypassed, so programming the ECC_IRQ_MAP bit field to a value in the range 00100–01010b does not pass the interrupt request to the CPU. However, if this bit field is programmed to 00011b with the S2 bit cleared, the ECC interrupt request is routed to the Master PIC IR2 input. If bit S5 in the MPICICW3 register is cleared (see page 12-33), the Slave 2 PIC is bypassed, so programming the ECC_IRQ_MAP bit field to a value in the range 01110–10100b does not pass the interrupt request to the CPU. However, if this field is programmed to 01101b with the S5 bit cleared, the ECC interrupt request is routed to Master PIC IR5 input.

Programming Notes

This register should be programmed only when the corresponding interrupt channel mask bits are set in the PIC.

For NMIs, this register should be programmed only when bit NMI_ENB is cleared in the PICICR register (see page 12-4). NMI_ENB can be set immediately after programming this register (ECCMAP) to allow NMIs to be passed to the CPU.

Programming more than one interrupt source to an interrupt channel results in interrupt sharing on that channel. Programming more than one interrupt source as an NMI source results in NMI sharing on the CPU's NMI input.

All interrupt and NMI sources can be found in the following register descriptions:

- Software Interrupt 16–1 Control, page 12-10
- Software Interrupt 22–17/NMI Control, page 12-13
- PCI Host Bridge Interrupt Mapping, page 12-17
- ECC Interrupt Mapping, page 12-19
- Other interrupt mapping registers (30), page 12-21

	Memory-Mapped
GP Timer 0 Interrupt Mapping (GPTMR0MAP)	MMCR Offset D1Ah
GP Timer 1 Interrupt Mapping (GPTMR1MAP)	MMCR Offset D1Bh
GP Timer 2 Interrupt Mapping (GPTMR2MAP)	MMCR Offset D1Ch
PIT 0 Interrupt Mapping (PIT0MAP)	MMCR Offset D20h
PIT 1 Interrupt Mapping (PIT1MAP)	MMCR Offset D21h
PIT 2 Interrupt Mapping (PIT2MAP)	MMCR Offset D22h
UART 1 Interrupt Mapping (UART1MAP)	MMCR Offset D28h
UART 2 Interrupt Mapping (UART2MAP)	MMCR Offset D29h
PCI Interrupt A Mapping (PCIINTAMAP)	MMCR Offset D30h
PCI Interrupt B Mapping (PCIINTBMAP)	MMCR Offset D31h
PCI Interrupt C Mapping (PCIINTCMAP)	MMCR Offset D32h
PCI Interrupt D Mapping (PCIINTDMAP)	MMCR Offset D33h
DMA Buffer Chaining Interrupt Mapping (DMABCINTMAP)	MMCR Offset D40h
SSI Interrupt Mapping (SSIMAP)	MMCR Offset D41h
Watchdog Timer Interrupt Mapping (WDTMAP)	MMCR Offset D42h
RTC Interrupt Mapping (RTCMAP)	MMCR Offset D43h
Write-Protect Violation Interrupt Mapping (WPVMAP)	MMCR Offset D44h
AMDebug™ Technology RX/TX Interrupt Mapping (ICEMAP)	MMCR Offset D45h
Floating Point Error Interrupt Mapping (FERRMAP)	MMCR Offset D46h
GPIRQ0 Interrupt Mapping (GP0IMAP)	MMCR Offset D50h
GPIRQ1 Interrupt Mapping (GP1IMAP)	MMCR Offset D51h
GPIRQ2 Interrupt Mapping (GP2IMAP)	MMCR Offset D52h
GPIRQ3 Interrupt Mapping (GP3IMAP)	MMCR Offset D53h
GPIRQ4 Interrupt Mapping (GP4IMAP)	MMCR Offset D54h
GPIRQ5 Interrupt Mapping (GP5IMAP)	MMCR Offset D55h
GPIRQ6 Interrupt Mapping (GP6IMAP)	MMCR Offset D56h
GPIRQ7 Interrupt Mapping (GP7IMAP)	MMCR Offset D57h
GPIRQ8 Interrupt Mapping (GP8IMAP)	MMCR Offset D58h
GPIRQ9 Interrupt Mapping (GP9IMAP)	MMCR Offset D59h
GPIRQ10 Interrupt Mapping (GP10IMAP)	MMCR Offset D5Ah

	7	6	5	4	3	2	1	0
Bit	Reserved			INT_MAP[4–0]				
Reset	0	0	0	0	0	0	0	0
R/W	RSV			R/W				

Register Description

These registers map each of the interrupt sources (except for ECC and PCI Host Bridge interrupts, see page 12-19 and page 12-17, respectively) to their desired interrupt channel or NMI.

Bit Definitions

Bit	Name	Function
7–5	Reserved	**Reserved** This bit field should be written to 0 for normal system operation.
4–0	INT_MAP[4–0]	**Interrupt Mapping** The value in this 5-bit field maps the interrupt source for the register to one of the following interrupt priority channels on the microcontroller or as an NMI. 00000 = Disables the interrupt source as an input 00001 = Priority P1 (Master PIC IR0) (Highest priority) 00010 = Priority P2 (Master PIC IR1) 00011 = Priority P3 (Slave 1 PIC IR0/Master PIC IR2) 00100 = Priority P4 (Slave 1 PIC IR1) 00101 = Priority P5 (Slave 1 PIC IR2) 00110 = Priority P6 (Slave 1 PIC IR3) 00111 = Priority P7 (Slave 1 PIC IR4) 01000 = Priority P8 (Slave 1 PIC IR5) 01001 = Priority P9 (Slave 1 PIC IR6) 01010 = Priority P10 (Slave 1 PIC IR7) 01011 = Priority P11 (Master PIC IR3) 01100 = Priority P12 (Master PIC IR4) 01101 = Priority P13 (Slave 2 PIC IR0/Master PIC IR5) 01110 = Priority P14 (Slave 2 PIC IR1) 01111 = Priority P15 (Slave 2 PIC IR2) 10000 = Priority P16 (Slave 2 PIC IR3) 10001 = Priority P17 (Slave 2 PIC IR4) 10010 = Priority P18 (Slave 2 PIC IR5) 10011 = Priority P19 (Slave 2 PIC IR6) 10100 = Priority P20 (Slave 2 PIC IR7) 10101 = Priority P21 (Master PIC IR6) 10110 = Priority P22 (Master PIC IR7) (Lowest priority) 10111 – 11110 = Disables the interrupt source as an input 11111 = NMI source For example, if INT_MAP = 01101b, the interrupt request is mapped to interrupt priority P13 in the microcontroller. If INT_MAP = 00000b or any binary value from 10111–11110b, the interrupt request is disabled from reaching the microcontroller's PIC. If this field is set to 11111b, then the interrupt is routed to generate an NMI. If bit S2 in the MPICICW3 register is cleared (see page 12-34), the Slave 1 PIC is bypassed, so programming the INT_MAP bit field to a value in the range 00100–01010b does not pass the interrupt request to the CPU. However, if this bit field is programmed to 00011b with the S2 bit cleared, the interrupt request is routed to the Master PIC IR2 input. If bit S5 in the MPICICW3 register is cleared (see page 12-33), the Slave 2 PIC is bypassed, so programming the INT_MAP bit field to a value in the range 01110–10100b does not pass the interrupt request to the CPU. However, if this field is programmed to 01101b with the S5 bit cleared, the interrupt request is routed to Master PIC IR5 input.

Programming Notes

This register should be programmed only when the corresponding interrupt channel mask bits are set in the PIC.

For NMIs, this register should be programmed only when bit NMI_ENB is cleared in the PICICR register (see page 12-4). NMI_ENB can be set immediately after programming this mapping register to allow NMIs to be passed to the CPU.

Programming more than one interrupt source to an interrupt channel results in interrupt sharing on that channel. Programming more than one interrupt source as an NMI source results in NMI sharing on the CPU's NMI input.

All interrupt and NMI sources can be found in the following register descriptions:

- Software Interrupt 16–1 Control, page 12-10
- Software Interrupt 22–17/NMI Control, page 12-13
- PCI Host Bridge Interrupt Mapping, page 12-17
- ECC Interrupt Mapping, page 12-19
- Other interrupt mapping registers (30), page 12-21

Master PIC Interrupt Request (MPICIR)

Direct-Mapped I/O Address 0020h

	7	6	5	4	3	2	1	0
Bit	IR7	IR6	IR5	IR4	IR3	IR2	IR1	IR0
Reset	x	x	x	x	x	x	x	x
R/W	R	R	R	R	R	R	R	R

Register Description

This register provides a real-time status of the interrupt request inputs to the Master PIC. This register latches all incoming interrupt requests and provides individual status of the requests to be acknowledged.

Bit Definitions

Bit	Name	Function
7	IR7	**Interrupt Request 7** 0 = The IR7 input to the Master PIC is not asserted. 1 = The IR7 input is asserted.
6	IR6	**Interrupt Request 6** 0 = The IR6 input to the Master PIC is not asserted. 1 = The IR6 input is asserted.
5	IR5	**Interrupt Request 5** 0 = The IR5 input to the Master PIC is not asserted. 1 = The IR5 input is asserted.
4	IR4	**Interrupt Request 4** 0 = The IR4 input to the Master PIC is not asserted. 1 = The IR4 input is asserted.
3	IR3	**Interrupt Request 3** 0 = The IR3 input to the Master PIC is not asserted. 1 = The IR3 input is asserted.
2	IR2	**Interrupt Request 2** 0 = The IR2 input to the Master PIC is not asserted. 1 = The IR2 input is asserted.
1	IR1	**Interrupt Request 1** 0 = The IR1 input to the Master PIC is not asserted. 1 = The IR1 input is asserted.
0	IR0	**Interrupt Request 0** 0 = The IR0 input to the Master PIC is not asserted. 1 = The IR0 input is asserted.

Programming Notes

This register (MPICIR) is accessed by first writing a value of 0Ah to Port 0020h followed by a read-back from Port 0020h.

Because the Slave 1 PIC cascades into Channel 2 of the Master PIC, the IR2 bit is a real-time status indication that one of the Slave 1 interrupt request inputs is asserted.

Because the Slave 2 PIC cascades into Channel 5 of the Master PIC, the IR5 bit is a real-time status indication that one of the Slave 2 interrupt request inputs is asserted.

Master PIC In-Service (MPICISR)

Direct-Mapped I/O Address 0020h

	7	6	5	4	3	2	1	0
Bit	IS7	IS6	IS5	IS4	IS3	IS2	IS1	IS0
Reset	x	x	x	x	x	x	x	x
R/W	R	R	R	R	R	R	R	R

Register Description

This register indicates the Master interrupt priority levels that are being serviced.

Bit Definitions

Bit	Name	Function
7	IS7	**Interrupt Request 7 In-Service** 0 = Interrupt request 7 is not being serviced. 1 = Interrupt request 7 is being serviced.
6	IS6	**Interrupt Request 6 In-Service** 0 = Interrupt request 6 is not being serviced. 1 = Interrupt request 6 is being serviced.
5	IS5	**Interrupt Request 5 In-Service** 0 = Interrupt request 5 is not being serviced. 1 = Interrupt request 5 is being serviced.
4	IS4	**Interrupt Request 4 In-Service** 0 = Interrupt request 4 is not being serviced. 1 = Interrupt request 4 is being serviced.
3	IS3	**Interrupt Request 3 In-Service** 0 = Interrupt request 3 is not being serviced. 1 = Interrupt request 3 is being serviced.
2	IS2	**Interrupt Request 2 In-Service** 0 = Interrupt request 2 is not being serviced. 1 = Interrupt request 2 is being serviced.
1	IS1	**Interrupt Request 1 In-Service** 0 = Interrupt request 1 is not being serviced. 1 = Interrupt request 1 is being serviced.
0	IS0	**Interrupt Request 0 In-Service** 0 = Interrupt request 0 is not being serviced. 1 = Interrupt request 0 is being serviced.

Programming Notes

This register (MPICISR) is accessed by first writing a value of 0Bh to Port 0020h followed by a read-back from Port 0020h.

Because the Slave 1 PIC cascades into Channel 2 of the Master PIC, the IS2 bit is asserted if any of the Slave 1 interrupt request levels is asserted.

Because the Slave 2 PIC cascades into Channel 5 of the Master PIC, the IS5 bit is asserted if any of the Slave 2 interrupt request levels is asserted.

Master PIC Initialization Control Word 1 (MPICICW1)

Direct-Mapped I/O Address 0020h

	7	6	5	4	3	2	1	0
Bit	Reserved			SLCT_ ICW1	LTIM	ADI	SNGL	IC4
Reset	x	x	x	x	x	1	x	x
R/W	RSV!			W	W	W	W	W

Register Description

This register is the first initialization register of the Master controller.

Bit Definitions

Bit	Name	Function
7–5	Reserved	**Reserved** This bit field should be written to 0 for normal system operation. This I/O address changes functions when read. See the programming notes for this register (MPICICW1) on page 12-27.
4	SLCT_ICW1	**Select ICW1** Software must set this bit to 1 when writing this address (Port 0020h) to access this register (MPICICW1). 0 = The write does not access this register (MPICICW1). Instead, either the MPICOCW2 register (see page 12-28) or the MPICOCW3 register (see page 12-30) is written, depending on the state of bit 3. 1 = The write accesses this register (MPICICW1). Subsequent writes to Port 0021h access additional initialization control words. See the programming notes for this register (MPICICW1) on page 12-27.
3	LTIM	**Level-Triggered Interrupt Mode** This bit is the global interrupt mode selection for the Master PIC. 0 = Edge-sensitive interrupt request detection 1 = Level-sensitive interrupt request detection If the M_GINT_MODE bit in the PICICR register is set (see page 12-5), the LTIM bit determines the interrupt mode for the Master PIC channels. If the M_GINT_MODE bit is cleared, the Master LTIM bit has no meaning, and the Master PIC channel modes can be programmed individually via the MPICMODE register (see page 12-6).
2	ADI	**Address Interval** 0 = Interrupt vectors are separated by eight locations (not valid in the ÉlanSC520 microcontroller). 1 = Interrupt vectors are separated by four locations. In the ÉlanSC520 microcontroller design, this PC/AT-compatible bit (ADI) is internally fixed to 1.
1	SNGL	**Single PIC** 0 = Cascade mode; MPICICW3 is expected. 1 = Single PIC in the system; MPICICW3 is not expected. Setting this bit logically removes the Slave 1 and Slave 2 controllers from the Master PIC. This routes the interrupt requests that were hooked to IR0 of the Slave 1 and Slave 2 controllers directly to interrupt requests IR2 and IR5 of the Master PIC, respectively. If this bit is set, then the internal register pointer skips MPICICW3 and points to MPICICW4 if MPICICW4 was selected to be programmed via the IC4 bit. See the programming notes for this register (MPICICW1) on page 12-27.

Bit	Name	Function
0	IC4	**Initialization Control Word 4** Software uses this bit to indicate whether it intends to explicitly program the MPICICW4 register (see page 12-35) after writing to the MPICICW3 register (see page 12-33). See the programming notes on this page for details. 0 = The MPICICW4 register is initialized internally when this register (MPICICW1) is written. The PIC does not expect software to write to the MPICICW4 register after writing to the MPICICW3 register. 1 = The MPICICW4 register is not initialized by the write to this register (MPICICW1). Software is expected to initialize the MPICICW4 register.

Programming Notes

The PIC's initialization control word (MPICICWx) registers 1–4 must be programmed in sequence. Writing to Port 0020h with bit 4 = 1 causes the MPICICW1 register to be written and also resets the PIC's internal state machine and the internal MPICICWx register pointer. Then MPICICWx registers 2–4 can be programmed by sequential writes to Port 0021h. Each time Port 0021h is written to (following the write to MPICICW1), the internal register pointer points to the next MPICICWx register. MPICICW1 and MPICICW2 must always be programmed. The MPICICW3 register is skipped if the SNGL bit in MPICICW1 is 1. The MPICICW4 register is skipped if the IC4 bit in MPICICW1 is 0.

I/O Port 0020h provides access to different Master PIC registers based on the data that is written. Table 12-3 provides a summary of bit patterns to write for access to each register.

Table 12-3 Master PIC I/O Port 0020h Access Summary

Bits									
7	6	5	4	3	2	1	0	Port 0020h Register Written	Next Port 0020h Read Returns
x	x	x	0	0	x	x	x	MPICOCW2 (page 12-28)	—
0	x	x	0	1	x	0	x	MPICOCW3 (page 12-30)	—
0	0	0	0	1	0	1	0	MPICOCW3	MPICIR (page 12-24)
0	0	0	0	1	0	1	1	MPICOCW3	MPICISR (page 12-25)
0	0	0	1	x	x	x	x	MPICICW1 (page 12-26)	—

Master PIC Operation Control Word 2 (MPICOCW2)

Direct-Mapped I/O Address 0020h

	7	6	5	4	3	2	1	0
Bit	R_SL_EOI[2–0]			SLCT_ ICW1	IS_OCW3	LS[2–0]		
Reset	x	x	x	x	x	x	x	x
R/W	W			W	W	W		

Register Description

This register provides control for various interrupt priority and end-of-interrupt (EOI) modes. It also controls write access for this register (MPICOCW2) and for the MPICOCW3 and MPICICW1 registers (see page 12-30 and page 12-26).

Bit Definitions

Bit	Name	Function
7–5	R_SL_EOI[2–0]	**Interrupt Request EOI and Priority Rotation Controls** 000 = Rotate in auto EOI mode (clear) 001 = Nonspecific EOI 010 = No operation 011 = Specific EOI 100 = Rotate in auto EOI mode (set) 101 = Rotate on nonspecific EOI command 110 = Set priority command 111 = Rotate on specific EOI command
4	SLCT_ICW1	**Select ICW1** Software must clear this bit to 0 when writing this address (Port 0020h) to access either this register (MPICOCW2) or the MPICOCW3 register. 0 = The write accesses either this register (MPICOCW2) or the MPICOCW3 register (see page 12-30), depending on the state of bit 3. 1 = The write accesses the MPICICW1 register (see page 12-26).
3	IS_OCW3	**Access is OCW3** Software must clear this bit (IS_OCW3) and clear SLCT_ICW1 when writing this address (Port 0020h) to access this register (MPICOCW2). 0 = The write accesses this register (MPICOCW2) if the SLCT_ICW1 bit is cleared. 1 = The write accesses the MPICOCW3 register (see page 12-30) if the SLCT_ICW1 bit is cleared.
2–0	LS[2–0]	**Specific EOI Level Select** Interrupt level that is acted upon when the SL bit = 1 (see bits 7–5 of this register): 000 = IR0 001 = IR1 010 = IR2 011 = IR3 100 = IR4 101 = IR5 110 = IR6 111 = IR7

Programming Notes

I/O Port 0020h provides access to different Master PIC registers based on the data that is written. Table 12-4 provides a summary of bit patterns to write for access to each register.

Table 12-4 Master PIC I/O Port 0020h Access Summary (Same as Table 12-3)

Bits									
7	6	5	4	3	2	1	0	Port 0020h Register Written	Next Port 0020h Read Returns
x	x	x	0	0	x	x	x	MPICOCW2 (page 12-28)	—
0	x	x	0	1	x	0	x	MPICOCW3 (page 12-30)	—
0	0	0	0	1	0	1	0	MPICOCW3	MPICIR (page 12-24)
0	0	0	0	1	0	1	1	MPICOCW3	MPICISR (page 12-25)
0	0	0	1	x	x	x	x	MPICICW1 (page 12-26)	—

Master PIC Operation Control Word 3 (MPICOCW3)

Direct-Mapped I/O Address 0020h

	7	6	5	4	3	2	1	0
Bit	Reserved	ESMM_SMM[1–0]		SLCT_ ICW1	IS_OCW3	P	RR_RIS[1–0]	
Reset	x	1	x	x	x	x	x	x
R/W	RSV!	W		W	W	W	W	

Register Description

This register controls the PIC's mask and poll modes. It also controls read access for the MPICIR and MPICISR registers (see page 12-24 and page 12-25), and write access for this register (MPICOCW3) and for the MPICOCW2 and MPICICW1 registers (see page 12-28 and page 12-26).

Bit Definitions

Bit	Name	Function
7	Reserved	**Reserved** This bit field should be written to 0 for normal system operation. This I/O address changes functions when read. See the programming notes for this register (MPICOCW3) on page 12-31.
6–5	ESMM_SMM [1–0]	**Special Mask Mode** 00 = No operation 01 = No operation 10 = Reset special mask 11 =Set special mask
4	SLCT_ICW1	**Initialization Control Word 1 Select** Software must clear this bit to 0 when writing this address (Port 0020h) to access either this register (MPICOCW3) or the MPICOCW2 register. 0 = The write accesses either this register (MPICOCW3) or the MPICOCW2 register (see page 12-28), depending on the state of bit 3. 1 = The write accesses the MPICICW1 register (see page 12-26).
3	IS_OCW3	**Access is OCW3** Software must set this bit (IS_OCW3) and clear SLCT_ICW1 when writing this address (Port 0020h) to access this register (MPICOCW3). 0 = The write accesses the MPICOCW2 register (see page 12-28) if the SLCT_ICW1 bit is cleared. 1 = The write accesses this register (MPICOCW3) if the SLCT_ICW1 bit is cleared.
2	P	**PIC Poll Command** A system designer can choose to use the PIC in a non-interrupting mode. In this case, the interrupt controller can be polled for the status of pending interrupts. To support this PC/AT-incompatible mode of operation, the PIC supports a special poll command that is invoked by setting this bit. 0 = Not poll command 1 = Poll command
1–0	RR_RIS[1–0]	**Status Register Select** 00 = No change from last state 01 = No change from last state 10 = Next Port 0020h read returns the MPICIR register's contents (see page 12-24). 11 = Next Port 0020h read returns the MPICISR register's contents (see page 12-25).

Programming Notes

I/O Port 0020h provides access to different Master PIC registers based on the data that is written. Table 12-5 provides a summary of bit patterns to write for access to each register.

Table 12-5 Master PIC I/O Port 0020h Access Summary (Same as Table 12-3)

Bits									
7	6	5	4	3	2	1	0	Port 0020h Register Written	Next Port 0020h Read Returns
x	x	x	0	0	x	x	x	MPICOCW2 (page 12-28)	—
0	x	x	0	1	x	0	x	MPICOCW3 (page 12-30)	—
0	0	0	0	1	0	1	0	MPICOCW3	MPICIR (page 12-24)
0	0	0	0	1	0	1	1	MPICOCW3	MPICISR (page 12-25)
0	0	0	1	x	x	x	x	MPICICW1 (page 12-26)	—

Master PIC Initialization Control Word 2 (MPICICW2)

Direct-Mapped I/O Address 0021h

	7	6	5	4	3	2	1	0
Bit	T7–T3					A10–A8		
Reset	x	x	x	x	x	x	x	x
R/W	W					W		

Register Description

This register is the second initialization register of the Master controller.

Bit Definitions

Bit	Name	Function
7–3	T7–T3	**Bits 7–3 of Base Interrupt Vector Number for this PIC** The PIC concatenates the T7–T3 bit field value to the 3-bit PIC interrupt request level (in the bit 2–0 position) to form the interrupt vector. For example, in a PC/AT-compatible system, bits T7–T3 for the Master PIC are programmed to 00001b so the Master PIC IR0 channel generates an interrupt 08h vector (PC/AT IRQ0); and bits T7–T3 for the Slave 1 PIC are programmed to 01110b so the Slave 1 PIC IR0 channel generates an interrupt 70h (PC/AT IRQ8).
2–0	A10–A8	**A10–A8 of Interrupt Vector** This bit field should be written to 0 for normal operation. (It is always = 0 in a PC/AT-compatible system.)

Programming Notes

The PIC's initialization control word (MPICICWx) registers 1–4 must be programmed in sequence. Writing to Port 0020h with bit 4 = 1 causes the MPICICW1 register to be written and also resets the PIC's internal state machine and the internal MPICICWx register pointer. Then MPICICWx registers 2–4 can be programmed by sequential writes to Port 0021h. Each time Port 0021h is written to (following the write to MPICICW1), the internal register pointer points to the next MPICICWx register. MPICICW1 and MPICICW2 must always be programmed. The MPICICW3 register is skipped if the SNGL bit in MPICICW1 is 1. The MPICICW4 register is skipped if the IC4 bit in MPICICW1 is 0.

Master PIC Initialization Control Word 3 (MPICICW3)

Direct-Mapped
I/O Address 0021h

	7	6	5	4	3	2	1	0
Bit	S7	S6	S5	S4	S3	S2	S1	S0
Reset	0	0	x	0	0	x	0	0
R/W	W	W	W	W	W	W	W	W

Register Description

This register is the 3rd initialization register of the Master controller.

Bit Definitions

Bit	Name	Function
7	S7	**Channel 7 Slave Cascade Select** 0 = I/O device attached to IR7 input 1 = IR7 input used for slave cascading (not valid in the ÉlanSC520 microcontroller) In the ÉlanSC520 microcontroller, this bit is internally fixed to 0.
6	S6	**Channel 6 Slave Cascade Select** 0 = I/O device attached to IR6 input 1 = IR6 input used for slave cascading (not valid in the ÉlanSC520 microcontroller) In the ÉlanSC520 microcontroller, this bit is internally fixed to 0.
5	S5	**Channel 5 Slave Cascade Select** 0 = I/O device attached to IR5 input 1 = IR5 input used for slave cascading This bit allows the Slave 2 controller to be logically cascaded or removed from the Master PIC. Since the output of the Slave 2 controller is hard-wired to interrupt channel 5 of the Master PIC, this bit can be used to make the Slave 2 controller transparent to the user. If this bit is set, Slave 2 controller is cascaded to the Master PIC. If this bit is cleared, Slave 2 is logically removed from the cascade chain. The interrupt request that is hooked to IR0 of the Slave 2 controller is now routed directly to interrupt channel 5 of the Master PIC, bypassing the Slave 2 controller. This is useful where fewer than eight interrupts are needed in a system. In this case, a single EOI would need to be generated instead of two.
4	S4	**Channel 4 Slave Cascade Select** 0 = I/O device attached to IR4 input 1 = IR4 input used for slave cascading (not valid in the ÉlanSC520 microcontroller) In the ÉlanSC520 microcontroller, this bit is internally fixed to 0.

Bit	Name	Function
3	S3	**Channel 3 Slave Cascade Select** 0 = I/O device attached to IR3 input 1 = IR3 input used for slave cascading (not valid in the ÉlanSC520 microcontroller) In the ÉlanSC520 microcontroller, this bit is internally fixed to 0.
2	S2	**Channel 2 Slave Cascade Select** 0 = I/O device attached to IR2 input 1 = IR2 input used for slave cascading This bit allows the Slave 1 controller to be logically cascaded or removed from the Master PIC. Since the output of the Slave 1 controller is hard-wired to interrupt channel 2 of the Master PIC, this bit can be used to make the Slave 1 controller transparent to the user. If this bit is set, Slave 1 controller is cascaded to the Master PIC. If this bit is cleared, Slave 1 is logically removed from the cascade chain. The interrupt request that is hooked to IR0 of the Slave 1 controller is now routed directly to interrupt channel 2 of the Master PIC, bypassing the Slave 1 controller. This is useful where fewer than eight interrupts are needed in a system. In this case, a single EOI would need to be generated instead of two.
1	S1	**Channel 1 Slave Cascade Select** 0 = I/O device attached to IR1 input 1 = IR1 input used for slave cascading (not valid in the ÉlanSC520 microcontroller) In the ÉlanSC520 microcontroller, this bit is internally fixed to 0.
0	S0	**Channel 0 Slave Cascade Select** 0 = I/O device attached to IR0 input 1 = IR0 input used for slave cascading (not valid in the ÉlanSC520 microcontroller) In the ÉlanSC520 microcontroller, this bit is internally fixed to 0.

Programming Notes

If bits S5 and S2 of this register (MPICICW3) are cleared, both the slave controllers are logically removed from the cascade chain to the Master controller and only eight interrupt request priority levels are available to the user.

The PIC's initialization control word (MPICICWx) registers 1–4 must be programmed in sequence. Writing to Port 0020h with bit 4 = 1 causes the MPICICW1 register to be written and also resets the PIC's internal state machine and the internal MPICICWx register pointer. Then MPICICWx registers 2–4 can be programmed by sequential writes to Port 0021h. Each time Port 0021h is written to (following the write to MPICICW1), the internal register pointer points to the next MPICICWx register. MPICICW1 and MPICICW2 must always be programmed. The MPICICW3 register is skipped if the SNGL bit in MPICICW1 is 1. The MPICICW4 register is skipped if the IC4 bit in MPICICW1 is 0.

Master PIC Initialization Control Word 4 (MPICICW4)

Direct-Mapped I/O Address 0021h

	7	6	5	4	3	2	1	0
Bit	Reserved			SFNM	BUF_M/S[1–0]		AEOI	PM
Reset	x	x	x	x	0	0	x	1
R/W	RSV!			W	W		W	W

Register Description

This register is the fourth initialization register of the Master controller.

Bit Definitions

Bit	Name	Function
7–5	Reserved	**Reserved** This bit field should be written to 0 for normal system operation. This bit field is write-only.
4	SFNM	**Special Fully Nested Mode Enable** 0 = Normal nested mode 1 = Special fully nested mode
3–2	BUF_M/S[1–0]	**Buffered Mode and Master/Slave Select** 00 = Non-buffered mode 01 = Non-buffered mode 10 = Buffered mode/slave 11 = Buffered mode/Master In the ÉlanSC520 microcontroller design, these bits are internally fixed to 00b.
1	AEOI	**Automatic EOI Mode** 0 = Normal EOI: the interrupt handler must send an End of Interrupt command to the PIC(s). 1 = Auto EOI: the EOI is automatically performed after the second interrupt acknowledge signal from the CPU.
0	PM	**Microprocessor Mode** 0 = 8080/8085 mode 1 = 8086 mode In the ÉlanSC520 microcontroller design, this PC/AT-compatible bit is internally fixed to 1.

Programming Notes

Initialization of this register is optional unless the IC4 bit is set in the MPICICW1 register (see page 12-27). If the IC4 bit is cleared, the ÉlanSC520 microcontroller uses 01h for the value of this register (MPICICW4).

The PIC's initialization control word (MPICICWx) registers 1–4 must be programmed in sequence. Writing to Port 0020h with bit 4 = 1 causes the MPICICW1 register to be written and also resets the PIC's internal state machine and the internal MPICICWx register pointer. Then MPICICWx registers 2–4 can be programmed by sequential writes to Port 0021h. Each time Port 0021h is written to (following the write to MPICICW1), the internal register pointer points to the next MPICICWx register. MPICICW1 and MPICICW2 must always be programmed. The MPICICW3 register is skipped if the SNGL bit in MPICICW1 is 1. The MPICICW4 register is skipped if the IC4 bit in MPICICW1 is 0.

Note that AEOI mode is not supported by the Slave 1 PIC or Slave 2 PIC.

Master PIC Interrupt Mask (MPICINTMSK)

Direct-Mapped I/O Address 0021h

	7	6	5	4	3	2	1	0
Bit	IM7	IM6	IM5	IM4	IM3	IM2	IM1	IM0
Reset	x	x	x	x	x	x	x	x
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Register Description

This register provides masking of individual interrupt requests for the Master controller. This register is also known as Operation Control Word 1 in other PC/AT-compatible designs.

Bit Definitions

Bit	Name	Function
7	IM7	**IR7 Mask** 0 = Unmask IR7 1 = Mask IR7
6	IM6	**IR6 Mask** 0 = Unmask IR6 1 = Mask IR6
5	IM5	**IR5 Mask** 0 = Unmask IR5 1 = Mask IR5 If the S5 bit in the MPICICW3 register is set (see page 12-33), setting the IM5 bit inhibits interrupt requests assigned to Slave 2 controller inputs from reaching the CPU. Clearing this bit allows interrupts from Slave 2 controller to be propagated to the CPU.
4	IM4	**IR4 Mask** 0 = Unmask IR4 1 = Mask IR4
3	IM3	**IR3 Mask** 0 = Unmask IR3 1 = Mask IR3
2	IM2	**IR2 Mask** 0 = Unmask IR2 1 = Mask IR2 If the S2 bit in the MPICICW3 register is set (see page 12-34), setting the IM2 bit inhibits interrupt requests assigned to Slave 1 controller inputs from reaching the CPU. Clearing this bit allows interrupts from Slave 1 controller to be propagated to the CPU.
1	IM1	**IR1 Mask** 0 = Unmask IR1 1 = Mask IR1
0	IM0	**IR0 Mask** 0 = Unmask IR0 1 = Mask IR0

Programming Notes

This register (MPICINTMSK) cannot be accessed during a Master PIC initialization control sequence, which is initiated by setting the SLCT_ICW1 bit in the MPICICW1 register (see page 12-26). When the MPICICWx register initialization sequence is not in effect, any read or write of Port 0021h accesses the MPICINTMSK register.

Slave 2 PIC Interrupt Request (S2PICIR)

Direct-Mapped I/O Address 0024h

	7	6	5	4	3	2	1	0
Bit	IR7	IR6	IR5	IR4	IR3	IR2	IR1	IR0
Reset	x	x	x	x	x	x	x	x
R/W	R	R	R	R	R	R	R	R

Register Description

This register provides a real-time status of the interrupt request inputs to the Slave 2 PIC. This register latches all incoming interrupt requests and provides individual status of the requests to be acknowledged.

Bit Definitions

Bit	Name	Function
7	IR7	**Interrupt Request 7** 0 = The IR7 input to the Slave 2 PIC is not asserted. 1 = The IR7 input is asserted.
6	IR6	**Interrupt Request 6** 0 = The IR6 input to the Slave 2 PIC is not asserted. 1 = The IR6 input is asserted.
5	IR5	**Interrupt Request 5** 0 = The IR5 input to the Slave 2 PIC is not asserted. 1 = The IR5 input is asserted.
4	IR4	**Interrupt Request 4** 0 = The IR4 input to the Slave 2 PIC is not asserted. 1 = The IR4 input is asserted.
3	IR3	**Interrupt Request 3** 0 = The IR3 input to the Slave 2 PIC is not asserted. 1 = The IR3 input is asserted.
2	IR2	**Interrupt Request 2** 0 = The IR2 input to the Slave 2 PIC is not asserted. 1 = The IR2 input is asserted.
1	IR1	**Interrupt Request 1** 0 = The IR1 input to the Slave 2 PIC is not asserted. 1 = The IR1 input is asserted.
0	IR0	**Interrupt Request 0** 0 = The IR0 input to the Slave 2 PIC is not asserted. 1 = The IR0 input is asserted.

Programming Notes

This register (S2PICIR) is accessed by first writing a value of 0Ah to Port 0024h followed by a read-back from Port 0024h.

If the S5 bit in the MPICICW3 register is cleared (see page 12-33), then the Slave 2 controller is bypassed and any interrupt requests latched in this register (S2PICIR) are not propagated to the CPU.

Slave 2 PIC In-Service (S2PICISR)

Direct-Mapped I/O Address 0024h

	7	6	5	4	3	2	1	0
Bit	IS7	IS6	IS5	IS4	IS3	IS2	IS1	IS0
Reset	x	x	x	x	x	x	x	x
R/W	R	R	R	R	R	R	R	R

Register Description

This register indicates the Slave 2 interrupt priority levels that are being serviced.

Bit Definitions

Bit	Name	Function
7	IS7	**Interrupt Request 7 In-Service** 0 = Interrupt request 7 is not being serviced. 1 = Interrupt request 7 is being serviced.
6	IS6	**Interrupt Request 6 In-Service** 0 = Interrupt request 6 is not being serviced. 1 = Interrupt request 6 is being serviced.
5	IS5	**Interrupt Request 5 In-Service** 0 = Interrupt request 5 is not being serviced. 1 = Interrupt request 5 is being serviced.
4	IS4	**Interrupt Request 4 In-Service** 0 = Interrupt request 4 is not being serviced. 1 = Interrupt request 4 is being serviced.
3	IS3	**Interrupt Request 3 In-Service** 0 = Interrupt request 3 is not being serviced. 1 = Interrupt request 3 is being serviced.
2	IS2	**Interrupt Request 2 In-Service** 0 = Interrupt request 2 is not being serviced. 1 = Interrupt request 2 is being serviced.
1	IS1	**Interrupt Request 1 In-Service** 0 = Interrupt request 1 is not being serviced. 1 = Interrupt request 1 is being serviced.
0	IS0	**Interrupt Request 0 In-Service** 0 = Interrupt request 0 is not being serviced. 1 = Interrupt request 0 is being serviced.

Programming Notes

This register (S2PICISR) is accessed by first writing a value of 0Bh to Port 0024h followed by a read-back from Port 0024h.

If the S5 bit in the MPICICW3 register is cleared (see page 12-33), then the Slave 2 controller is bypassed and interrupt requests latched are not serviced.

Slave 2 PIC Initialization Control Word 1 (S2PICICW1)

Direct-Mapped I/O Address 0024h

	7	6	5	4	3	2	1	0
Bit	Reserved			SLCT_ ICW1	LTIM	ADI	SNGL	IC4
Reset	x	x	x	x	x	1	0	x
R/W	RSV!			W	W	W	W	W

Register Description

This register is the first initialization register of the Slave 2 controller.

Bit Definitions

Bit	Name	Function
7–5	Reserved	**Reserved** This bit field should be written to 0 for normal system operation. This I/O address changes functions when read. See the programming notes for this register (S2PICICW1) on page 12-40.
4	SLCT_ICW1	**Initialization Control Word 1 Select** Software must set this bit to 1 when writing this address (Port 0024h) to access this register (S2PICICW1). 0 = The write does not access this register (S2PICICW1). Instead, either the S2PICOCW2 register (see page 12-41) or the S2PICOCW3 register (see page 12-43) is written, depending on the state of bit 3. 1 = The write accesses this register (S2PICICW1). Subsequent writes to Port 0025h access additional initialization control words. See the programming notes for this register (S2PICICW1) on page 12-40.
3	LTIM	**Level-Triggered Interrupt Mode** This bit is the global interrupt mode selection for the Slave 2 PIC. 0 = Edge-sensitive interrupt request detection 1 = Level-sensitive interrupt request detection If the S2_GINT_MODE bit in the PICICR register is set (see page 12-4), the LTIM bit determines the interrupt mode for the Slave 2 PIC channels. If the S2_GINT_MODE bit is cleared, the Slave 2 LTIM bit has no meaning, and the Slave 2 PIC channel modes can be programmed individually via the SL2PICMODE register (see page 12-9).
2	ADI	**Address Interval** 0 = Interrupt vectors are separated by eight locations. 1 = Interrupt vectors are separated by four locations. In the ÉlanSC520 microcontroller design, this bit is internally fixed to 1.

Bit	Name	Function
1	SNGL	**Single PIC** 0 = Cascade mode; S2PICICW3 is expected. 1 = Single PIC in the system; S2PICICW3 is not expected (not valid in the ÉlanSC520 microcontroller). In the ÉlanSC520 microcontroller design, this bit is internally fixed to 0. Because this bit is internally fixed to 0, software must always write the S2PICICW3 register after writing S1PICICW2. See the programming notes on this page for details.
0	IC4	**Initialization Control Word 4** Software uses this bit to indicate whether it intends to explicitly program the S2PICICW4 register (see page 12-47) after writing to the S2PICICW3 register (see page 12-46). See the programming notes on this page for details. 0 = The S2PICICW4 register is initialized internally when this register (S2PICICW1) is written. The PIC does not expect software to write to the S2PICICW4 register. 1 = The S2PICICW4 register is not initialized by the write to this register (S2PICICW1). Software is expected to initialize the S2PICICW4 register after writing to the S2PICICW3 register.

Programming Notes

The PIC's initialization control word (S2PICICWx) registers 1–4 must be programmed in sequence. Writing to Port 0024h with bit 4 = 1 causes the S2PICICW1 register to be written and also resets the PIC's internal state machine and the internal S2PICICWx register pointer. Then, S2PICICWx registers 2–4 can be programmed by sequential writes to Port 0025h. Each time Port 0025h is written to (following the write to S2PICICW1), the internal register pointer points to the next S2PICICWx register. S2PICICW1 and S2PICICW2 must always be programmed. Also, the S2PICICW3 register must always be programmed in this design because the SNGL bit in S2PICICW1 is internally fixed to 0. The S2PICICW4 register is skipped if the IC4 bit in S2PICICW1 is 0.

If the S5 bit in the MPICICW3 register is cleared (see page 12-33), then the Slave 2 controller is bypassed and programming this register does not affect other registers.

I/O Port 0024h provides access to different Slave 2 PIC registers based on the data that is written. Table 12-6 provides a summary of bit patterns to write for access to each register.

Table 12-6 Slave 2 PIC I/O Port 0024h Access Summary

Bits									
7	6	5	4	3	2	1	0	Port 0024h Register Written	Next Port 0024h Read Returns
x	x	x	0	0	x	x	x	S2PICOCW2 (page 12-41)	—
0	x	x	0	1	x	0	x	S2PICOCW3 (page 12-43)	—
0	0	0	0	1	0	1	0	S2PICOCW3	S2PICIR (page 12-37)
0	0	0	0	1	0	1	1	S2PICOCW3	S2PICISR (page 12-38)
0	0	0	1	x	x	x	x	S2PICICW1 (page 12-39)	—

Slave 2 PIC Operation Control Word 2 (S2PICOCW2)

Direct-Mapped I/O Address 0024h

	7	6	5	4	3	2	1	0
Bit	R_SL_EOI[2–0]			SLCT_ ICW1	IS_OCW3	LS[2–0]		
Reset	x	x	x	x	x	x	x	x
R/W	W			W	W	W		

Register Description

This register provides control for various interrupt priority and end-of-interrupt (EOI) modes. It also controls write access for this register (S2PICOCW2) and for the S2PICOCW3 and S2PICICW1 registers (see page 12-43 and page 12-39).

Bit Definitions

Bit	Name	Function
7–5	R_SL_EOI[2–0]	**Interrupt Request EOI and Priority Rotation Controls** 000 = Rotate in auto EOI mode (clear) 001 = Nonspecific EOI 010 = No operation 011 = Specific EOI 100 = Rotate in auto EOI mode (set) 101 = Rotate on nonspecific EOI command 110 = Set priority command 111 = Rotate on specific EOI command
4	SLCT_ICW1	**Initialization Control Word 1 Select** Software must clear this bit to 0 when writing this address (Port 0024h) to access either this register (S2PICOCW2) or the S2PICOCW3 register. 0 = The write accesses either this register (S2PICOCW2) or the S2PICOCW3 register (see page 12-43), depending on the state of bit 3. 1 = The write accesses the S2PICICW1 register (see page 12-39).
3	IS_OCW3	**Access is OCW3** Software must clear this bit (IS_OCW3) and clear SLCT_ICW1 when writing this address (Port 0024h) to access this register (S2PICOCW2). 0 = The write accesses this register (S2PICOCW2) if the SLCT_ICW1 bit is cleared. 1 = The write accesses the S2PICOCW3 register (see page 12-43) if the SLCT_ICW1 bit is cleared.
2–0	LS[2–0]	**Specific EOI Level Select** Interrupt level that is acted upon when the SL bit = 1 (see bits 7–5 of this register): 000 = IR0 001 = IR1 010 = IR2 011 = IR3 100 = IR4 101 = IR5 110 = IR6 111 = IR7

Programming Notes

If the S5 bit in the MPICICW3 register is cleared (see page 12-33), then the Slave 2 controller is bypassed and programming this register does not affect other registers.

I/O Port 0024h provides access to different Slave 2 PIC registers based on the data that is written. Table 12-7 provides a summary of bit patterns to write for access to each register.

Table 12-7 Slave 2 PIC I/O Port 0024h Access Summary (Same as Table 12-6)

Bits									
7	6	5	4	3	2	1	0	Port 0024h Register Written	Next Port 0024h Read Returns
x	x	x	0	0	x	x	x	S2PICOCW2 (page 12-41)	—
0	x	x	0	1	x	0	x	S2PICOCW3 (page 12-43)	—
0	0	0	0	1	0	1	0	S2PICOCW3	S2PICIR (page 12-37)
0	0	0	0	1	0	1	1	S2PICOCW3	S2PICISR (page 12-38)
0	0	0	1	x	x	x	x	S2PICICW1 (page 12-39)	—

Slave 2 PIC Operation Control Word 3 (S2PICOCW3)

Direct-Mapped I/O Address 0024h

	7	6	5	4	3	2	1	0
Bit	Reserved	ESMM_SMM[1–0]		SLCT_ ICW1	IS_OCW3	P	RR_RIS[1–0]	
Reset	x	1	x	x	x	x	x	x
R/W	RSV!	W		W	W	W	W	

Register Description

This register controls the PIC's mask and poll modes. It also controls read access for the S2PICIR and S2PICISR registers (see page 12-37 and page 12-38), and write access for this register (S2PICOCW3) and for the S2PICOCW2 and S2PICICW1 registers (see page 12-41 and page 12-39).

Bit Definitions

Bit	Name	Function
7	Reserved	**Reserved** This bit field should be written to 0 for normal system operation. This I/O address changes functions when read. See the programming notes for this register (S2PICOCW3) on page 12-44.
6–5	ESMM_SMM [1–0]	**Special Mask Mode** 00 = No operation 01 = No operation 10 = Reset special mask 11 =Set special mask
4	SLCT_ICW1	**Initialization Control Word 1 Select** Software must clear this bit to 0 when writing this address (Port 0024h) to access either this register (S2PICOCW3) or the S2PICOCW2 register. 0 = The write accesses either this register (S2PICOCW3) or the S2PICOCW2 register (see page 12-41), depending on the state of bit 3. 1 = The write accesses the S2PICICW1 register (see page 12-39).
3	IS_OCW3	**Access is OCW3** Software must set this bit (IS_OCW3) and clear SLCT_ICW1 when writing this address (Port 0024h) to access this register (S2PICOCW3). 0 = The write accesses the S2PICOCW2 register (see page 12-41) if the SLCT_ICW1 bit is cleared. 1 = The write accesses this register (S2PICOCW3) if the SLCT_ICW1 bit is cleared.
2	P	**PIC Poll Command** A system designer can choose to use the PIC in a non-interrupting mode. In this case, the interrupt controller can be polled for the status of pending interrupts. To support this mode of operation, the PIC supports a special poll command that is invoked by setting this bit. 0 = Not poll command 1 = Poll command
1–0	RR_RIS[1–0]	**Status Register Select** 00 = No change from last state 01 = No change from last state 10 = Next Port 0024h read returns the S2PICIR register's contents (see page 12-37). 11 = Next Port 0024h read returns in the S2PICISR register's contents (see page 12-38).

Programming Notes

If the S5 bit in the MPICICW3 register is cleared (see page 12-33), then the Slave 2 controller is bypassed and programming this register does not affect other registers.

I/O Port 0024h provides access to different Slave 2 PIC registers based on the data that is written. Table 12-8 provides a summary of bit patterns to write for access to each register.

Table 12-8 Slave 2 PIC I/O Port 0024h Access Summary (Same as Table 12-6)

Bits									
7	6	5	4	3	2	1	0	Port 0024h Register Written	Next Port 0024h Read Returns
x	x	x	0	0	x	x	x	S2PICOCW2 (page 12-41)	—
0	x	x	0	1	x	0	x	S2PICOCW3 (page 12-43)	—
0	0	0	0	1	0	1	0	S2PICOCW3	S2PICIR (page 12-37)
0	0	0	0	1	0	1	1	S2PICOCW3	S2PICISR (page 12-38)
0	0	0	1	x	x	x	x	S2PICICW1 (page 12-39)	—

Slave 2 PIC Initialization Control Word 2 (S2PICICW2)

Direct-Mapped I/O Address 0025h

	7	6	5	4	3	2	1	0
Bit	T7–T3					A10–A8		
Reset	x	x	x	x	x	x	x	x
R/W	W					W		

Register Description

This register is the second initialization register of the Slave 2 controller.

Bit Definitions

Bit	Name	Function
7–3	T7–T3	**Bits 7–3 of Base Interrupt Vector Number for this PIC** The PIC concatenates the T7–T3 bit field value to the 3-bit PIC interrupt request level (in the bit 2–0 position) to form the interrupt vector. For example, if bits T7–T3 are programmed to 11110b, the IR0 channel generates an interrupt F0h vector.
2–0	A10–A8	**A10–A8 of Interrupt Vector** Software should write these to 0.

Programming Notes

If the S5 bit in the MPICICW3 register is cleared (see page 12-33), then the Slave 2 controller is bypassed and programming this register does not affect other registers.

The PIC's initialization control word (S2PICICWx) registers 1–4 must be programmed in sequence. Writing to Port 0024h with bit 4 = 1 causes the S2PICICW1 register to be written and also resets the PIC's internal state machine and the internal S2PICICWx register pointer. Then, S2PICICWx registers 2–4 can be programmed by sequential writes to Port 0025h. Each time Port 0025h is written to (following the write to S2PICICW1), the internal register pointer points to the next S2PICICWx register. S2PICICW1 and S2PICICW2 must always be programmed. Also, the S2PICICW3 register must always be programmed in this design because the SNGL bit in S2PICICW1 is internally fixed to 0. The S2PICICW4 register is skipped if the IC4 bit in S2PICICW1 is 0.

Slave 2 PIC Initialization Control Word 3 (S2PICICW3)

Direct-Mapped I/O Address 0025h

	7	6	5	4	3	2	1	0
Bit	Reserved					ID2–ID0		
Reset	x	x	x	x	x	1	0	1
R/W	RSV!					W		

Register Description

This register is the third initialization register of the Slave 2 controller.

Bit Definitions

Bit	Name	Function
7–3	Reserved	**Reserved** This bit field should be written to 0 for normal system operation. This bit field is write-only.
2–0	ID2–ID0	**Slave 2 PIC ID 2–0** These bits contain the binary Slave 2 PIC ID (000b–111b) that the PIC responds to on the cascade bus. In the ÉlanSC520 microcontroller, these bits are internally fixed to 101b.

Programming Notes

The PIC's initialization control word (S2PICICWx) registers 1–4 must be programmed in sequence. Writing to Port 0024h with bit 4 = 1 causes the S2PICICW1 register to be written and also resets the PIC's internal state machine and the internal S2PICICWx register pointer. Then, S2PICICWx registers 2–4 can be programmed by sequential writes to Port 0025h. Each time Port 0025h is written to (following the write to S2PICICW1), the internal register pointer points to the next S2PICICWx register. S2PICICW1 and S2PICICW2 must always be programmed. Also, the S2PICICW3 register must always be programmed in this design because the SNGL bit in S2PICICW1 is internally fixed to 0. The S2PICICW4 register is skipped if the IC4 bit in S2PICICW1 is 0.

If the S5 bit is set in the MPICICW3 register (see page 12-33), a write to this register (S2PICICW3) is always expected in the ÉlanSC520 microcontroller because the SNGL bit is fixed to 0 in the S2PICICW1 register (page 12-40).

If the S5 bit is cleared in the MPICICW3 register, then the Slave 2 controller is bypassed and the value of this register (S2PICICW3) has no effect.

Slave 2 PIC Initialization Control Word 4 (S2PICICW4)

Direct-Mapped I/O Address 0025h

	7	6	5	4	3	2	1	0
Bit	Reserved			SFNM	BUF_M/S[1–0]		AEOI	PM
Reset	x	x	x	x	0	0	0	1
R/W	RSV!			W	W		W	W

Register Description

This register is the fourth initialization register of the Slave 2 controller.

Bit Definitions

Bit	Name	Function
7–5	Reserved	**Reserved** This bit field should be written to 0 for normal system operation. This bit field is write-only.
4	SFNM	**Special Fully Nested Mode Enable** 0 = Normal nested mode 1 = Special fully nested mode
3–2	BUF_M/S[1–0]	**Buffered Mode and Master/Slave Select** 00 = Non buffered mode 01 = Non buffered mode 10 = Buffered mode/Slave 11 = Buffered mode/Master In the ÉlanSC520 microcontroller, these bits are internally fixed to 00b.
1	AEOI	**Automatic EOI Mode** 0 = Normal EOI: the interrupt handler must send an End of Interrupt command to the PIC(s). 1 = Auto EOI: the EOI is automatically performed after the second interrupt acknowledge signal from the CPU. In the ÉlanSC520 microcontroller, this bit is internally fixed to 0. The Slave 1 PIC and Slave 2 PIC do not support automatic EOI mode.
0	PM	**Microprocessor Mode** 0 = 8080/8085 mode 1 = 8086 mode In the ÉlanSC520 microcontroller design, this bit is internally fixed to 1.

Programming Notes

The PIC's initialization control word (S2PICICWx) registers 1–4 must be programmed in sequence. Writing to Port 0024h with bit 4 = 1 causes the S2PICICW1 register to be written and also resets the PIC's internal state machine and the internal S2PICICWx register pointer. Then, S2PICICWx registers 2–4 can be programmed by sequential writes to Port 0025h. Each time Port 0025h is written to (following the write to S2PICICW1), the internal register pointer points to the next S2PICICWx register. S2PICICW1 and S2PICICW2 must always be programmed. Also, the S2PICICW3 register must always be programmed in this design because the SNGL bit in S2PICICW1 is internally fixed to 0. The S2PICICW4 register is skipped if the IC4 bit in S2PICICW1 is 0.

Software is expected to initialize this register (S2PICICW4) if the IC4 bit is set in the S2PICICW1 register (see page 12-40). If the IC4 bit is cleared, the ÉlanSC520 microcontroller uses 01h for the value of this register (S2PICICW4).

However, if the S5 bit in the MPICICW3 register is cleared (see page 12-33), then the Slave 2 controller is bypassed and the value of this register (S2PICICW4) has no effect.

Slave 2 PIC Interrupt Mask (S2PICINTMSK)

Direct-Mapped I/O Address 0025h

	7	6	5	4	3	2	1	0
Bit	IM7	IM6	IM5	IM4	IM3	IM2	IM1	IM0
Reset	x	x	x	x	x	x	x	x
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Register Description

This register provides masking of individual interrupt requests for the Slave 2 controller. This register is also known as Operation Control Word 1 in other PC/AT-compatible designs.

Bit Definitions

Bit	Name	Function
7	IM7	**IR7 Mask** 0 = Unmask IR7 1 = Mask IR7
6	IM6	**IR6 Mask** 0 = Unmask IR6 1 = Mask IR6
5	IM5	**IR5 Mask** 0 = Unmask IR5 1 = Mask IR5
4	IM4	**IR4 Mask** 0 = Unmask IR4 1 = Mask IR4
3	IM3	**IR3 Mask** 0 = Unmask IR3 1 = Mask IR3
2	IM2	**IR2 Mask** 0 = Unmask IR2 1 = Mask IR2
1	IM1	**IR1 Mask** 0 = Unmask IR1 1 = Mask IR1
0	IM0	**IR0 Mask** 0 = Unmask IR0 1 = Mask IR0

Programming Notes

If the S5 bit in the MPICICW3 register is cleared (see page 12-33), then the Slave 2 controller is bypassed and the value of this register (S2PICINTMSK) has no effect.

This register (S2PICINTMSK) cannot be accessed during a Slave 2 PIC initialization control sequence, which is initiated by setting the SLCT_ICW1 bit in the S2PICICW1 register (see page 12-39).

When the S2PICICWx register initialization sequence is not in effect, any read or write of Port 0025h accesses the S2PICINTMSK register.

Slave 1 PIC Interrupt Request (S1PICIR)

Direct-Mapped I/O Address 00A0h

	7	6	5	4	3	2	1	0
Bit	IR7	IR6	IR5	IR4	IR3	IR2	IR1	IR0
Reset	x	x	x	x	x	x	x	x
R/W	R	R	R	R	R	R	R	R

Register Description

This register provides a real-time status of the interrupt request inputs to the Slave 1 PIC. This register latches all incoming interrupt requests and provides individual status of the requests to be acknowledged.

Bit Definitions

Bit	Name	Function
7	IR7	**Interrupt Request 7** 0 = The IR7 input to the Slave 1 PIC is not asserted. 1 = The IR7 input is asserted.
6	IR6	**Interrupt Request 6** 0 = The IR6 input to the Slave 1 PIC is not asserted. 1 = The IR6 input is asserted.
5	IR5	**Interrupt Request 5** 0 = The IR5 input to the Slave 1 PIC is not asserted. 1 = The IR5 input is asserted.
4	IR4	**Interrupt Request 4** 0 = The IR4 input to the Slave 1 PIC is not asserted. 1 = The IR4 input is asserted.
3	IR3	**Interrupt Request 3** 0 = The IR3 input to the Slave 1 PIC is not asserted. 1 = The IR3 input is asserted.
2	IR2	**Interrupt Request 2** 0 = The IR2 input to the Slave 1 PIC is not asserted. 1 = The IR2 input is asserted.
1	IR1	**Interrupt Request 1** 0 = The IR1 input to the Slave 1 PIC is not asserted. 1 = The IR1 input is asserted.
0	IR0	**Interrupt Request 0** 0 = The IR0 input to the Slave 1 PIC is not asserted. 1 = The IR0 input is asserted.

Programming Notes

This register (S1PICIR) is accessed by first writing a value of 0Ah to Port 00A0h followed by a read-back from Port 00A0h.

If the S2 bit in the MPICICW3 register is cleared (see page 12-34), then the Slave 1 controller is bypassed and any interrupt requests latched in this register (S1PICIR) are not propagated to the CPU.

Slave 1 PIC In-Service (S1PICISR)

Direct-Mapped I/O Address 00A0h

	7	6	5	4	3	2	1	0
Bit	IS7	IS6	IS5	IS4	IS3	IS2	IS1	IS0
Reset	x	x	x	x	x	x	x	x
R/W	R	R	R	R	R	R	R	R

Register Description

This register indicates the Slave 1 interrupt priority levels that are being serviced.

Bit Definitions

Bit	Name	Function
7	IS7	**Interrupt Request 7 In-Service** 0 = Interrupt request 7 is not being serviced. 1 = Interrupt request 7 is being serviced.
6	IS6	**Interrupt Request 6 In-Service** 0 = Interrupt request 6 is not being serviced. 1 = Interrupt request 6 is being serviced.
5	IS5	**Interrupt Request 5 In-Service** 0 = Interrupt request 5 is not being serviced. 1 = Interrupt request 5 is being serviced.
4	IS4	**Interrupt Request 4 In-Service** 0 = Interrupt request 4 is not being serviced. 1 = Interrupt request 4 is being serviced.
3	IS3	**Interrupt Request 3 In-Service** 0 = Interrupt request 3 is not being serviced. 1 = Interrupt request 3 is being serviced.
2	IS2	**Interrupt Request 2 In-Service** 0 = Interrupt request 2 is not being serviced. 1 = Interrupt request 2 is being serviced.
1	IS1	**Interrupt Request 1 In-Service** 0 = Interrupt request 1 is not being serviced. 1 = Interrupt request 1 is being serviced.
0	IS0	**Interrupt Request 0 In-Service** 0 = Interrupt request 0 is not being serviced. 1 = Interrupt request 0 is being serviced.

Programming Notes

This register (S1PICISR) is accessed by writing a value of 0Bh to Port 00A0h followed by a read back from Port 00A0h.

If the S2 bit in the MPICICW3 register is cleared (see page 12-34), then the Slave 1 controller is bypassed and interrupt requests latched are not serviced.

Slave 1 PIC Initialization Control Word 1 (S1PICICW1)

Direct-Mapped I/O Address 00A0h

	7	6	5	4	3	2	1	0
Bit	Reserved			SLCT_ ICW1	LTIM	ADI	SNGL	IC4
Reset	x	x	x	x	x	1	0	x
R/W	RSV!			W	W	W	W	W

Register Description

This register is the first initialization register of the Slave 1 controller.

Bit Definitions

Bit	Name	Function
7–5	Reserved	**Reserved** This bit field should be written to 0 for normal system operation. This I/O address changes functions when read. See the programming notes for this register (S1PICICW1) on page 12-52.
4	SLCT_ICW1	**Initialization Control Word 1 Select** Software must set this bit to 1 when writing this address (Port 00A0h) to access this register (S1PICICW1). 0 = The write does not access this register (S1PICICW1). Instead, either the S1PICOCW2 register (see page 12-53) or the S1PICOCW3 register (see page 12-55) is written, depending on the state of bit 3. 1 = The write accesses this register (S1PICICW1). Subsequent writes to Port 00A1h access additional initialization control words. See the programming notes for this register (S1PICICW1) on page 12-52.
3	LTIM	**Level-Triggered Interrupt Mode** This bit is the global interrupt mode selection for the Slave 1 PIC. 0 = Edge-sensitive interrupt request detection 1 = Level-sensitive interrupt request detection If the S1_GINT_MODE bit in the PICICR register is set (see page 12-5), the LTIM bit determines the interrupt mode for the Slave 1 PIC channels. If the S1_GINT_MODE bit is cleared, the Slave 1 LTIM bit has no meaning, and the Slave 1 PIC channel modes can be programmed individually via the SL1PICMODE register (see page 12-8).
2	ADI	**Address Interval** 0 = Interrupt vectors are separated by eight locations. 1 = Interrupt vectors are separated by four locations. In the ÉlanSC520 microcontroller design, this PC/AT-compatible bit is internally fixed to 1.

Bit	Name	Function
1	SNGL	**Single PIC** 0 = Cascade mode; S1PICICW3 is expected. 1 = Single PIC in the system; S1PICICW3 is not expected (not valid in the ÉlanSC520 microcontroller). In the ÉlanSC520 microcontroller design, this bit is internally fixed to 0. Because this bit is internally fixed to 0, software must always write the S1PICICW3 register after writing S1PICICW2. See the programming notes on this page for details.
0	IC4	**Initialization Control Word 4** Software uses this bit to indicate whether it intends to explicitly program the S1PICICW4 register (see page 12-59) after writing to the S1PICICW3 register (see page 12-58). See the programming notes on this page for details. 0 = The S1PICICW4 register is initialized internally when this register (S1PICICW1) is written. The PIC does not expect software to write to the S1PICICW4 register. 1 = The S1PICICW4 register is not initialized by the write to this register (S1PICICW1). Software is expected to initialize the S1PICICW4 register after writing to the S1PICICW3 register.

Programming Notes

The PIC's initialization control word (S1PICICWx) registers 1–4 must be programmed in sequence. Writing to Port 00A0h with bit 4 = 1 causes the S1PICICW1 register to be written and also resets the PIC's internal state machine and the internal S1PICICWx register pointer. Then, S1PICICWx registers 2–4 can be programmed by sequential writes to Port 00A1h. Each time Port 00A1h is written to (following the write to S1PICICW1), the internal register pointer points to the next S1PICICWx register. S1PICICW1 and S1PICICW2 must always be programmed. Also, the S1PICICW3 register must always be programmed in this design because the SNGL bit in S1PICICW1 is internally fixed to 0. The S1PICICW4 register is skipped if the IC4 bit in S1PICICW1 is 0.

If the S2 bit in the MPICICW3 register is cleared (see page 12-34), then the Slave 1 controller is bypassed and programming this register does not affect other registers.

I/O Port 00A0h provides access to different Slave 1 PIC registers based on the data that is written. Table 12-9 provides a summary of bit patterns to write for access to each register.

Table 12-9 Slave 1 PIC I/O Port 00A0h Access Summary

Bits									
7	6	5	4	3	2	1	0	Port 00A0h Register Written	Next Port 00A0h Read Returns
x	x	x	0	0	x	x	x	S1PICOCW2 (page 12-53)	—
0	x	x	0	1	x	0	x	S1PICOCW3 (page 12-55)	—
0	0	0	0	1	0	1	0	S1PICOCW3	S1PICIR (page 12-49)
0	0	0	0	1	0	1	1	S1PICOCW3	S1PICISR (page 12-50)
0	0	0	1	x	x	x	x	S1PICICW1 (page 12-51)	—

Slave 1 PIC Operation Control Word 2 (S1PICOCW2)

Direct-Mapped I/O Address 00A0h

	7	6	5	4	3	2	1	0
Bit	R_SL_EOI[1–0]			SLCT_ ICW1	IS_OCW3	LS[2–0]		
Reset	x	x	x	x	x	x	x	x
R/W	W			W	W	W		

Register Description

This register provides control for various interrupt priority and end-of-interrupt (EOI) modes. It also controls write access for this register (S1PICOCW2) and for the S1PICOCW3 and S1PICICW1 registers (see page 12-55 and page 12-51).

Bit Definitions

Bit	Name	Function
7–5	R_SL_EOI[1–0]	**Interrupt Request EOI and Priority Rotation Controls** 000 = Rotate in auto EOI mode (clear) 001 = Nonspecific EOI 010 = No operation 011 = Specific EOI 100 = Rotate in auto EOI mode (set) 101 = Rotate on nonspecific EOI command 110 = Set priority command 111 = Rotate on specific EOI command
4	SLCT_ICW1	**Initialization Control Word 1 Select** Software must clear this bit to 0 when writing this address (Port 00A0h) to access either this register (S1PICOCW2) or the S1PICOCW3 register. 0 = The write accesses either this register (S1PICOCW2) or the S1PICOCW3 register (see page 12-55), depending on the state of bit 3. 1 = The write accesses the S1PICICW1 register (see page 12-51).
3	IS_OCW3	**Access is OCW3** Software must clear this bit (IS_OCW3) and clear SLCT_ICW1 when writing this address (Port 00A0h) to access this register (S1PICOCW2). 0 = The write accesses this register (S1PICOCW2) if the SLCT_ICW1 bit is cleared. 1 = The write accesses the S1PICOCW3 register (see page 12-55) if the SLCT_ICW1 bit is cleared.

Bit	Name	Function
2–0	LS[2–0]	**Specific EOI Level Select** Interrupt level that is acted upon when the SL bit = 1 (see bits 7–5] below: 000 = IR0 001 = IR1 010 = IR2 011 = IR3 100 = IR4 101 = IR5 110 = IR6 111 = IR7

Programming Notes

If the S2 bit in the MPICICW3 register is cleared (see page 12-34), then the Slave 1 controller is bypassed and programming this register does not affect other registers.

I/O Port 00A0h provides access to different Slave 1 PIC registers based on the data that is written. Table 12-10 provides a summary of bit patterns to write for access to each register.

Table 12-10 Slave 1 PIC I/O Port 00A0h Access Summary (Same as Table 12-9)

Bits									
7	6	5	4	3	2	1	0	Port 00A0h Register Written	Next Port 00A0h Read Returns
x	x	x	0	0	x	x	x	S1PICOCW2 (page 12-53)	—
0	x	x	0	1	x	0	x	S1PICOCW3 (page 12-55)	—
0	0	0	0	1	0	1	0	S1PICOCW3	S1PICIR (page 12-49)
0	0	0	0	1	0	1	1	S1PICOCW3	S1PICISR (page 12-50)
0	0	0	1	x	x	x	x	S1PICICW1 (page 12-51)	—

Slave 1 PIC Operation Control Word 3 (S1PICOCW3)

Direct-Mapped I/O Address 00A0h

	7	6	5	4	3	2	1	0
Bit	Reserved	ESMM_SMM[1–0]		SLCT_ ICW1	IS_OCW3	P	RR_RIS[1–0]	
Reset	x	1	x	x	x	x	x	x
R/W	RSV!	W		W	W	W	W	

Register Description

This register controls the PIC's mask and poll modes. It also controls read access for the S1PICIR and S1PICISR registers (see page 12-49 and page 12-50), and write access for this register (S1PICOCW3) and for the S1PICOCW2 and S1PICICW1 registers (see page 12-53 and page 12-51).

Bit Definitions

Bit	Name	Function
7	Reserved	**Reserved** This bit field should be written to 0 for normal system operation. This I/O address changes functions when read. See the programming notes for this register (S1PICICW1) on page 12-56.
6–5	ESMM_SMM [1–0]	**Special Mask Mode** 00 = No operation 01 = No operation 10 = Reset special mask 11 =Set special mask
4	SLCT_ICW1	**Initialization Control Word 1 Select** Software must clear this bit to 0 when writing this address (Port 00A0h) to access either this register (S1PICOCW3) or the S1PICOCW2 register. 0 = The write accesses either this register (S1PICOCW3) or the S1PICOCW2 register (see page 12-53), depending on the state of bit 3. 1 = The write accesses the S1PICICW1 register (see page 12-51).
3	IS_OCW3	**Access is OCW3** Software must set this bit (IS_OCW3) and clear SLCT_ICW1 when writing this address (Port 00A0h) to access this register (S1PICOCW3). 0 = The write accesses the S1PICOCW2 register (see page 12-53) if the SLCT_ICW1 bit is cleared. 1 = The write accesses this register (S1PICOCW3) if the SLCT_ICW1 bit is cleared.
2	P	**PIC Poll Command** A system designer can choose to use the PIC in a non-interrupting mode. In this case, the interrupt controller can be polled for the status of pending interrupts. To support this PC/AT-incompatible mode of operation, the PIC supports a special poll command that is invoked by setting this bit. 0 = Not poll command 1 = Poll command
1–0	RR_RIS[1–0]	**Status Register Select** 00 = No change from last state 01 = No change from last state 10 = Next Port 00A0h read returns the S1PICIR register's contents (see page 12-49). 11 = Next Port 00A0h read returns the S1PICISR register's contents (see page 12-50).

Programming Notes

If the S2 bit in the MPICICW3 register is cleared (see page 12-34), then the Slave 1 controller is bypassed and programming this register does not affect other registers.

I/O Port 00A0h provides access to different Slave 1 PIC registers based on the data that is written. Table 12-11 provides a summary of bit patterns to write for access to each register.

Table 12-11 Slave 1 PIC I/O Port 00A0h Access Summary (Same as Table 12-9)

Bits									
7	6	5	4	3	2	1	0	Port 00A0h Register Written	Next Port 00A0h Read Returns
x	x	x	0	0	x	x	x	S1PICOCW2 (page 12-53)	—
0	x	x	0	1	x	0	x	S1PICOCW3 (page 12-55)	—
0	0	0	0	1	0	1	0	S1PICOCW3	S1PICIR (page 12-49)
0	0	0	0	1	0	1	1	S1PICOCW3	S1PICISR (page 12-50)
0	0	0	1	x	x	x	x	S1PICICW1 (page 12-51)	—

Slave 1 PIC Initialization Control Word 2 (S1PICICW2)

Direct-Mapped I/O Address 00A1h

	7	6	5	4	3	2	1	0
Bit	T7–T3					A10–A8		
Reset	x	x	x	x	x	x	x	x
R/W	W					W		

Register Description

This register is the second initialization register of the Slave 1 controller.

Bit Definitions

Bit	Name	Function
7–3	T7–T3	**Bits 7–3 of Base Interrupt Vector Number for this PIC** The PIC concatenates the T7–T3 bit field value to the 3-bit PIC interrupt request level (in the bit 2–0 position) to form the interrupt vector. For example, in a PC/AT-compatible system, bits T7–T3 for the Master PIC are programmed to 00001b so the Master PIC IR0 channel generates an interrupt 08h vector (PC/AT IRQ0); and bits T7–T3 for the Slave 1 PIC are programmed to 01110b so the Slave 1 PIC IR0 channel generates an interrupt 70h (PC/AT IRQ8).
2–0	A10–A8	**A10–A8 of Interrupt Vector** This bit field should always be written to 0 for normal operation. (It is always 0 in a PC/AT-compatible system.)

Programming Notes

If the S2 bit in the MPICICW3 register is cleared (see page 12-34), then the Slave 1 controller is bypassed and programming this register does not affect other registers.

The PIC's initialization control word (S1PICICWx) registers 1–4 must be programmed in sequence. Writing to Port 00A0h with bit 4 = 1 causes the S1PICICW1 register to be written and also resets the PIC's internal state machine and the internal S1PICICWx register pointer. Then, S1PICICWx registers 2–4 can be programmed by sequential writes to Port 00A1h. Each time Port 00A1h is written to (following the write to S1PICICW1), the internal register pointer points to the next S1PICICWx register. S1PICICW1 and S1PICICW2 must always be programmed. Also, the S1PICICW3 register must always be programmed in this design because the SNGL bit in S1PICICW1 is internally fixed to 0. The S1PICICW4 register is skipped if the IC4 bit in S1PICICW1 is 0.

Slave 1 PIC Initialization Control Word 3 (S1PICICW3)

Direct-Mapped I/O Address 00A1h

	7	6	5	4	3	2	1	0
Bit	Reserved					ID2–ID0		
Reset	x	x	x	x	x	0	1	0
R/W	RSV!					W		

Register Description

This register is the third initialization register of the Slave 1 controller.

Bit Definitions

Bit	Name	Function
7–3	Reserved	**Reserved** This bit field should be written to 0 for normal system operation. This bit field is write-only.
2–0	ID2–ID0	**Slave 1 PIC ID 2–0** These bits contain the binary Slave 1PIC ID (000b–111b) that the PIC responds to on the cascade bus. In the ÉlanSC520 microcontroller design, these bits are internally fixed to 010b.

Programming Notes

The PIC's initialization control word (S1PICICWx) registers 1–4 must be programmed in sequence. Writing to Port 00A0h with bit 4 = 1 causes the S1PICICW1 register to be written and also resets the PIC's internal state machine and the internal S1PICICWx register pointer. Then, S1PICICWx registers 2–4 can be programmed by sequential writes to Port 00A1h. Each time Port 00A1h is written to (following the write to S1PICICW1), the internal register pointer points to the next S1PICICWx register. S1PICICW1 and S1PICICW2 must always be programmed. Also, the S1PICICW3 register must always be programmed in this design because the SNGL bit in S1PICICW1 is internally fixed to 0. The S1PICICW4 register is skipped if the IC4 bit in S1PICICW1 is 0.

If the S2 bit in the MPICICW3 register is cleared (see page 12-34), a write to this register (S1PICICW3) is always expected in the ÉlanSC520 microcontroller because the SNGL bit is fixed to 0 in the S1PICICW1 register (page 12-52).

If the S2 bit in the MPICICW3 register is cleared, then the Slave 1 controller is bypassed and the value of this register (S1PICICW3) has no effect.

Slave 1 PIC Initialization Control Word 4 (S1PICICW4)

Direct-Mapped I/O Address 00A1h

	7	6	5	4	3	2	1	0
Bit	Reserved			SFNM	BUF_M/S[1–0]		AEOI	PM
Reset	x	x	x	x	0	0	0	1
R/W	RSV!			W	W		W	W

Register Description

This register is the fourth initialization register of the Slave 1 controller.

Bit Definitions

Bit	Name	Function
7–5	Reserved	**Reserved** This bit field should be written to 0 for normal system operation. This bit field is write-only.
4	SFNM	**Special Fully Nested Mode Enable** 0 = Normal nested mode 1 = Special fully nested mode For PC/AT compatibility, this bit should be programmed as 0.
3–2	BUF_M/S[1–0]	**Buffered Mode and Master/Slave Select** 00 = Non buffered mode 01 = Non buffered mode 10 = Buffered mode/Slave 11 = Buffered mode/Master In the ÉlanSC520 microcontroller, these PC/AT-compatible bits are internally fixed to 00b.
1	AEOI	**Automatic EOI Mode** 0 = Normal EOI: the interrupt handler must send an End of Interrupt command to the PIC(s). 1 = Auto EOI: the EOI is automatically performed after the second interrupt acknowledge signal from the CPU. In the ÉlanSC520 microcontroller, this bit is internally fixed to 0. The Slave 1 PIC and Slave 2 PIC do not support automatic EOI mode.
0	PM	**Microprocessor Mode** 0 = 8080/8085 mode 1 = 8086 mode In the ÉlanSC520 microcontroller design, this PC/AT-compatible bit is internally fixed to 1.

Programming Notes

The PIC's initialization control word (S1PICICWx) registers 1–4 must be programmed in sequence. Writing to Port 00A0h with bit 4 = 1 causes the S1PICICW1 register to be written and also resets the PIC's internal state machine and the internal S1PICICWx register pointer. Then, S1PICICWx registers 2–4 can be programmed by sequential writes to Port 00A1h. Each time Port 00A1h is written (following the write to S1PICICW1), the internal register pointer points to the next S1PICICWx register. S1PICICW1 and S1PICICW2 must always be programmed. Also, the S1PICICW3 register must always be programmed in this design because the SNGL bit in S1PICICW1 is internally fixed to 0. The S1PICICW4 register is skipped if the IC4 bit in S1PICICW1 is 0.

Software is expected to initialize this register (S1PICICW4) if the IC4 bit is set in the S1PICICW1 register (see page 12-52). If the IC4 bit is cleared, the ÉlanSC520 microcontroller uses 01h for the value of this register (S1PICICW4).

However, if the S2 bit is cleared in the MPICICW3 register (see page 12-34), then the Slave 1 controller is bypassed and programming this register (S1PICICW4) does not affect other registers.

Slave 1 PIC Interrupt Mask (S1PICINTMSK)

Direct-Mapped I/O Address 00A1h

	7	6	5	4	3	2	1	0
Bit	IM7	IM6	IM5	IM4	IM3	IM2	IM1	IM0
Reset	x	x	x	x	x	x	x	x
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Register Description

This register provides masking of individual interrupt requests for the Slave 1 controller. This register is also known as Operation Control Word 1 in other PC/AT-compatible designs.

Bit Definitions

Bit	Name	Function
7	IM7	**IR7 Mask** 0 = Unmask IR7 1 = Mask IR7
6	IM6	**IR6 Mask** 0 = Unmask IR6 1 = Mask IR6
5	IM5	**IR5 Mask** 0 = Unmask IR5 1 = Mask IR5
4	IM4	**IR4 Mask** 0 = Unmask IR4 1 = Mask IR4
3	IM3	**IR3 Mask** 0 = Unmask IR3 1 = Mask IR3
2	IM2	**IR2 Mask** 0 = Unmask IR2 1 = Mask IR2
1	IM1	**IR1 Mask** 0 = Unmask IR1 1 = Mask IR1
0	IM0	**IR0 Mask** 0 = Unmask IR0 1 = Mask IR0

Programming Notes

If the S2 bit in the MPICICW3 register is cleared (see page 12-34), then the Slave 1 controller is bypassed and the value of this register (S1PICINTMSK) has no effect.

This register (S1PICINTMSK) cannot be accessed during a Slave 1 PIC initialization control sequence, which is initiated by setting the SLCT_ICW1 bit in the S1PICICW1 register (see page 12-51).

When the S1PICICWx register initialization sequence is not in effect, any read or write of Port 00A1h accesses the S1PICINTMSK register.

Floating Point Error Interrupt Clear (FPUERRCLR)

Direct-Mapped
I/O Address 00F0h

	7	6	5	4	3	2	1	0
Bit	FPUERR_RST							
Reset	0	0	0	0	0	0	0	0
R/W	W							

Register Description

This register is used to clear the Am5$_x$86 CPU floating point unit (FPU) error interrupt request for DOS-compatible error handling.

Bit Definitions

Bit	Name	Function
7–0	FPUERR_RST	**Clear FPU Error Interrupt Request** Any write to this register address clears the interrupt request that is generated by an Am5$_x$86 CPU floating point unit error. (The CPU asserts the $\overline{ferr}$ signal internally.) At the conclusion of the write to this register, the CPU's ignne signal is asserted internally to allow non-control instructions to be executed. This function is provided for systems that require DOS-compatible FPU error handling.

Programming Notes

CHAPTER

13 PROGRAMMABLE INTERVAL TIMER REGISTERS

13.1 OVERVIEW

This chapter describes the programmable interval timer (PIT) registers of the ÉlanSC520 microcontroller.

There are three PIT channels. The PIT module is one of four ÉlanSC520 microcontroller timer modules. The other timer modules are described in the following chapters:

- Chapter 15, “Software Timer Registers”
- Chapter 14, “General-Purpose Timer Registers”
- Chapter 16, “Watchdog Timer Registers”

The PIT register set consists of 10 direct-mapped I/O registers used to configure, control, and read status for the three PIT channels. It also includes the System Control Port B (SYSCTLB), which provides certain PIT-related functions. See the *Élan™SC520 Microcontroller User’s Manual*, order #22004, for details about the PIT channels.

Table 13-1 lists the PIT registers in offset order, with the corresponding description’s page number.

13.2 REGISTERS

Table 13-1 Programmable Interval Timer Direct-Mapped Registers

Register Name	Mnemonic	I/O Address	Page Number
PIT Channel 0 Count	PIT0CNT	0040h	page 13-2
PIT Channel 1 Count	PIT1CNT	0041h	page 13-3
PIT Channel 2 Count	PIT2CNT	0042h	page 13-4
PIT 0 Status	PIT0STA	0040h	page 13-5
PIT 1 Status	PIT1STA	0041h	page 13-5
PIT 2 Status	PIT2STA	0042h	page 13-5
PIT Mode Control	PITMODECTL	0043h	page 13-7
PIT Counter Latch Command	PITCNTLAT	0043h	page 13-10
PIT Read-Back Command	PITRDBACK	0043h	page 13-11
System Control Port B	SYSCTLB	0061h	page 13-13

PIT Channel 0 Count (PIT0CNT)

Direct-Mapped I/O Address 0040h

	7	6	5	4	3	2	1	0
Bit	CH0_CNT[15–0]							
Reset	0	0	0	0	0	0	0	0
R/W	R/W							

Register Description

This register stores the current count values for PIT Channel 0.

Bit Definitions

Bit	Name	Function
7–0	CH0_CNT[15–0]	**16-bit Counter for Programmable Interval Timer, Channel 0** This bit field provides read or write access to either the counter high byte only, low byte only, or low byte followed by high byte, as defined by the CTR_RW_LATCH bit field of the PITMODECTL register (see page 13-7). Either a latched or unlatched (free-running) count value can be read from this bit field (CH0_CNT). A latched count can be read immediately after a counter latch command is issued via the PITCNTLAT register (see page 13-10). After the one or two bytes of latched count data are read, subsequent reads return the unlatched count. A latched count can also be read immediately following a read-back command, issued via the PITRDBACK register, in which the LCNT bit is 0 and the CNT0 bit is 1 (see page 13-11). In this case also, after the one or two bytes of latched count are read, subsequent reads return the unlatched count.

Programming Notes

If a read-back command is issued by writing the PITRDBACK register with the LSTAT bit clear and the CNT0 bit set (see page 13-11), the subsequent read to this address (Port 0040h) returns the PIT0STA register status byte (see page 13-5). If a read-back command is issued in which PITRDBACK register bits LSTAT and LCNT are clear and bit CNT0 is set, the first subsequent read from this address (Port 0040h) returns the status byte, and the following one or two reads return latched count data as defined by the CTR_RW_LATCH bit field of the PITMODECTL register (see page 13-7).

This counter can be configured for either binary-coded decimal (BCD) or 16-bit binary operation via the PITMODECTL register's BCD bit (see page 13-8). The counter range is 0–FFFFh in binary mode or 0–9999d in BCD. However, the maximum value in either mode can be achieved by clearing this register (PIT0CNT) to 0.

All three PIT counters run at the same rate. If the CLK_PIN_DIR bit is 0 in the CLKSEL register (see page 20-9), the CLKTIMER input signal drives the PIT counters. Otherwise the PIT counters run at 1.1892 MHz.

PIT Channel 1 Count (PIT1CNT)

Direct-Mapped I/O Address 0041h

	7	6	5	4	3	2	1	0
Bit	CH1_CNT[15–0]							
Reset	0	0	0	0	0	0	0	0
R/W	R/W							

Register Description

This register stores the current count values for PIT Channel 1.

Bit Definitions

Bit	Name	Function
7–0	CH1_CNT[15–0]	**16-bit Counter for Programmable Interval Timer, Channel 1** This bit field provides read or write access to either the counter high byte only, low byte only, or low byte followed by high byte, as defined by the CTR_RW_LATCH bit field of the PITMODECTL register (see page 13-7). Either a latched or unlatched (free-running) count value can be read from this bit field (CH1_CNT). A latched count can be read immediately after a counter latch command is issued via the PITCNTLAT register (see page 13-10). After the one or two bytes of latched count data are read, subsequent reads return the unlatched count. A latched count can also be read immediately following a read-back command, issued via the PITRDBACK register, in which the LCNT bit is 0 and the CNT1 bit is 1 (see page 13-11). In this case also, after the one or two bytes of latched count are read, subsequent reads return the unlatched count.

Programming Notes

If a read-back command is issued by writing the PITRDBACK register with the LSTAT bit clear and the CNT1 bit set (see page 13-11), the subsequent read to this address (Port 0041h) returns the PIT1STA register status byte (see page 13-5). If a read-back command is issued in which PITRDBACK register bits LSTAT and LCNT are clear and bit CNT1 is set, the first subsequent read from this address (Port 0041h) returns the status byte, and the following one or two reads return latched count data as defined by the CTR_RW_LATCH bit field of the PITMODECTL register (see page 13-7).

This counter can be configured for either binary-coded decimal (BCD) or 16-bit binary operation via the PITMODECTL register's BCD bit (see page 13-8). The counter range is 0–FFFFh in binary mode or 0–9999d in BCD. However, the maximum value in either mode can be achieved by clearing this register (PIT1CNT) to 0.

All three PIT counters run at the same rate. If the CLK_PIN_DIR bit is 0 in the CLKSEL register (see page 20-9), the CLKTIMER input signal drives the PIT counters. Otherwise the PIT counters run at 1.1892 MHz.

PIT Channel 2 Count (PIT2CNT)

Direct-Mapped I/O Address 0042h

	7	6	5	4	3	2	1	0
Bit	CH2_CNT[15–0]							
Reset	0	0	0	0	0	0	0	0
R/W	R/W							

Register Description

This register stores the current count values for PIT Channel 2.

Bit Definitions

Bit	Name	Function
7–0	CH2_CNT[15–0]	**16-bit Counter for Programmable Interval Timer, Channel 2** This bit field provides read or write access to either the counter high byte only, low byte only, or low byte followed by high byte, as defined by the CTR_RW_LATCH bit field of the PITMODECTL register (see page 13-7). Either a latched or unlatched (free-running) count value can be read from this bit field (CH2_CNT). A latched count can be read immediately after a counter latch command is issued via the PITCNTLAT register (see page 13-10). After the one or two bytes of latched count data are read, subsequent reads return the unlatched count. A latched count can also be read immediately following a read-back command, issued via the PITRDBACK register, in which the LCNT bit is 0 and the CNT2 bit is 1 (see page 13-11). In this case also, after the one or two bytes of latched count are read, subsequent reads return the unlatched count.

Programming Notes

If a read-back command is issued by writing the PITRDBACK register with the LSTAT bit clear and the CNT2 bit set (see page 13-11), the subsequent read to this address (Port 0042h) returns the PIT2STA register status byte (see page 13-5). If a read-back command is issued in which PITRDBACK register bits LSTAT and LCNT are clear and bit CNT2 is set, the first subsequent read from this address (Port 0042h) returns the status byte, and the following one or two reads return latched count data as defined by the CTR_RW_LATCH bit field of the PITMODECTL register (see page 13-7).

This counter can be configured for either binary-coded decimal (BCD) or 16-bit binary operation via the PITMODECTL register's BCD bit (see page 13-8). The counter range is 0–FFFFh in binary mode or 0–9999d in BCD. However, the maximum value in either mode can be achieved by clearing this register (PIT2CNT) to 0.

All three PIT counters run at the same rate. If the CLK_PIN_DIR bit is 0 in the CLKSEL register (see page 20-9), the CLKTIMER input signal drives the PIT counters. Otherwise the PIT counters run at 1.1892 MHz.

Direct-Mapped

PIT 0 Status (PIT0STA) — I/O Address 0040h

PIT 1 Status (PIT1STA) — I/O Address 0041h

PIT 2 Status (PIT2STA) — I/O Address 0042h

	7	6	5	4	3	2	1	0
Bit	OUTPUT	NULL_CNT	RW[1–0]		CTR_MODE_STA[2–0]			BCD
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R		R			R

Register Description

These registers contain the programmed mode and the null count value for each timer channel.

Bit Definitions

Bit	Name	Function
7	OUTPUT	**Output State** Output signal state for the timer channel. Each timer channel has an output that is driven High or Low based on the current mode. See the CTR_MODE_STA bit description on page 13-6 for a list of modes. 0 = Output is Low. 1 = Output is High. The three PIT outputs can be mapped individually to the microcontroller's internal programmable interrupt controller inputs. In addition, the PIT Channel 2 output can be enabled to drive the PITOUT2 pin. See the PIT_OUT2_ENB bit description on page 13-13.
6	NULL_CNT	**Null Count** When programming a new count value into one of the timers, the new value does not take effect until it has actually been transferred to the counting element, which can take some time. When attempting to read back a count value, software can test this bit to determine whether the read-back value is valid or not. 0 = Counter is available for reading. 1 = Null count; read-back of the counter is invalid.
5–4	RW[1–0]	**Counter Read/Write Operation Control or Counter Latch Command** Reflects the last bit setting that was programmed into the corresponding bits (5–4) of the PITMODECTL or PITCNTLAT register for this PIT channel (see page 13-7 and page 13-10). 00 = Counter Latch command (PITCNTLAT register). 01 = Read/write counter bits 7–0 only (PITMODECTL register). 10 = Read/write counter bits 15–8 only (PITMODECTL register). 11 = Read/write counter bits 7–0 followed immediately by bits 15–8 (PITMODECTL register).

Bit	Name	Function
3–1	CTR_MODE_STA [2–0]	**Counter Mode Status** Reflects the last counter mode setting for counters 0, 1, or 2 that was programmed into this channel via the CTR_MODE bit field of the PITMODECTL register (see page 13-8). Gate = High unless noted 000 = Mode 0: Interrupt on terminal count 001 = Mode 1: Hardware retriggerable one-shot 010 = Mode 2: Rate generator 011 = Mode 3: Square wave generator 100 = Mode 4: Software retriggerable strobe 101 = Mode 5: Hardware retriggerable strobe 110 = Alias for mode 2 111 = Alias for mode 3 See Table 13-2 on page 13-9 for more detail on counter modes.
0	BCD	**Binary Coded Decimal Select Status** Reflects the last BCD setting for counters 0, 1, or 2 that was programmed into this channel via the BCD bit field of the PITMODECTL register (see page 13-8). 0 = 16-bit binary counter with a range of 0–FFFFh 1 = BCD counter with a range of 0–9999d

Programming Notes

The PITxSTA register is available only after a read-back command is issued by writing the PITRDBACK register with the LSTAT bit clear and the appropriate CNTx bit set (see page 13-11).

PIT Mode Control (PITMODECTL)

Direct-Mapped I/O Address 0043h

	7	6	5	4	3	2	1	0
Bit	CTR_SEL[1–0]		CTR_RW_LATCH[1–0]		CTR_MODE[2–0]			BCD
Reset	0	0	0	0	0	0	0	0
R/W	W		W		W			W

Register Description

This register stores the control word used to define the operation of the channels, including mode.

Bit Definitions

Bit	Name	Function
7–6	CTR_SEL[1–0]	**PIT Counter Select** When this address (Port 0043h) is written with bits 7–6 ≠ 11b and bits 5–4 ≠ 00b, the PITMODECTL register is addressed, and this CTR_SEL bit field specifies which of the three PIT channels is affected by the settings written to bits 5–0. 00 = Select PIT counter 0. 01 = Select PIT counter 1. 10 = Select PIT counter 2. 11 = The PITRDBACK register is addressed (see page 13-11). When this address (Port 0043h) is written with bits 7–6 = 11b, this address is redefined (for the duration of the current I/O write) as the PITRDBACK register (see page 13-11). When this address (Port 0043h) is written with bits 7–6 ≠ 11b, and bits 5–4 = 00b, this address is redefined (for the duration of the current I/O write) as the PITCNTLAT register (see page 13-10).
5–4	CTR_RW_LATCH [1–0]	**Counter Read/Write Operation Control** When this address (Port 0043h) is written with bits 7–6 ≠ 11b and bits 5–4 ≠ 00b, the PITMODECTL register is addressed, and these CTR_RW_LATCH bits define what can be written to the PITxCNT register, where x is selected by bits 7–6 of this PITMODECTL register. 00 = The PITCNTLAT register is addressed (see page 13-10). 01 = Read/write counter bits 7–0 only. 10 = Read/write counter bits 15–8 only. 11 = Read/write counter bits 7–0 followed immediately by bits 15–8. See the PITxCNT register descriptions starting on page 13-2. Note that if a given PIT counter (as specified by the CTR_SEL bit field) is programmed for only eight bit accesses (i.e., this CTR_RW_LATCH bit field is 01b or 10b), a subsequent write to the count register automatically clears the eight bits of the associated internal 16-bit counting element that were not explicitly written to.

Bit	Name	Function
3–1	CTR_MODE[2–0]	**Counter Mode** When this address (Port 0043h) is written with bits 7–6 ≠ 11b and bits 5–4 ≠ 00b, the PITMODECTL register is addressed, and these bits control the counter operation: 000 = Mode 0: Interrupt on terminal count 001 = Mode 1: Hardware retriggerable one-shot (for PIT Channel 2) 010 = Mode 2: Rate generator 011 = Mode 3: Square wave generator 100 = Mode 4: Software retriggerable strobe 101 = Mode 5: Hardware retriggerable strobe (for PIT Channel 2) 110 = Alias for mode 2 111 = Alias for mode 3 See Table 13-2 on page 13-9 for more detail on this bit field.
0	BCD	**Binary Coded Decimal Select** When this address (Port 0043h) is written with bits 7–6 ≠ 11b and bits 5–4 ≠ 00b, the PITMODECTL register is addressed, and this bit controls whether the counter indicated by bits 7–6 of this register counts in binary with a range of 0–FFFFh or in binary coded decimal (BCD) with a range of 0–9999d. 0 = 16-bit binary counter 1 = BCD counter

Programming Notes

Writing to this register (PITMODECTL) to change the mode for a particular counter resets the control logic and the associated output. Reads of this register return an undefined value.

When a counter's current value is required, software should obtain it by issuing a counter latch or read-back command via the PITCNTLAT or PITRDBACK register (see page 13-10 or page 13-11). A counter latch or read-back command does not stop a counter from running, but rather takes a snapshot of the current value. Once the count has been latched, further latch commands are ignored until all latched count data is read back from the associated count register.

A read-back command is a higher priority command than the counter latch command. The counter latch command is a subset of the read-back command because only one channel can have its counter latched per counter latch command.

The programmable interval timer does not provide any way to read back the original count programmed into any of the three count registers.

Table 13-2 PIT Counter Mode Settings

CTR_MODE Bit Field Setting	Mode Name	Description
000b = Mode 0	Interrupt on terminal count	When the counter is programmed, the counter output transitions to 0. When the counter reaches 0, the counter output transitions to 1 and remains there until another count value is written. Before a counter is programmed to generate interrupts, the corresponding PITxMAP register (see page 12-21) must be configured to route the interrupt to the appropriate interrupt request level and priority.
001b = Mode 1	Hardware retriggerable one-shot[1]	Available for PIT Channel 2 only. When the Channel 2 counter is programmed, the Channel 2 output transitions to 1. After a Low-to-High transition of the internal gate 2 signal (controlled by the PITGATE2 signal or the PIT_GATE2 bit of the SYSCTLB register, see page 13-13), the counter output transitions to 0 until the count reaches 0, then the counter output transitions to 1 until the next Low-to-High transition on the gate input.
010b = Mode 2	Rate generator	Each time the count transitions to 1, the counter output transitions to 0 and remains there for one cycle of the input clock, and then the counter output transitions to 1. The count is automatically reloaded, and the process repeats. By default, PC/AT-compatible systems program Channel 0 for this mode.
011b = Mode 3	Square wave generator	When the count is loaded, the counter output transitions to 1. When 1/2 of the count has expired, the counter output transitions to 0. When count transitions to 0, the counter output transitions to 1, and count is automatically reloaded. By default, PC/AT-compatible systems program Channels 1 and 2 to use this mode to drive DRAM refresh and the speaker, respectively.
100b = Mode 4	Software retriggerable strobe	When the count is loaded, the counter output transitions to 1. When count transitions to 0, the counter output transitions to 0 for one clock period and then transitions to 1.
101b = Mode 5	Hardware retriggerable strobe[1]	Available for PIT Channel 2 only. The counter output behaves just as in mode 4 except that the triggering is done on a Low-to-High transition of the internal gate 2 signal (controlled by the PITGATE2 signal or the PIT_GATE2 bit of the SYSCTLB register, see page 13-13). A trigger seen during the count reloads the count to the initial value and then counting continues.
110b	—	Same as mode 2
111b	—	Same as mode 3

Notes:

1. Modes 1 and 5 require a rising edge on the gate input for each timer channel. Only PIT Channel 2 has a gate control (see the PIT_GATE2 bit in the SYSCTLB register, page 13-13) so only PIT Channel 2 is capable of running all modes. The gate controls for PIT channels 0 and 1 are fixed internally to 1, so they are capable of operation only in modes 0, 2, 3 and 4.

PIT Counter Latch Command (PITCNTLAT)

Direct-Mapped I/O Address 0043h

	7	6	5	4	3	2	1	0
Bit	CTR_SEL[1–0]		CTR_CMD[1–0]		Reserved			
Reset	0	0	0	0	0	0	0	0
R/W	W		W		RSV			

Register Description

This register allows the current count of the selected channel to be read.

Bit Definitions

Bit	Name	Function
7–6	CTR_SEL[1–0]	**PIT Counter Select** When this address (Port 0043h) is written with bits 7–6 ≠ 11b and bits 5–4 = 00b, the PITCNTLAT register is addressed, and this CTR_SEL bit field specifies which of the three counter elements to latch for read back from the associated count register: 00 = Select PIT counter 0. 01 = Select PIT counter 1. 10 = Select PIT counter 2. 11 = The PITRDBACK register is addressed (see page 13-11). When this address (Port 0043h) is written with bits 7–6 = 11b, this address is redefined (for the duration of the current I/O write) as the PITRDBACK register (see page 13-11). When this address (Port 0043h) is written with bits 7–6 ≠ 11b and bits 5–4 ≠ 00b, this address is redefined (for the duration of the current I/O write) as the PITMODECTL register (see page 13-7).
5–4	CTR_CMD[1–0]	**Counter Latch Command** When this address (Port 0043h) is written with bits 7–6 ≠ 11b and bits 5–4 = 00b, the PITCNTLAT register is addressed, and this CTR_CMD bit field invokes the counter latch command. 00 = The Counter Latch command is invoked. 01–11 = The PITMODECTL or PITRDBACK register is addressed (see page 13-7 and page 13-11). Latched counts are read back from the PITxCNT registers (see the descriptions starting on page 13-2) based on the current read/write mode selected for each counter via the CTR_RW_LATCH bit field of the PITMODECTL register (see page 13-7).
3–0	Reserved	**Reserved** This bit field should be written to 0 for normal system operation.

Programming Notes

Reads of this register (PITCNTLAT) return an undefined value.

When a counter's current value is required, software should obtain it by issuing a counter latch or read-back command via the PITCNTLAT or PITRDBACK register (see page 13-10 or page 13-11). A counter latch or read-back command does not stop a counter from running, but rather takes a snapshot of the current value. Once the count has been latched, further latch commands are ignored until all latched count data is read back from the associated count register.

A read-back command is a higher priority command than the counter latch command. The counter latch command is a subset of the read-back command because only one channel can have its counter latched per counter latch command.

The programmable interval timer does not provide any way to read back the original count programmed into any of the three count registers.

PIT Read-Back Command (PITRDBACK)

Direct-Mapped I/O Address 0043h

	7	6	5	4	3	2	1	0
Bit	CTR_SEL[1–0]		LCNT	LSTAT	CNT2	CNT1	CNT0	Reserved
Reset	0	0	0	0	0	0	0	0
R/W	W		W	W	W	W	W	RSV

Register Description

This register allows the status and current count of each channel to be read.

Bit Definitions

Bit	Name	Function
7–6	CTR_SEL[1–0]	**PIT Counter Select/Read-back Command** When this address (Port 0043h) is written with bits 7–6 = 11b, the PITRDBACK register is addressed, and this CTR_SEL bit field invokes the read-back command. 00–10 = The PITMODECTL or PITCNTLAT register is addressed (see page 13-7 and page 13-10). 11 = Read-back command: the values selected by bits 5–1 of this register are made available to be read back from the PITxCNT registers (see the descriptions starting on page 13-2) immediately following completion of the I/O write that invokes this read-back command. Latched counts are read back from the PITxCNT registers (see the descriptions starting on page 13-2) based on the current read/write mode selected for each counter via the CTR_RW_LATCH bit field of the PITMODECTL register (see page 13-7). The latched status bytes are returned in the PITxSTA register format (see page 13-5). If both LSTAT and LCNT = 0b, the status byte is made available at the respective PITxCNT register first. When this byte has been read, the latched count bytes are made available, bits 7–0 first, and then bits 15–8 (if the channel is set up to read back all 16 bits of count via the CTR_RW_LATCH bit field of the PITMODECTL). When this address (Port 0043h) is written with bits 7–6 ≠ 11b and bits 5–4 ≠ 00b, this address is redefined (for the duration of the current I/O write) as the PITMODECTL register (see page 13-7) When this address (Port 0043h) is written with bits 7–6 ≠ 11b, and bits 5–4 = 00b, this address is redefined (for the duration of the current I/O write) as the PITCNTLAT register (see page 13-10).
5	LCNT	**Latch Count (Low True)** 0 = Latch count data for the counters selected via bits CNT2–CNT0. 1 = Do not latch count data for the counters selected via bits CNT2–CNT0.
4	LSTAT	**Latch Status (Low True)** 0 = Latch the status byte for the counters selected via bits CNT2–CNT0. 1 = Do not latch the status byte for the counters selected via bits CNT2–CNT0.
3	CNT2	**Select Counter 2** 0 = Counter 2 is not selected for operations specified by bits LCNT and LSTAT. 1 = Counter 2 is selected for operations specified by bits LCNT and LSTAT.
2	CNT1	**Select Counter 1** 0 = Counter 1 is not selected for operations specified by bits LCNT and LSTAT. 1 = Counter 1 is selected for operations specified by bits LCNT and LSTAT.
1	CNT0	**Select Counter 0** 0 = Counter 0 is not selected for operations specified by bits LCNT and LSTAT. 1 = Counter 0 is selected for operations specified by bits LCNT and LSTAT.

Bit	Name	Function
0	Reserved	**Reserved** This bit field should be written to 0 for normal system operation.

Programming Notes

Reads of this register (PITRDBACK) return an undefined value.

When a counter's current value is required, software should obtain it by issuing a counter latch or read-back command via the PITCNTLAT or PITRDBACK register (see page 13-10 or page 13-11). A counter latch or read-back command does not stop a counter from running, but rather takes a snapshot of the current value. Once the count has been latched, further latch commands are ignored until all latched count data is read back from the associated count register.

A read-back command is a higher priority command than the counter latch command. The counter latch command is a subset of the read-back command because only one channel can have its counter latched per counter latch command.

The programmable interval timer does not provide any way to read back the original count programmed into any of the three count registers.

System Control Port B (SYSCTLB)

Direct-Mapped I/O Address 0061h

	7	6	5	4	3	2	1	0
Bit	PERR	IOCHCK	PITOUT2_STA	RFD	Reserved		PIT_OUT2_ENB	PIT_GATE2
Reset	0	0	1	x	0		0	0
R/W	R	R	R	R	RSV!		R/W	R/W

Register Description

This register contains the PC/AT-compatible System Control Port B register bits that pertain to the programmable interval timer (PIT).

Bit Definitions

Bit	Name	Function
7	PERR	**PC/AT Parity Error Indicator** This PC/AT-compatible bit is not supported (always reads back 0).
6	IOCHCK	**PC/AT Channel Check Indicator** This PC/AT-compatible bit is not supported (always reads back 0).
5	PIT_OUT2_STA	**PIT Timer 2 Output Pin State** This status bit directly reflects the state of the output signal on Channel 2 of the programmable interval timer (PIT) and is sampled before the gate controlled by the PIT_OUT2_ENB bit. 0 = PIT Channel 2 output is Low. 1 = PIT Channel 2 output is High.
4	RFD	**DRAM Refresh Indicator** On the original PC/AT, the PIT Channel 1 output was used to generate the refresh signal. In this design, this bit is tied directly to the 32-KHz clock source.
3–2	Reserved	**Reserved** This bit field is ignored in the ÉlanSC520 microcontroller. On the original PC/AT, this bit field was used to enable I/O channel check and RAM parity check. For software using this register to remain PC/AT-compatible, read-modify-write operations should be used to preserve this bit field's state.
1	PIT_OUT2_ENB	**PIT Output Channel 2 Enable** This bit enables the PIT Channel 2 output to the PITOUT2 pin. 0 = Do not propagate the PIT Channel 2 output to the PITOUT2 pin. The PITOUT2 pin is driven Low. 1 = Propagate the PIT Channel 2 output to the PITOUT2 pin. The PITOUT2 pin is connected to the system speaker in PC/AT-compatible systems.
0	PIT_GATE2	**Timer 2 Gate Input Control** This bit drives the gate input signal for PIT Channel 2. The PIT Channel 2 gate input controls the channel's operation if the mode is set appropriately via the channel's CTR_MODE bit field in the PITMODECTL register (see page 13-8). 0 = The PIT Channel 2 gate input is controlled by the PITGATE2 pin. 1 = The PIT Channel 2 gate input is asserted. If the PITGATE2 pin is configured for its alternate function, then the pin's input to the OR gate is held Low, and this bit (PIT_GATE2) effectively controls the Channel 2 gate input directly. ***Note:*** *This bit (PIT_GATE2) is logically ORed with the PITGATE2 signal. Software using this bit should consider the configuration and state of the PITGATE2 pin.*

Programming Notes

CHAPTER

14 GENERAL-PURPOSE TIMER REGISTERS

14.1 OVERVIEW

This chapter describes the general-purpose (GP) timer registers of the ÉlanSC520 microcontroller.

There are three GP timers. The GP timer module is one of four ÉlanSC520 microcontroller timer modules. The other timer modules are described in the following chapters:

- Chapter 15, “Software Timer Registers”
- Chapter 13, “Programmable Interval Timer Registers”
- Chapter 16, “Watchdog Timer Registers”

The GP timer register set consists of 12 memory-mapped configuration region (MMCR) registers used to configure and control each of the three GP timers. See the *Élan™SC520 Microcontroller User’s Manual*, order #22004, for details about the GP timers.

Table 14-1 lists the GP timer registers in offset order, with the corresponding description’s page number.

14.2 REGISTERS

Table 14-1 General-Purpose Timer MMCR Registers

Register Name	Mnemonic	MMCR Offset	Page Number
GP Timers Status	GPTMRSTA	C70h	page 14-2
GP Timer 0 Mode/Control	GPTMR0CTL	C72h	page 14-3
GP Timer 0 Count	GPTMR0CNT	C74h	page 14-6
GP Timer 0 Maxcount Compare A	GPTMR0MAXCMPA	C76h	page 14-7
GP Timer 0 Maxcount Compare B	GPTMR0MAXCMPB	C78h	page 14-8
GP Timer 1 Mode/Control	GPTMR1CTL	C7Ah	page 14-9
GP Timer 1 Count	GPTMR1CNT	C7Ch	page 14-12
GP Timer 1 Maxcount Compare A	GPTMR1MAXCMPA	C7Eh	page 14-13
GP Timer 1 Maxcount Compare B	GPTMR1MAXCMPB	C80h	page 14-14
GP Timer 2 Mode/Control	GPTMR2CTL	C82h	page 14-15
GP Timer 2 Count	GPTMR2CNT	C84h	page 14-17
GP Timer 2 Maxcount Compare A	GPTMR2MAXCMPA	C8Eh	page 14-18

GP Timers Status (GPTMRSTA)

Memory-Mapped
MMCR Offset C70h

	7	6	5	4	3	2	1	0
Bit	Reserved					T2_INT_STA	T1_INT_STA	T0_INT_STA
Reset	0	0	0	0	0	0	0	0
R/W	RSV					R/W!	R/W!	R/W!

Register Description

This register contains the interrupt status information for the general-purpose timers.

Bit Definitions

Bit	Name	Function
7–3	Reserved	**Reserved** This bit field should be written to 0 for normal system operation.
2	T2_INT_STA	**GP Timer 2 Interrupt Status** 0 = GP Timer 2 has not caused an interrupt, or software cleared this bit by writing 1. 1 = GP Timer 2 has caused an interrupt. This bit is set when the GP Timer 2 interrupt request is asserted. The interrupt request remains asserted until this status bit is cleared. Software must write a 1 to clear this bit.
1	T1_INT_STA	**GP Timer 1 Interrupt Status** 0 = GP Timer 1 has not caused an interrupt, or software cleared this bit by writing 1. 1 = GP Timer 1 has caused an interrupt. This bit is set when the GP Timer 1 interrupt request is asserted. The interrupt request remains asserted until this status bit is cleared. Software must write a 1 to clear this bit.
0	T0_INT_STA	**GP Timer 0 Interrupt Status** 0 = GP Timer 0 has not caused an interrupt, or software cleared this bit by writing 1. 1 = GP Timer 0 has caused an interrupt. This bit is set when the GP Timer 0 interrupt request is asserted. The interrupt request remains asserted until this status bit is cleared. Software must write a 1 to clear this bit.

Programming Notes

GP Timer 0 Mode/Control (GPTMR0CTL)

Memory-Mapped
MMCR Offset C72h

	15	14	13	12	11	10	9	8
Bit	ENB	P_ENB_WR	INT_ENB	MAX_CNT_RIU	Reserved			
Reset	0	0	0	0	0	0	0	0
R/W	R/W!	W	R/W	R	RSV			

	7	6	5	4	3	2	1	0
Bit	Reserved		MAX_CNT	RTG	PSC_SEL	EXT_CLK	ALT_CMP	CONT_CMP
Reset	0	0	0	0	0	0	0	0
R/W	RSV		R/W!	R/W	R/W	R/W	R/W	R/W

Register Description

This register is used to control the functionality and modes of operation of GP Timer 0

Bit Definitions

Bit	Name	Function
15	ENB	**GP Timer 0 Enable** 0 = GP Timer 0 is inhibited from counting. 1 = GP Timer 0 counting is enabled. If GP Timer 0 was previously enabled (via setting the ENB bit) and is operating, and then software clears the ENB bit by writing a 0, then GP Timer 0 is inhibited from counting but is not reset. The various timer status bits and the TMROUT0 output pin also remain stable. In this scenario, setting this ENB bit again causes the timer to continue from the state that it was in just prior to being disabled. This bit can only be modified (set or cleared) via software, if the P_ENB_WR bit of this register is set (i.e., written as 1) during the same write cycle access to this register. The ENB bit is automatically cleared by hardware under certain circumstances if noncontinuous mode is selected. See the CONT_CMP bit description on page 14-5.
14	P_ENB_WR	**GP Timer 0 Permit Enable Bit Write** 0 = Software cannot modify the ENB bit in this write cycle. 1 = Software can modify the ENB bit in this write cycle. This bit allows selective software modifications of the ENB bit. When the P_ENB_WR bit is set during a write cycle access to this register, the ENB bit can be modified in that same write cycle. When the P_ENB_WR bit is written as a 0 during a write cycle access to this register, the ENB bit cannot be modified. This bit is always read back as a 0.

Bit	Name	Function
13	INT_ENB	**GP Timer 0 Interrupt Enable** This bit allows the timer to generate an interrupt when the timer counter value reaches a maximum count compare register value. 0 = GP Timer 0 interrupt request generation is disabled. 1 = GP Timer 0 interrupt request generation is enabled. If the INT_ENB bit is 1, the T0_INT_STA bit is set in the GPTMRSTA register (see page 14-2) and an interrupt is generated when one of the following conditions occurs: ■ The ALT_CMP bit is 1 (see page 14-5) and the GPTMR0CNT register value (page 14-6) equals the value of either register GPTMR0MAXCMPA (page 14-7) or GPTMR0MAXCMPB (page 14-8). ■ The ALT_CMP bit is 0 and the GPTMR0CNT register value equals the value of the GPTMR0MAXCMPA register only. When the INT_ENB bit is 0, the timer does not cause the T0_INT_STA bit to be set in the GPTMRSTA register (see page 14-2), and therefore a timer interrupt is not generated. Before GP Timer 0 interrupts are enabled, the GPTMR0MAP register (see page 12-21) must be configured to route the interrupt to the appropriate interrupt request level and priority.
12	MAX_CNT_RIU	**GP Timer 0 Maxcount Compare Register In Use** This bit can be used by software with the MAX_CNT bit to determine where the timer is in its current count sequence. 0 = Hardware clears this MAX_CNT_RIU bit when the GPTMR0MAXCMPA register (see page 14-7) is being used for comparison to the GP Timer 0 count value. 1 = Hardware sets this MAX_CNT_RIU bit when the GPTMR0MAXCMPB register (see page 14-8) is being used for comparison to the GP Timer 0 count value. Hardware also clears this bit any time hardware disables the timer by clearing the ENB bit (i.e., at the end of the timer count when in noncontinuous mode). See the CONT_CMP bit description on page 14-5.
11–6	Reserved	**Reserved** This bit field should be written to 0 for normal system operation.
5	MAX_CNT	**GP Timer 0 Maximum Count** This bit can be used by software with the MAX_CNT_RIU bit to determine where the timer is in its current count sequence. 0 = Software must clear this bit by writing a 0 to it. This bit is never automatically cleared by hardware. 1 = This bit is set by hardware any time the timer count value reaches a maximum count value (maximum count value A or maximum count value B). This bit cannot be set by software. When GP Timer 0 is in alternate compare mode (the ALT_CMP bit = 1), the MAX_CNT bit is set whenever the timer 0 count value equals the value of either register GPTMR0MAXCMPA (see page 14-7) or GPTMR0MAXCMPB (page 14-8). The MAX_CNT bit is set for this condition regardless of the state of the INT_ENB bit. The MAX_CNT bit can be used to monitor timer status through software polling instead of making use of interrupt generation.
4	RTG	**GP Timer 0 Retrigger** This bit determines the control function provided by the GP Timer 0 input pin (TMRIN0) when TMRIN0 is not configured as the timer clock source (i.e., when the EXT_CLK bit is 0). 0 = 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. 1 = If the timer is enabled, a 0 to 1 edge transition on the TMRIN0 pin resets the existing GP Timer 0 count value and then counting continues. This bit is ignored when external clocking is selected (i.e., when the EXT_CLK bit is 1).

Bit	Name	Function
3	PSC_SEL	**GP Timer 0 Prescaler** This bit selects the GP Timer 0 clock source when the TMRIN0 input pin is not configured as the timer clock source (i.e., when the EXT_CLK bit is 0). 0 = The GP Timer 0 clock source is the internal clock (33.000 MHz or 33.333 MHz, depending on the crystal frequency). 1 = The GP Timer 0 is pre-scaled by GP Timer 2 (i.e., the internal GP Timer 2 output is used as the input clock source for GP Timer 0). This bit is ignored when external clocking is enabled (i.e., when the EXT_CLK bit is 1).
2	EXT_CLK	**GP Timer 0 External Clock** This bit selects the external GP Timer 0 clock source. 0 = An internal GP Timer 0 clock source is used as configured via the PSC_SEL bit. When the timer clock is not being sourced from GP Timer 2, the timer advances every 4th CPU clock period. 1 = An external GP Timer 0 clock source is used (i.e., the TMRIN0 pin). GP Timer 0 advances upon every positive edge driven on the TMRIN0 input pin. In this mode, the maximum timer input clock frequency is 1/4th of the CPU clock speed.
1	ALT_CMP	**GP Timer 0 Alternate Compare** This bit selects whether the GP Timer 0 count is compared to a single maximum count register value, or alternately to both maximum count register values. 0 = Single compare mode: the timer counts to the GPTMR0MAXCMPA register value (see page 14-7) and then resets the GPTMR0CNT register value to 0 (page 14-6). In this mode, the GPTMR0MAXCMPB register is not used. In single compare mode, the TMROUT0 pin is high while the counter is counting and being compared to the GPTMR0MAXCMPA register. The TMROUT0 pin is pulsed Low for a single CPU clock cycle after the maximum value is reached. 1 = Alternate compare (square wave) mode: if the timer is enabled, the timer counts to the GPTMR0MAXCMPA register value and then resets the GPTMR0CNT register to 0. Then the timer counts to the GPTMR0MAXCMPB register value (page 14-8) and then resets the GPTMR0CNT register value to 0. In alternate compare mode, the TMROUT0 pin is high while the counter is counting and being compared to the GPTMR0MAXCMPA register. The TMROUT0 pin is Low while the counter is counting and being compared to the GPTMR0MAXCMPB register. If the CONT_CMP bit is set, alternate compare mode generates a square wave signal on the TMROUT0 pin with a frequency and duty cycle determined by the two maximum count register values. ***Note:*** *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 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_CMP bit description for a more detailed description of how the comparison registers are used in continuous and noncontinuous modes.
0	CONT_CMP	**GP Timer 0 Continuous Mode** This bit is used to configure GP Timer 0 for continuous or noncontinuous mode. 0 = Noncontinuous mode: the GPTMR0CNT register (see page 14-12) is cleared and the timer halts whenever the count reaches the maximum count value. The ENB bit is also cleared by hardware after every counter sequence. 1 = Continuous mode: The timer count is reset to 0 after it reaches the maximum count value (A or B), and the timer immediately begins counting again. If the CONT_CMP bit is cleared and the ALT_CMP bit is set, GP Timer 0 counts to the GPTMR0MAXCMPA register value (see page 14-13) and then resets the count value. After the timer count has been reset, the timer continues operation by counting to the GPTMR0MAXCMPB register value (page 14-14). When the timer count reaches the GPTMR0MAXCMPB register value, the timer clears its count value, clears the ENB bit and then halts.

Programming Notes

GP Timer 0 Count (GPTMR0CNT)

Memory-Mapped
MMCR Offset C74h

	15	14	13	12	11	10	9	8
Bit	CNT[15–8]							
Reset	0	0	0	0	0	0	0	0
R/W	R/W							

	7	6	5	4	3	2	1	0
Bit	CNT[7–0]							
Reset	0	0	0	0	0	0	0	0
R/W	R/W							

Register Description

This register contains the current count of GP Timer 0.

Bit Definitions

Bit	Name	Function
15–0	CNT[15–0]	**GP Timer 0 Count Register** This bit field contains the current count of GP Timer 0. The count rate depends on the value of the PSC_SEL and EXT_CLK bits in the GPTMR0CTL register (see page 14-5 and page 14-5). ■ If the EXT_CLK and PSC_SEL bits are both 0, the count is incremented every fourth processor clock cycle. ■ If the EXT_CLK bit is 0 and the PSC_SEL bit is 1, the count is incremented each time the GP Timer 2 maxcount is reached. ■ If the EXT_CLK bit is 1, the count is incremented on every positive edge driven on the TMRIN0 input pin (up to 1/4 of the CPU clock speed). This register (GPTMR0CNT) can be read at any time to determine the remaining count duration until a maximum count value is reached, at which time this register is reset by hardware. This register can also be written at any time. If this register is written while the counter is enabled, the new value is latched into the GP Timer 0 counter and counting proceeds from this new value.

Programming Notes

Each time this GPTMR0CNT register is incremented, its value is compared with the value of register GPTMR0MAXCMPA or GPTMR0MAXCMPB (see page 14-7 and page 14-8) and various actions are taken when a maximum count is reached. For details, see the GPTMR0CTL register bits INT_ENB, MAX_CNT_RIU, MAX_CNT, ALT_CMP, and CONT_CMP, starting on page 14-4.

GP Timer 0 Maxcount Compare A (GPTMR0MAXCMPA)

Memory-Mapped
MMCR Offset C76h

	15	14	13	12	11	10	9	8
Bit	MCA[15–8]							
Reset	0	0	0	0	0	0	0	0
R/W	R/W							

	7	6	5	4	3	2	1	0
Bit	MCA[7–0]							
Reset	0	0	0	0	0	0	0	0
R/W	R/W							

Register Description

This register contains one of the compare values for the GPTMR0CNT register (see page 14-6).

Bit Definitions

Bit	Name	Function
15–0	MCA[15–0]	**GP Timer 0 Maxcount Compare Register A** This register contains one of the maximum values that GP Timer 0 can count to before resetting its count register to 0.

Programming Notes

GP Timer 0 and GP Timer 1 each have two maxcount compare registers, GPTMRxMAXCMPA and GPTMRxMAXCMPB.

If the maxcount compare register that is in use contains 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. For details, see the GPTMR0CTL register bits INT_ENB, MAX_CNT_RIU, MAX_CNT, ALT_CMP, and CONT_CMP, starting on page 14-4.

If the maxcount compare register that is in use contains a value other than 0000h and the timer is enabled, the timer counts to the programmed maxcount value.

GP Timer 0 Maxcount Compare B (GPTMR0MAXCMPB)

Memory-Mapped
MMCR Offset C78h

	15	14	13	12	11	10	9	8
Bit	MCB[15–8]							
Reset	0	0	0	0	0	0	0	0
R/W	R/W							

	7	6	5	4	3	2	1	0
Bit	MCB[7–0]							
Reset	0	0	0	0	0	0	0	0
R/W	R/W							

Register Description

This register contains one of the compare values for the GPTMR0CNT register (see page 14-6).

Bit Definitions

Bit	Name	Function
15–0	MCB[15–0]	**GP Timer 0 Maxcount Compare Register B** This register contains one of the maximum values that GP Timer 0 can count to before resetting its count register to 0.

Programming Notes

GP Timer 0 and GP Timer 1 each have two maxcount compare registers, GPTMRxMAXCMPA and GPTMRxMAXCMPB.

If the maxcount compare register that is in use contains 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. For details, see the GPTMR0CTL register bits INT_ENB, MAX_CNT_RIU, MAX_CNT, ALT_CMP, and CONT_CMP, starting on page 14-4.

If the maxcount compare register that is in use contains a value other than 0000h and the timer is enabled, the timer counts to the programmed maxcount value.

GP Timer 1 Mode/Control (GPTMR1CTL)

Memory-Mapped
MMCR Offset C7Ah

	15	14	13	12	11	10	9	8
Bit	ENB	P_ENB_WR	INT_ENB	MAX_CNT_RIU	Reserved			
Reset	0	0	0	0	0	0	0	0
R/W	R/W!	W	R/W	R	RSV			

	7	6	5	4	3	2	1	0
Bit	Reserved		MAX_CNT	RTG	PSC_SEL	EXT_CLK	ALT_CMP	CONT_CMP
Reset	0	0	0	0	0	0	0	0
R/W	RSV		R/W!	R/W	R/W	R/W	R/W	R/W

Register Description

This register is used to control the functionality and modes of operation of GP Timer 1.

Bit Definitions

Bit	Name	Function
15	ENB	**GP Timer 1 Enable** 0 = GP Timer 1 is inhibited from counting. 1 = GP Timer 1 counting is enabled. If GP Timer 1 was previously enabled (via setting the ENB bit) and is operating, and then software clears the ENB bit by writing a 0, then GP Timer 1 is inhibited from counting but is not reset. The various timer status bits and the TMROUT1 output pin also remain stable. In this scenario, setting this ENB bit again causes the timer to continue from the state that it was in just prior to being disabled. This bit can only be modified (set or cleared) via software, if the P_ENB_WR bit of this register is set (i.e., written as 1) during the same write cycle access to this register. The ENB bit is automatically cleared by hardware under certain circumstances if noncontinuous mode is selected. See the CONT_CMP bit description on page 14-11.
14	P_ENB_WR	**GP Timer 1 Permit Enable Bit Write** 0 = Software cannot modify the ENB bit in this write cycle. 1 = Software can modify the ENB bit in this write cycle. This bit allows selective software modifications of the ENB bit. When the P_ENB_WR bit is set during a write cycle access to this register, the ENB bit can be modified in that same write cycle. When the P_ENB_WR bit is written as a 0 during a write cycle access to this register, the ENB bit cannot be modified. This bit is always read back as a 0.

Bit	Name	Function
13	INT_ENB	**GP Timer 1 Interrupt Enable** This bit allows the timer to generate an interrupt when the timer counter value reaches a maximum count compare register value. 0 = GP Timer 1 interrupt request generation is disabled. 1 = GP Timer 1 interrupt request generation is enabled. If the INT_ENB bit is 1, the T1_INT_STA bit is set in the GPTMRSTA register (see page 14-2) and an interrupt is generated when one of the following conditions occurs: ■ The ALT_CMP bit is 1 (see page 14-11) and the GPTMR1CNT register value (page 14-12) equals the value of either register GPTMR1MAXCMPA (page 14-13) or GPTMR1MAXCMPB (page 14-14). ■ The ALT_CMP bit is 0 and the GPTMR1CNT register value equals the value of the GPTMR1MAXCMPA register only. When the INT_ENB bit is 0, the timer does not cause the T1_INT_STA bit to be set in the GPTMRSTA register (see page 14-2), and therefore a timer interrupt is not generated. Before GP Timer 1 interrupts are enabled, the GPTMR1MAP register (see page 12-21) must be configured to route the interrupt to the appropriate interrupt request level and priority.
12	MAX_CNT_RIU	**GP Timer 1 Maxcount Compare Register In Use** This bit can be used by software with the MAX_CNT bit to determine where the timer is in its current count sequence. 0 = Hardware clears this MAX_CNT_RIU bit when the GPTMR1MAXCMPA register (see page 14-13) is being used for comparison to the GP Timer 1 count value. 1 = Hardware sets this MAX_CNT_RIU bit when the GPTMR1MAXCMPB register (see page 14-14) is being used for comparison to the GP Timer 1 count value. Hardware also clears this bit any time hardware disables the timer by clearing the ENB bit (i.e., at the end of the timer count when in noncontinuous mode). See the CONT_CMP bit description on page 14-11.
11–6	Reserved	**Reserved** This bit field should be written to 0 for normal system operation.
5	MAX_CNT	**GP Timer 1 Maximum Count** This bit can be used by software with the MAX_CNT_RIU bit to determine where the timer is in its current count sequence. 0 = Software must clear this bit by writing a 0 to it. This bit is never automatically cleared by hardware. 1 = This bit is set by hardware any time the timer count value reaches a maximum count value (maximum count value A or maximum count value B). This bit cannot be set by software. When GP Timer 1 is in alternate compare mode (the ALT_CMP bit = 1), the MAX_CNT bit is set whenever the GP Timer 1 count value equals the value of either register GPTMR1MAXCMPA (see page 14-13) or GPTMR1MAXCMPB (page 14-14). The MAX_CNT bit is set for this condition regardless of the state of the INT_ENB bit. The MAX_CNT bit can be used to monitor timer status through software polling instead of making use of interrupt generation.
4	RTG	**GP Timer 1 Retrigger** This bit determines the control function provided by the GP Timer 1 input pin (TMRIN1) when TMRIN1 is not configured as the timer clock source (i.e., when the EXT_CLK bit is 0). 0 = 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. 1 = If the timer is enabled, a 0 to 1 edge transition on the TMRIN1 pin resets the existing GP Timer 1 count value and then counting continues. This bit is ignored when external clocking is selected (i.e., when the EXT_CLK bit is 1).

Bit	Name	Function
3	PSC_SEL	**GP Timer 1 Prescaler** This bit selects the GP Timer 1 clock source when the TMRIN1 input pin is not configured as the timer clock source (i.e., when the EXT_CLK bit is 0). 0 = The GP Timer 1 clock source is the internal clock (33.000 MHz or 33.333 MHz, depending on the crystal frequency). 1 = The GP Timer 1 is pre-scaled by GP Timer 2 (i.e., the internal GP Timer 2 output is used as the input clock source for GP Timer 1). This bit is ignored when external clocking is enabled (i.e., the EXT_CLK bit is 1).
2	EXT_CLK	**GP Timer 1 External Clock** This bit selects the external GP Timer 1 clock source. 0 = An internal GP Timer 1 clock source is used as configured via the PSC_SEL bit. When the timer clock is not being sourced from GP Timer 2, the timer advances every 4th CPU clock period. 1 = An external GP Timer 1 clock source is used (i.e., the TMRIN1 pin). GP Timer 1 advances upon every positive edge driven on the TMRIN1 input pin. In this mode, the maximum timer input clock frequency is 1/4th of the CPU clock speed.
1	ALT_CMP	**GP Timer 1 Alternate Compare** This bit selects whether the GP Timer 1 count is compared to a single maximum count register value, or alternately to both maximum count register values. 0 = Single compare mode: the timer counts to the GPTMR1MAXCMPA register value (see page 14-13) and then resets the GPTMR1CNT register value to 0 (page 14-12). In this mode, the GPTMR1MAXCMPB register is not used. In single compare mode, the TMROUT1 pin is high while the counter is counting and being compared to the GPTMR1MAXCMPA register. The TMROUT1 pin is pulsed Low for a single CPU clock cycle after the maximum value is reached. 1 = Alternate compare (square wave) mode: if the timer is enabled, the timer counts to the GPTMR1MAXCMPA register value and then resets the GPTMR1CNT register to 0. Then the timer counts to the GPTMR1MAXCMPB register value (page 14-14) and then resets the GPTMR1CNT register value to 0. In alternate compare mode, the TMROUT1 pin is high while the counter is counting and being compared to the GPTMR1MAXCMPA register. The TMROUT1 pin is Low while the counter is counting and being compared to the GPTMR1MAXCMPB register. If the CONT_CMP bit is set, alternate compare mode generates a square wave signal on the TMROUT1 pin with a frequency and duty cycle determined by the two maximum count register values. ***Note:*** *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 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 Continuous mode bit description below for a more detailed description of how the comparison registers are used in continuous and noncontinuous modes.
0	CONT_CMP	**GP Timer 1 Continuous Mode** This bit is used to configure GP Timer 1 for continuous or noncontinuous mode. 0 = Noncontinuous mode: the GPTMR1CNT register (see page 14-12) is cleared and the timer halts whenever the count reaches the maximum count value. The ENB bit is also cleared by hardware after every counter sequence. 1 = Continuous mode: The timer count is reset to 0 after it reaches the maximum count value (A or B), and the timer immediately begins counting again. If the CONT_CMP bit is cleared and the ALT_CMP bit is set, GP Timer 1 counts to the GPTMR1MAXCMPA register value (see page 14-13) and then resets the count value. After the timer count has been reset, the timer continues operation by counting to the GPTMR1MAXCMPB register value (page 14-14). When the timer count reaches the GPTMR1MAXCMPB register value, the timer clears its count value, clears the ENB bit and then halts.

Programming Notes

GP Timer 1 Count (GPTMR1CNT)

Memory-Mapped
MMCR Offset C7Ch

	15	14	13	12	11	10	9	8
Bit	CNT[15–8]							
Reset	0	0	0	0	0	0	0	0
R/W	R/W							

	7	6	5	4	3	2	1	0
Bit	CNT[7–0]							
Reset	0	0	0	0	0	0	0	0
R/W	R/W							

Register Description

This register contain the current count of GP Timer 1.

Bit Definitions

Bit	Name	Function
15–0	CNT[15–0]	**GP Timer 1 Count Register** This bit field contains the current count of GP Timer 1. The count rate depends on the value of the PSC_SEL and EXT_CLK bits in the GPTMR1CTL register (see page 14-11 and page 14-11). ■ If the EXT_CLK and PSC_SEL bits are both 0, the count is incremented every fourth processor clock cycle. ■ If the EXT_CLK bit is 0 and the PSC_SEL bit is 1, the count is incremented each time the GP Timer 2 maxcount is reached. ■ If the EXT_CLK bit is 1, the count is incremented on every positive edge driven on the TMRIN1 input pin (up to 1/4 of the CPU clock speed). This register (GPTMR1CNT) can be read at any time to determine the remaining count duration until a maximum count value is reached, at which time this register is reset by hardware. This register can also be written at any time. If this register is written while the counter is enabled, the new value is latched into the GP Timer 1 counter and counting proceeds from this new value.

Programming Notes

Each time this GPTMR1CNT register is incremented, its value is compared with the value of register GPTMR1MAXCMPA or GPTMR1MAXCMPB (see page 14-13 and page 14-14) and various actions are taken when a maximum count is reached. For details, see the GPTMR1CTL register bits INT_ENB, MAX_CNT_RIU, MAX_CNT, ALT_CMP, and CONT_CMP, starting on page 14-10.

GP Timer 1 Maxcount Compare A (GPTMR1MAXCMPA)

Memory-Mapped
MMCR Offset C7Eh

	15	14	13	12	11	10	9	8
Bit	MCA[15–8]							
Reset	0	0	0	0	0	0	0	0
R/W	R/W							

	7	6	5	4	3	2	1	0
Bit	MCA[7–0]							
Reset	0	0	0	0	0	0	0	0
R/W	R/W							

Register Description

This register contains one of the compare values for the GPTMR1CNT register (see page 14-12).

Bit Definitions

Bit	Name	Function
15–0	MCA[15–0]	**GP Timer 1 Maxcount Compare Register A** This register contains one of the maximum values that GP Timer 1 can count to before resetting its count register to 0.

Programming Notes

GP Timer 0 and GP Timer 1 each have two maxcount compare registers, GPTMRxMAXCMPA and GPTMRxMAXCMPB.

If the maxcount compare register that is in use contains 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. For details, see the GPTMR1CTL register bits INT_ENB, MAX_CNT_RIU, MAX_CNT, ALT_CMP, and CONT_CMP, starting on page 14-10.

If the maxcount compare register that is in use contains a value other than 0000h and the timer is enabled, the timer counts to the programmed maxcount value.

GP Timer 1 Maxcount Compare B (GPTMR1MAXCMPB)

Memory-Mapped
MMCR Offset C80h

	15	14	13	12	11	10	9	8
Bit	MCB[15–8]							
Reset	0	0	0	0	0	0	0	0
R/W	R/W							

	7	6	5	4	3	2	1	0
Bit	MCB[7–0]							
Reset	0	0	0	0	0	0	0	0
R/W	R/W							

Register Description

This register contains one of the compare values for the GPTMR1CNT register (see page 14-12).

Bit Definitions

Bit	Name	Function
15–0	MCB[15–0]	**GP Timer 1 Maxcount Compare Register B** This register contains one of the maximum values that GP Timer 1 can count to before resetting its count register to 0.

Programming Notes

GP Timer 0 and GP Timer 1 each have two maxcount compare registers, GPTMRxMAXCMPA and GPTMRxMAXCMPB.

If the maxcount compare register that is in use contains 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. For details, see the GPTMR1CTL register bits INT_ENB, MAX_CNT_RIU, MAX_CNT, ALT_CMP, and CONT_CMP, starting on page 14-10.

If the maxcount compare register that is in use contains a value other than 0000h and the timer is enabled, the timer counts to the programmed maxcount value.

GP Timer 2 Mode/Control (GPTMR2CTL)

Memory-Mapped
MMCR Offset C82h

	15	14	13	12	11	10	9	8
Bit	ENB	P_ENB_WR	INT_ENB	Reserved				
Reset	0	0	0	0	0	0	0	0
R/W	R/W!	W	R/W	RSV				

	7	6	5	4	3	2	1	0
Bit	Reserved		MAX_CNT	Reserved				CONT_ CMP
Reset	0	0	0	0	0	0	0	0
R/W	RSV		R/W!	RSV				R/W

Register Description

This register is used to control the functionality and modes of operation of GP Timer 2.

Bit Definitions

Bit	Name	Function
15	ENB	**GP Timer 2 Enable** 0 = GP Timer 2 is inhibited from counting. 1 = GP Timer 2 counting is enabled. If GP Timer 2 was previously enabled (via setting the ENB bit) and is operating, and then software clears the ENB bit by writing a 0, then GP Timer 2 is inhibited from counting but is not reset. The various timer status bits also remain stable. In this scenario, setting this ENB bit again causes the timer to continue from the state that it was in just prior to being disabled. This bit can only be modified (set or cleared) via software, if the P_ENB_WR bit of this register is set (i.e., written as 1) during the same write cycle access to this register. The ENB bit is automatically cleared by hardware under certain circumstances if noncontinuous mode is selected. See the CONT_CMP bit description on page 14-16.
14	P_ENB_WR	**GP Timer 2 Permit Enable Bit Write** 0 = Software cannot modify the ENB bit in this write cycle. 1 = Software can modify the ENB bit in this write cycle. This bit allows selective software modifications of the ENB bit. When the P_ENB_WR bit is set during a write cycle access to this register, the ENB bit can be modified in that same write cycle. When the P_ENB_WR bit is written as a 0 during a write cycle access to this register, the ENB bit cannot be modified. This bit is always read back as a 0.

Bit	Name	Function
13	INT_ENB	**GP Timer 2 Interrupt Enable** This bit allows the timer to generate an interrupt when the timer count value reaches the maximum count compare register value. 0 = GP Timer 2 interrupt request generation is disabled. 1 = GP Timer 2 interrupt request generation is enabled. If the INT_ENB bit is 1, the T2_INT_STA bit is set in the GPTMRSTA register (see page 14-2) and an interrupt is generated when the GPTMR2CNT register value (page 14-17) equals the value of the GPTMR2MAXCMPA register (page 14-18). When the INT_ENB bit is 0, the timer does not cause the T2_INT_STA bit to be set in the GPTMRSTA register (see page 14-2), and therefore a timer interrupt is not generated. Before GP Timer 2 interrupts are enabled, the GPTMR2MAP register (see page 12-21) must be configured to route the interrupt to the appropriate interrupt request level and priority.
12–6	Reserved	**Reserved** This bit field should be written to 0 for normal system operation.
5	MAX_CNT	**GP Timer 2 Maximum Count** This bit can be used by software to determine where the timer is in its current count sequence. 0 = Software must clear this bit by writing a 0 to it. This bit is never automatically cleared by hardware. 1 = This bit is set by hardware any time the timer count value reaches its maximum count value (GPTMR2MAXCMPA register, see page 14-18). The MAX_CNT bit is set for this condition regardless of the state of the INT_ENB bit. The MAX_CNT bit can be used to monitor timer status through software polling instead of making use of interrupt generation.
4–1	Reserved	**Reserved** This bit field should be written to 0 for normal system operation.
0	CONT_CMP	**GP Timer 2 Continuous Mode** This bit is used to configure GP Timer 2 for continuous or noncontinuous mode. 0 = Noncontinuous mode: the GPTMR2CNT register (see page 14-12) is cleared and the timer halts whenever the count reaches the maximum count value (GPTMR2MAXCMPA register, see page 14-18). The ENB bit is also cleared by hardware after every counter sequence. 1 = Continuous mode: The timer count is reset to 0 after it reaches the maximum count value, and the timer immediately begins counting again.

Programming Notes

GP Timer 2 Count (GPTMR2CNT)

Memory-Mapped
MMCR Offset C84h

	15	14	13	12	11	10	9	8
Bit	CNT[15–8]							
Reset	0	0	0	0	0	0	0	0
R/W	R/W							

	7	6	5	4	3	2	1	0
Bit	CNT[7–0]							
Reset	0	0	0	0	0	0	0	0
R/W	R/W							

Register Description

This register contains the current count of GP Timer 2.

Bit Definitions

Bit	Name	Function
15–0	CNT[15–0]	**GP Timer 2 Count Register** This bit field contains the current count of GP Timer 2. The count is incremented every fourth processor clock cycle. This register (GPTMR2CNT) can be read at any time to determine the remaining count duration until a maximum count value is reached, at which time this register is reset by hardware. This GPTMR2CNT register can also be written at any time. If this register is written while the counter is enabled, the new value is latched into the GP Timer 2 counter and counting proceeds from this new value.

Programming Notes

Each time this GPTMR2CNT register is incremented, its value is compared with the GPTMR2MAXCMPA register (see page 14-18) and various actions are taken when a maximum count is reached. For details, see the GPTMR2CTL register bits INT_ENB, MAX_CNT, and CONT_CMP, starting on page 14-16.

GP Timer 2 Maxcount Compare A (GPTMR2MAXCMPA)

Memory-Mapped
MMCR Offset C8Eh

	15	14	13	12	11	10	9	8
Bit	MCA[15–8]							
Reset	0	0	0	0	0	0	0	0
R/W	R/W							

	7	6	5	4	3	2	1	0
Bit	MCA[7–0]							
Reset	0	0	0	0	0	0	0	0
R/W	R/W							

Register Description

This register contains the single compare values for the GPTMR2CNT register (see page 14-17).

Bit Definitions

Bit	Name	Function
15–0	MCA[15–0]	**GP Timer 2 Maxcount Compare Register A** This register contains the maximum value that GP Timer 2 can count to before resetting its count register to 0.

Programming Notes

GP Timer 2 has only one maxcount compare register. If this register (GPTMR2MAXCMPA) 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. For details, see the GPTMR2CTL register bits INT_ENB, MAX_CNT, and CONT_CMP, starting on page 14-16.

If the maxcount compare register contains a value other than 0000h and the timer is enabled, the timer counts to the programmed maxcount value.

CHAPTER

15 SOFTWARE TIMER REGISTERS

15.1 OVERVIEW

This chapter describes the software timer registers of the ÉlanSC520 microcontroller.

The software timer is one of four ÉlanSC520 microcontroller timer modules. The other timer modules are described in the following chapters:

- Chapter 14, “General-Purpose Timer Registers”
- Chapter 13, “Programmable Interval Timer Registers”
- Chapter 16, “Watchdog Timer Registers”

The software timer register set consists of three memory-mapped configuration region (MMCR) registers used access and control the timer. See the *Élan™SC520 Microcontroller User’s Manual*, order #22004, for details about the software timer.

Table 15-1 lists the software timer registers in offset order, with the corresponding description’s page number.

15.2 REGISTERS

Table 15-1 Software Timer MMCR Registers

Register Name	Mnemonic	MMCR Offset	Page Number
Software Timer Millisecond Count	SWTMRMILLI	C60h	page 15-2
Software Timer Microsecond Count	SWTMRMICRO	C62h	page 15-3
Software Timer Configuration	SWTMRCFG	C64h	page 15-4

Software Timer Millisecond Count (SWTMRMILLI)

Memory-Mapped
MMCR Offset C60h

	15	14	13	12	11	10	9	8
Bit	MS_CNT[15–8]							
Reset	0	0	0	0	0	0	0	0
R/W	R!							

	7	6	5	4	3	2	1	0
Bit	MS_CNT[7–0]							
Reset	0	0	0	0	0	0	0	0
R/W	R!							

Register Description

This register contains the current millisecond count of the software timer.

Bit Definitions

Bit	Name	Function
15–0	MS_CNT[15–0]	**16-bit Millisecond Count** This read-only bit field increments at a rate of 1000 counts per second. This millisecond counter is set to zero at system reset. ***Note:*** *When read, this counter is automatically reset to 0.* This millisecond counter is *not* reset when the US_CNT bit field in the SWTMRMICRO register is read (see page 15-3). In order to maintain a millisecond time base, this MS_CNT bit field must be read at least once every 65.5 seconds. When this MS_CNT bit field is read, the value in the internal free-running microsecond up counter is automatically latched into the US_CNT bit field in the SWTMRMICRO register. The internal microsecond counter increments at a rate of 1 MHz and counts from 0 to 999. It rolls over from 999 back to zero. Every time the microsecond counter rolls over, the MS_CNT bit field is incremented. Note that the XTAL_FREQ bit in the SWTMRCFG register (see page 15-4) must be set appropriately in order for the increment rate to be correct.

Programming Notes

A 32-bit read of the SWTMRMILLI register address is broken up into two 16-bit reads. The first 16-bit read returns the SWTMRMILLI register and latches the SWTMRMICRO register (see page 15-3). The second 16-bit read returns the (newly latched) SWTMRMICRO register contents. Thus, the 32-bit value returned includes the correct millisecond and microsecond values at the time of software's 32-bit read, with the millisecond value stored in the lower 16 bits and the microsecond value stored in the upper 16 bits.

Byte (8-bit) reads of the SWTMRMILLI register are not allowed.

Software Timer Microsecond Count (SWTMRMICRO)

Memory-Mapped
MMCR Offset C62h

	15	14	13	12	11	10	9	8
Bit	Reserved						US_CNT[9–8]	
Reset	0	0	0	0	0	0	0	0
R/W	RSV						R	

	7	6	5	4	3	2	1	0
Bit	US_CNT[7–0]							
Reset	0	0	0	0	0	0	0	0
R/W	R							

Register Description

This register contains the current latched microsecond count of the software timer.

Bit Definitions

Bit	Name	Function
15–10	Reserved	**Reserved** This bit field should be written to 0 for normal system operation.
9–0	US_CNT [9–0]	**SWT Microsecond Count** This read-only bit field holds the latched microsecond count value from a free running microsecond counter. Each read of this bit field returns the currently latched microsecond count value. On each read of the SWTMRMILLI register (see page 15-2), the value in the internal microsecond counter is latched into this bit field (US_CNT). This US_CNT bit field is reset to 0 at system reset. The internal microsecond counter increments at a rate of 1 MHz and counts from 0 to 999. It rolls over from 999 back to 0. Every time the microsecond counter rolls over, the MS_CNT bit field in the SWTMRMILLI register is incremented (see page 15-2). Note that the XTAL_FREQ bit in the SWTMRCFG register (see page 15-4) must be set appropriately in order for the increment rate to be correct.

Programming Notes

Software Timer Configuration (SWTMRCFG)

Memory-Mapped
MMCR Offset C64h

	7	6	5	4	3	2	1	0
Bit	Reserved							XTAL_ FREQ
Reset	0	0	0	0	0	0	0	0
R/W	RSV							R/W

Register Description

This register is used to calibrate the software timer to the crystal frequency being used.

Bit Definitions

Bit	Name	Function
7–1	Reserved	**Reserved** This bit field should be written to 0 for normal system operation.
0	XTAL_FREQ	**Crystal Frequency** This field is used to specify the frequency of the crystal used to drive the main system clock. Configure this bit appropriately to ensure that the software timer increments at the correct rate for the system. 0 = Specifies that the system is using a 33.333 MHz crystal. 1 = Specifies that the system is using a 33.000 MHz crystal. Note that this bit has no effect on the general-purpose timers, programmable interval timer, or watchdog timer.

Programming Notes

CHAPTER

16 WATCHDOG TIMER REGISTERS

16.1 OVERVIEW

This chapter describes the watchdog timer registers of the ÉlanSC520 microcontroller.

The watchdog timer is one of four ÉlanSC520 microcontroller timer modules. The other timer modules are described in the following chapters:

- Chapter 15, “Software Timer Registers”
- Chapter 14, “General-Purpose Timer Registers”
- Chapter 13, “Programmable Interval Timer Registers”

The watchdog timer register set consists of three memory-mapped configuration region (MMCR) registers used to configure, control, and monitor the status of the watchdog timer. See the *Élan™SC520 Microcontroller User’s Manual*, order #22004, for details about the watchdog timer.

Table 16-1 lists the watchdog timer registers in offset order, with the corresponding description’s page number.

16.2 REGISTERS

Table 16-1 Watchdog Timer MMCR Registers

Register Name	Mnemonic	MMCR Offset	Page Number
Watchdog Timer Control	WDTMRCTL	CB0h	page 16-2
Watchdog Timer Count Low	WDTMRCNTL	CB2h	page 16-4
Watchdog Timer Count High	WDTMRCNTH	CB4h	page 16-5
Reserved	—	CB6h	—

Watchdog Timer Control (WDTMRCTL)

Memory-Mapped
MMCR Offset CB0h

	15	14	13	12	11	10	9	8
Bit	ENB	WRST_ENB	Reserved	IRQ_FLG	Reserved			
Reset	0	1	0	0	0	0	0	0
R/W	R/W!	R/W!	RSV	R/W!	RSV			

	7	6	5	4	3	2	1	0
Bit	EXP_SEL[7–0]							
Reset	1	0	0	0	0	0	0	0
R/W	R/W!							

Register Description

This register is used to control the watchdog timer. This register can only be accessed after writing special keyed sequences as described in the programming notes.

Bit Definitions

Bit	Name	Function
15	ENB	**Watchdog Timer Enable** This is the watchdog timer enable bit. 0 = The watchdog timer is disabled. 1 = The watchdog timer is enabled. When this bit is set, the current count is automatically reset to 0, the WRST_ENB and EXP_SEL fields become read-only, and the watchdog timer counter begins counting. Before the watchdog timer is enabled with the WRST_ENB bit cleared, the WDTMAP register (see page 12-21) must be configured to route the interrupt to the appropriate interrupt request level and priority.
14	WRST_ENB	**Watchdog Timer Reset Enable** This is a programmable bit to select between a reset or interrupt as the time-out action. 0 = The watchdog timer time-out generates an interrupt request if the IRQ_FLG bit is not already set. If IRQ_FLG bit is set, a system reset is generated instead. 1 = The watchdog timer time-out always generates a system reset. When the watchdog timer generates a system reset, the WDT_RST_DET bit is set in the RESSTA register (see page 3-5).
13	Reserved	**Reserved** This bit field should be written to 0 for normal system operation.
12	IRQ_FLG	**Interrupt Request Flag** This bit provides an indication of interrupt events generated by the watchdog timer time-out. This bit is accessible even after the ENB bit is set. 0 = No watchdog timer interrupt has occurred. 1 = Watchdog timer interrupt has occurred. This bit is cleared by writing a 1. After a watchdog timer time-out occurs, if this bit is not cleared before a second time-out of the watchdog timer, a watchdog timer system reset is generated instead of a second interrupt request.
11-8	Reserved	**Reserved** This bit field should be written to 0 for normal system operation.

Bit	Name	Function
7-0	EXP_SEL[7–0]	**Exponent Select** This bit field determines the duration of the watchdog timer time-out interval. Table 16-2 shows the values for the EXP_SEL field. The bit values shown in the table are programmed to select a time-out exponent as shown in the “Exponent” column. The selected exponent determines the time-out duration according to the following equation: *Time-out duration* = $2^{exponent}$ / *CPU Frequency* where *Time-out duration* is the time-out period in seconds *exponent* is the value selected from Table 16-2 *CPU Frequency* is the operating speed of the CPU in Hz. When multiple bits are set in the EXP_SEL field, the least significant bit set is used to select the exponent.

Table 16-2 Watchdog Timer Exponent Selections

Bits 7	6	5	4	3	2	1	0	Exponent	Time-Out Interval 33.000 MHz	Time-Out Interval 33.333 MHz
0	0	0	0	0	0	0	0	N/A	Infinity	Infinity
X	X	X	X	X	X	X	1	14	496 µs	492 µs
X	X	X	X	X	X	1	0	24	508 ms	503 ms
X	X	X	X	X	1	0	0	25	1.02 s	1.01 s
X	X	X	X	1	0	0	0	26	2.03 s	2.01 s
X	X	X	1	0	0	0	0	27	4.07 s	4.03 s
X	X	1	0	0	0	0	0	28	8.13 s	8.05 s
X	1	0	0	0	0	0	0	29	16.27 s	16.11 s
1	0	0	0	0	0	0	0	30	32.54 s	32.21 s

For example, to program a maximum time-out of about 32 seconds, the EXP_SEL field is set to 80h. The time-out value can then be calculated as follows for the ÉlanSC520 microcontroller with a 33.000 MHz CPU clock.

Time-out interval = 2^{30} / (33.000 MHz crystal frequency)

= 2^{30} / (33,000,000)

= 32.54 seconds.

Programming Notes

The watchdog timer can only be programmed after special keyed sequences are written to this address. Two special keyed sequences are recognized:

- The key sequence of 3333h followed by CCCCh is called the write key and is used to open this Watchdog Timer Control (WDTMRCTL) register for a single write.
- The key sequence of AAAAh followed by 5555h is called the clear-count key and is used to clear the current watchdog timer counter.

Note: *All keys are written to the WDTMRCTL register address (MMCR offset CB0h).*

Normally this register is read-only. The write key sequence must be written before a value can be programmed into this register; after this single value is written, the write key sequence must be applied again before another value can be programmed.

The ENB bit must be 0 before the WRST_ENB bit or the EXP_SEL bit field can be written.

When the microcontroller is in AMDebug™ technology mode, the WDTMRCTL register can still be accessed in the normal manner. However, the AMDebug technology stops the count registers from incrementing further to prevent the watchdog timer from causing an interrupt or reset in AMDebug technology mode.

Watchdog Timer Count Low (WDTMRCNTL)

Memory-Mapped
MMCR Offset CB2h

	15	14	13	12	11	10	9	8
Bit	COUNTL[15–8]							
Reset	0	0	0	0	0	0	0	0
R/W	R							

	7	6	5	4	3	2	1	0
Bit	COUNTL[7–0]							
Reset	0	0	0	0	0	0	0	0
R/W	R							

Register Description

This read-only register contains the lower 16 bits of the watchdog timer counter.

In normal operation the clear-count key sequence (AAAAh followed by 5555h) clears the watchdog timer counter. Note that the clear-count key is written to the WDTMRCTL register address (see page 16-2).

Bit Definitions

Bit	Name	Function
15–0	COUNTL[15–0]	**Current Count Low** This field contains the low word [15–0] of the watchdog timer current count. The counter value is automatically reset when the watchdog timer is enabled.

Programming Notes

Although both the WDTMRCNTH and WDTMRCNTL registers can be read with a single 32-bit CPU instruction, the 32-bit access is split into two 16-bit accesses. See the GP Timer chapter in the *Élan™SC520 Microcontroller User's Manual*, order #22004, for suggestions if it is necessary to read an accurate 32-bit value from the watchdog timer counter.

Watchdog Timer Count High (WDTMRCNTH)

Memory-Mapped
MMCR Offset CB4h

	15	14	13	12	11	10	9	8
Bit	Reserved	COUNTH[14–8]						
Reset	0	0	0	0	0	0	0	0
R/W	RSV	R						

	7	6	5	4	3	2	1	0
Bit	COUNTH[7–0]							
Reset	0	0	0	0	0	0	0	0
R/W	R							

Register Description

This read-only register contains the higher 15 bits of the watchdog timer counter.

In normal operation the clear-count key sequence (AAAAh followed by 5555h) clears the watchdog timer counter. Note that the clear-count key is written to the WDTMRCTL register address (see page 16-2).

Bit Definitions

Bit	Name	Function
15	Reserved	**Reserved** This bit field should be written to 0 for normal system operation.
14-0	COUNTH[14–0]	**Current Count High** This field contains the high word [30–16] of the watchdog timer current count. The counter value is automatically reset when the watchdog timer is enabled.

Programming Notes

Although both the WDTMRCNTH and WDTMRCNTL registers can be read with a single 32-bit CPU instruction, the 32-bit access is split into two 16-bit accesses. See the GP Timer chapter in the *Élan™SC520 Microcontroller User's Manual*, order #22004, for suggestions if it is necessary to read an accurate 32-bit value from the watchdog timer counter.

CHAPTER

17 REAL-TIME CLOCK REGISTERS

17.1 OVERVIEW

This chapter describes the real-time clock (RTC) registers of the ÉlanSC520 microcontroller.

The ÉlanSC520 microcontroller includes a PC/AT-compatible RTC. The RTC register set includes two groups of registers:

- Two direct-mapped I/O addresses used to access the RTC configuration space.
- 14 RTC-indexed configuration registers used to set, read, and configure the RTC.
- 114 bytes of RTC-indexed nonvolatile RAM locations used in PC/AT-compatible systems to store various system parameters.

See the *Élan™SC520 Microcontroller User's Manual*, order #22004, for details about the RTC.

Table 17-1 and Table 17-2 list each type of RTC register in offset order, with the corresponding description's page number.

17.2 REGISTERS

Table 17-1 Real-Time Clock Direct-Mapped Registers

Register Name	Mnemonic	I/O Address	Page Number
RTC/CMOS RAM Index	RTCIDX	0070h	page 17-2
RTC/CMOS RAM Data Port	RTCDATA	0071h	page 17-3

Table 17-2 Real-Time Clock Indexed Registers

Register Name	Mnemonic	I/O Address	RTC Index	Page Number
RTC Current Second	RTCCURSEC	70h/71h	00h	page 17-4
RTC Alarm Second	RTCALMSEC	70h/71h	01h	page 17-5
RTC Current Minute	RTCCURMIN	70h/71h	02h	page 17-6
RTC Alarm Minute	RTCALMMIN	70h/71h	03h	page 17-7
RTC Current Hour	RTCCURHR	70h/71h	04h	page 17-8
RTC Alarm Hour	RTCALMHR	70h/71h	05h	page 17-9
RTC Current Day of the Week	RTCCURDOW	70h/71h	06h	page 17-10
RTC Current Day of the Month	RTCCURDOM	70h/71h	07h	page 17-11
RTC Current Month	RTCCURMON	70h/71h	08h	page 17-12
RTC Current Year	RTCCURYR	70h/71h	09h	page 17-13
RTC Control A	RTCCTLA	70h/71h	0Ah	page 17-14
RTC Control B	RTCCTLB	70h/71h	0Bh	page 17-16
RTC Status C	RTCSTAC	70h/71h	0Ch	page 17-18
RTC Status D	RTCSTAD	70h/71h	0Dh	page 17-20
General-Purpose CMOS RAM (114 bytes)	RTCCMOS	70h/71h	0E–7Fh	page 17-21

RTC/CMOS RAM Index (RTCIDX)

Direct-Mapped I/O Address 70h

	7	6	5	4	3	2	1	0
Bit	Reserved	CMOSIDX[6–0]						
Reset	x	0	0	0	0	0	0	0
R/W	RSV	W						

Register Description

This register is used to specify the RTC index address to be accessed via the RTCDATA register (see page 17-3).

Bit Definitions

Bit	Name	Function
7	Reserved	**Reserved**
6–0	CMOSIDX[6–0]	**RTC/CMOS RAM Index** This bit field is used to specify the RTC or CMOS RAM index to be read or written via the RTCDATA register (see page 17-3).

Programming Notes

Bit 7 of this register is typically used as the NMI enable bit in PC/AT-compatible systems. In the ÉlanSC520 microcontroller this function has been moved to the NMI_ENB bit in the PICICR register (see page 12-4). See the programmable interrupt controller chapter of the *Élan™SC520 Microcontroller User's Manual*, order #22004, for more information.

RTC/CMOS RAM Data Port (RTCDATA)

Direct-Mapped I/O Address 71h

	7	6	5	4	3	2	1	0
Bit	CMOSDATA[7–0]							
Reset	x	x	x	x	x	x	x	x
R/W	R/W							

Register Description

This register is the RTC data port used to access the RTC index specified in the RTCIDX register (see page 17-2)

Bit Definitions

Bit	Name	Function
7–0	CMOSDATA [7–0]	**RTC/CMOS Data Port** This bit field is used to write to or read from the RTC index specified in the RTCIDX register (see page 17-2).

Programming Notes

RTC Current Second (RTCCURSEC)

I/O Address 70h/71h
RTC Index 00h

	7	6	5	4	3	2	1	0
Bit	SECOND[7–0]							
Reset	x	x	x	x	x	x	x	x
R/W	R/W!							

Register Description

This register is the RTC current second initialization and read-back register.

Bit Definitions

Bit	Name	Function
7–0	SECOND[7–0]	**RTC Current Second** Software initializes the seconds value for the RTC by writing data to this bit field in binary or binary-coded decimal (BCD) format. The seconds component of the RTC time can be read from this bit field. The RTC logic updates this bit field once per second. Valid values for this bit field range from 0 to 59d. If a value greater than 59d is programmed, the bit field value increments up to FFh, wraps around to 0, and only then does the value remain in the valid range.

Programming Notes

Software can suspend updating of this register via the SET bit in the RTCCTLB register (see page 17-16).

Software selects binary or BCD format via the DATE_MODE bit in the RTCCTLB register.

RTC Alarm Second (RTCALMSEC)

I/O Address 70h/71h
RTC Index 01h

	7	6	5	4	3	2	1	0
Bit	ALM_SECOND[7–0]							
Reset	x	x	x	x	x	x	x	x
R/W	R/W!							

Register Description

This register is used to initialize and read back the RTC alarm second.

Bit Definitions

Bit	Name	Function
7–0	ALM_ SECOND[7–0]	**RTC Alarm Second** Software initializes the alarm seconds value for the RTC by writing data to this bit field in either binary or binary-coded decimal (BCD) format. The alarm seconds component of the RTC time can be read from this bit field. Writing any value from C0h to FFh to this bit field makes the seconds component of the alarm a wild card. For example, setting the hours, minutes, and seconds alarm registers to C0h causes an RTC alarm event to be generated once per second. The wild card based once-per-second alarm does not occur unless the hours and minutes alarm settings are also wild cards. The RTC logic checks once per second to see if an alarm has occurred. Valid values for this bit field range from 0 to 59d and all wild card values.

Programming Notes

Software can suspend updating of the RTC via the SET bit in the RTCCTLB register (see page 17-16).

Software selects binary or BCD format via the DATE_MODE bit in the RTCCTLB register.

Software can enable the RTC alarm as an interrupt via the ALM_INT_ENB bit in the RTCCTLB register. The ALM_INT_FLG bit in the RTCSTAC register indicates whether an alarm event has occurred (see page 17-18).

RTC Current Minute (RTCCURMIN)

I/O Address 70h/71h
RTC Index 02h

	7	6	5	4	3	2	1	0
Bit	MINUTE[7–0]							
Reset	x	x	x	x	x	x	x	x
R/W	R/W!							

Register Description

This register is used to initialize and read back the RTC current minute.

Bit Definitions

Bit	Name	Function
7–0	MINUTE[7–0]	**RTC Current Minute** Software initializes the minutes value for the RTC by writing data to this bit field in either binary or binary-coded decimal (BCD) format. The minutes component of the RTC time can be read from this bit field. The RTC logic updates this bit field once per second. Valid values for this bit field range from 0 to 59d. If a value greater than 59d is programmed, the bit field value increments up to FFh, wraps around to 0, and only then does the value remain in the valid range.

Programming Notes

Software can suspend updating of the RTC via the SET bit in the RTCCTLB register (see page 17-16).

Software selects binary or BCD format via the DATE_MODE bit in the RTCCTLB register.

RTC Alarm Minute (RTCALMMIN)

I/O Address 70h/71h
RTC Index 03h

	7	6	5	4	3	2	1	0
Bit	ALM_MINUTE[7–0]							
Reset	x	x	x	x	x	x	x	x
R/W	R/W!							

Register Description

This register used to initialize and read back the RTC alarm minute.

Bit Definitions

Bit	Name	Function
7–0	ALM_ MINUTE[7–0]	**RTC Alarm Minute** Software initializes the alarm minutes value for the RTC by writing data to this bit field in either binary or binary-coded decimal (BCD) format. The alarm minutes component of the RTC time can be read from this bit field. Writing a value of C0–FFh to this bit field makes the minutes component of the alarm a wild card. For example, setting the hours and minutes alarm registers to C0h causes an RTC alarm event to be generated once per minute. The wild card based once-per-minute alarm does not occur unless the hours alarm setting is also a wild card. The RTC logic checks once per second to see if an alarm has occurred. Valid values for this bit field range from 0 to 59d and all wild card values.

Programming Notes

Software can suspend updating of the RTC via the SET bit in the RTCCTLB register (see page 17-16).

Software selects binary or BCD format via the DATE_MODE bit in the RTCCTLB register.

Software can enable the RTC alarm as an interrupt via the ALM_INT_ENB bit in the RTCCTLB register. The ALM_INT_FLG bit in the RTCSTAC register indicates whether an alarm event has occurred (see page 17-18).

RTC Current Hour (RTCCURHR)

I/O Address 70h/71h
RTC Index 04h

	7	6	5	4	3	2	1	0
Bit	AM_PM	HOUR[6–0]						
Reset	x	x	x	x	x	x	x	x
R/W	R/W!	R/W!						

Register Description

This register used to initialize and read back the RTC current hour.

Bit Definitions

Bit	Name	Function
7	AM_PM	**RTC AM/PM Indicator** If the HOUR_MODE_SEL bit is 0 in the RTCCTLB register (see page 17-17), the AM_PM bit is used to indicate whether the current hour is AM or PM. 0 = The current hour is AM (12-hour mode only). In 24-hour mode, always clear this bit to 0. 1 = The current hour is PM (12-hour mode only). The RTC logic updates this bit field once per second.
6–0	HOUR[6–0]	**RTC Current Hour** Software initializes the hours value for the RTC by writing data to this bit field in either binary or binary-coded decimal (BCD) format. The hours component of the RTC time can be read from this bit field. The RTC logic updates this bit field once per second. In 24-hour mode, valid values for this bit field range from 0 to 23d. In 12-hour mode, valid values for this bit field range from 1 to 12d. If a value outside of the valid range is programmed, the register value (including the AM_PM bit) increments up to FFh, wraps around to 0, and only then does the value remain within the valid range.

Programming Notes

Software selects 12-hour or 24-hour mode via the HOUR_MODE_SEL bit in the RTCCTLB register (see page 17-17).

Software can suspend updating of the RTC via the SET bit in the RTCCTLB register.

Software selects binary or BCD format via the DATE_MODE bit in the RTCCTLB register.

RTC Alarm Hour (RTCALMHR)

I/O Address 70h/71h
RTC Index 05h

	7	6	5	4	3	2	1	0
Bit	ALM_ AM_PM	ALM_HOUR[6–0]						
Reset	x	x	x	x	x	x	x	x
R/W	R/W!	R/W!						

Register Description

This register used to initialize and read back the RTC alarm hour.

Bit Definitions

Bit	Name	Function
7	ALM_ AM_PM	**RTC Alarm AM/PM Indicator** If the HOUR_MODE_SEL bit is 0 in the RTCCTLB register (see page 17-17), the ALM_AM_PM bit is used to indicate whether the alarm hour is AM or PM. 0 = The alarm hour is AM (12-hour mode only). In 24-hour mode, always clear this bit to 0 unless programming a wild card (see the ALM_HOUR bit description). 1 = The alarm hour is PM (12-hour mode only). The RTC logic checks this bit field once per second.
6–0	ALM_HOUR [6–0]	**RTC Alarm Hour** Software initializes the alarm hour value for the RTC by writing data to this bit field in either binary or binary-coded decimal (BCD) format. The alarm hours component of the RTC time can be read from this bit field. Writing a value of C0–FFh to this register makes the hours component of the alarm a wild card. For example setting the hours alarm register to C0h causes an RTC alarm event to be generated once per hour. Note that if this register is written with a wild card when 12-hour mode is selected, the ALM_AM_PM bit is a “don’t care” because an alarm occurs every hour regardless. The RTC logic checks once per second to see if an alarm has occurred. In 24-hour mode, valid values for this bit field range from 0 to 23d and all wild card values. In 12-hour mode, valid values for this bit field range from 1 to 12d and all wild card values.

Programming Notes

Software selects 12-hour or 24-hour mode via the HOUR_MODE_SEL bit in the RTCCTLB register (see page 17-17).

Software can suspend updating of the RTC via the SET bit in the RTCCTLB register.

Software selects binary or BCD format via the DATE_MODE bit in the RTCCTLB register.

Software can enable the RTC alarm as an interrupt via the ALM_INT_ENB bit in the RTCCTLB register. The ALM_INT_FLG bit in the RTCSTAC register indicates whether an alarm event has occurred (see page 17-18).

RTC Current Day of the Week (RTCCURDOW)

I/O Address 70h/71h
RTC Index 06h

	7	6	5	4	3	2	1	0
Bit	DAY_OF_WEEK[7–0]							
Reset	x	x	x	x	x	x	x	x
R/W	R/W!							

Register Description

This register used to initialize and read back the RTC current day of the week.

Bit Definitions

Bit	Name	Function
7–0	DAY_OF_ WEEK[7–0]	**RTC Current Day of the Week** Software initializes the day of week value for the RTC by writing data to this bit field in either binary or binary-coded decimal (BCD) formats. The RTC logic updates this bit field once per second. Valid values for this bit field range from 1d to 7d where: 1d = Sunday 2d = Monday 3d = Tuesday 4d = Wednesday 5d = Thursday 6d = Friday 7d = Saturday If a value greater than 7d is programmed, the bit field value increments up to FFh, wraps around to 0, and only then does the value remain within the valid range.

Programming Notes

Software can suspend updating of the RTC via the SET bit in the RTCCTLB register (see page 17-16).

Software selects binary or BCD format via the DATE_MODE bit in the RTCCTLB register.

RTC Current Day of the Month (RTCCURDOM)

I/O Address 70h/71h
RTC Index 07h

	7	6	5	4	3	2	1	0
Bit	DAY_OF_MTH[7–0]							
Reset	x	x	x	x	x	x	x	x
R/W	R/W!							

Register Description

This register used to initialize and read back the RTC current day of the month.

Bit Definitions

Bit	Name	Function
7–0	DAY_OF_MTH [7–0]	**RTC Current Day of the Month** Software initializes the day of month value for the RTC by writing data to this bit field in either binary or binary-coded decimal (BCD) formats. The RTC logic updates this bit field once per second. Valid values for this bit field range from 1 to 31. However, a value in this range is considered invalid if it is inappropriate for the month programmed in the MONTH bit field in the RTCCURMON register (see page 17-12). If a value greater than the number of days in the current month is programmed, the bit field value increments up to FFh, wraps around to 0, and only then does the value remain in the valid range. If the MONTH bit field in the RTCCURMON register is 2d (February), and the YEAR bit field value in the RTCCURYR register (see page 17-13) is a leap year, the DAY_OF_MTH bit field does leap year compensation automatically.

Programming Notes

Software can suspend updating of the RTC via the SET bit in the RTCCTLB register (see page 17-16).

Software selects binary or BCD format via the DATE_MODE bit in the RTCCTLB register.

RTC Current Month (RTCCURMON)

I/O Address 70h/71h
RTC Index 08h

	7	6	5	4	3	2	1	0
Bit	MONTH[7–0]							
Reset	x	x	x	x	x	x	x	x
R/W	R/W!							

Register Description

This register used to initialize and read back the RTC current month

Bit Definitions

Bit	Name	Function
7–0	MONTH[7–0]	**RTC Current Month** Software initializes current month value for the RTC by writing data to this bit field in either binary or binary-coded decimal (BCD) formats. The RTC logic updates this bit field once per second. Valid values for this bit field range from 1d to 12d where: 1d = January 2d = February 3d = March 4d = April 5d = May 6d = June 7d = July 8d = August 9d = September 10d = October 11d = November 12d = December If a value greater than 12d is programmed, the bit field value increments up to FFh, wraps around to 0, and only then does the value remain within the valid range.

Programming Notes

Software can suspend updating of the RTC via the SET bit in the RTCCTLB register (see page 17-16).

Software selects binary or BCD format via the DATE_MODE bit in the RTCCTLB register.

RTC Current Year (RTCCURYR)

I/O Address 70h/71h
RTC Index 09h

	7	6	5	4	3	2	1	0
Bit	YEAR[7–0]							
Reset	x	x	x	x	x	x	x	x
R/W	R/W!							

Register Description

This register used to initialize and read back the RTC current year

Bit Definitions

Bit	Name	Function
7–0	YEAR[7–0]	**RTC Current Year** Software initializes current year value for the RTC by writing data to this bit field in either binary or binary-coded decimal (BCD) formats. The RTC logic updates this bit field once per second. Valid values for this bit field range from 0 to 99d. If a value greater than 99d is programmed, the bit field value increments up to FFh, wraps around to 0, and only then does the value remain within the valid range.

Programming Notes

Software can suspend updating of the RTC via the SET bit in the RTCCTLB register (see page 17-16).

Software selects binary or BCD format via the DATE_MODE bit in the RTCCTLB register.

Refer to the RTC chapter in the *Élan™SC520 Microcontroller User's Manual*, order #22004, for guidelines on storing current century information to address year-2000 issues.

RTC Control A (RTCCTLA)

I/O Address 70h/71h
RTC Index 0Ah

	7	6	5	4	3	2	1	0
Bit	UIP	OSC_CTL[2–0]			RATE_SEL[3–0]			
Reset	x	x	x	x	x	x	x	x
R/W	R!	R/W			R/W			

Register Description

The RTC Control A register is used to determine if an RTC update is in progress, and to control the RTC internal oscillator and periodic interrupt rate.

Bit Definitions

Bit	Name	Function
7	UIP	**Update in Progress** This bit is provided for use by software that needs to modify the time, calendar or alarm registers in the real-time clock. 0 = Software has a guaranteed minimum window of 244 μs in which modifications to the time, calendar, and alarm registers are allowed. While the UIP bit is 0, the time, calendar, and alarm information in RTC RAM is not in transition, and are fully available to software. 1 = The time, calendar, and alarm registers are unavailable for access by software because internal RTC logic is using them. Setting the SET bit in the RTCCTLB register (see page 17-16) inhibits RTC register update cycles and clears the UIP status bit. The UIP bit is a read-only bit and is not affected by reset.
6–4	OSC_CTL[2–0]	**Internal Oscillator Control Bits** 010 = Enable the RTC divider chain to run at the internal time base frequency, which results in one time base update per second. This is the normal operational setting for the OSC_CTL bit. 110 = Hold the RTC divider chain in the reset state. In this mode, the time and date update cycles do not occur. This mode is useful for precision setting of the clock. Time and date update cycles begin 500 milliseconds after the value of 010b is written to this field. 111 = Same as 110b. All other values are reserved. Setting the OSC_CTL value to anything other than 11xb or 010b causes the RTC time base updates to occur at a frequency other than 1Hz. These three bits are not affected by an RTC-only reset, and must be initialized to ensure correct operation.

Bit	Name	Function
3–0	RATE_SEL [3–0]	**Rate Selection** The periodic interrupt output of the RTC is internally tied to the programmable interrupt controller (PIC) and is available for use. The RATE_SEL bit field controls the rate at which periodic interrupts are driven to PIC as follows: 0000 = Periodic interrupt disabled 0001 = 3.906 ms 0010 = 7.812 ms 0011 = 122.070 µs 0100 = 244.141 µs 0101 = 488.281 µs 0110 = 976.563 µs 0111 = 1.953 ms 1000 = 3.906 ms 1001 = 7.812 ms 1010 = 15.625 ms 1011 = 31.250 ms 1100 = 62.500 ms 1101 = 125.000 ms 1110 = 250.000 ms 1111 = 500.000 ms The periodic interrupt is enabled by the PER_INT_ENB bit field in the RTCCTLB register (see page 17-16). The PER_INT_FLG bit in the RTCSTAC register provides latched status for the RTC periodic interrupt event (see page 17-18).

Programming Notes

RTC Control B (RTCCTLB)

I/O Address 70h/71h
RTC Index 0Bh

	7	6	5	4	3	2	1	0
Bit	SET	PER_ INT_ENB	ALM_ INT_ENB	UPD_ INT_ENB	Reserved	DATE_ MODE	HOUR_ MODE_SEL	DS_ENB
Reset	x	0	0	0	0	x	x	x
R/W	R/W	R/W	R/W	R/W	RSV	R/W	R/W	R/W

Register Description

The RTC Control B register is used to temporarily inhibit RTC updates is in progress, to enable RTC interrupts, and to control date encoding, 12/24-hour mode, and daylight savings.

.

Bit Definitions

Bit	Name	Function
7	SET	**Set** 0 = Time and date update cycles are enabled and occur once per second. 1 = Time and date update cycles are disabled, and any update in progress is aborted. The SET bit feature is useful for allowing time and date registers to be updated by software without being disturbed by an automatic update cycle occurring during the change. Neither internal functions nor RTC-only resets affect the SET bit. The SET bit should be set to 1 while changing the DATE_MODE, HOUR_MODE_SEL, or DS_ENB bits; and cleared afterward.
6	PER_ INT_ENB	**Periodic Interrupt Enable** 0 = No RTC periodic interrupt is generated. 1 = The RTC periodic interrupt is enabled. When the PER_INT_FLG bit in the RTCSTAC register transitions from 0 to 1 (see page 17-18), the RTC periodic interrupt latches the INT_FLG bit in the RTCSTAC register to 1 (see page 17-18). If the PER_INT_FLG bit is 1 when the PER_INT_ENB bit is set by software, the INT_FLG bit is asserted immediately. The PER_INT_ENB bit is not modified by any internal RTC functions, but is cleared by an RTC-only reset. The periodic interrupt rate is configured with the RATE_SEL bit field in the RTCCTLA register (see page 17-15). The PER_INT_FLG bit in the RTCSTAC register provides latched status for the RTC periodic event (see page 17-18). Before any RTC interrupt is enabled, the RTCMAP register (see page 12-21) must be configured to route the interrupt to the appropriate interrupt request level and priority.
5	ALM_ INT_ENB	**Alarm Interrupt Enable** 0 = No RTC alarm interrupt is generated. 1 = The RTC alarm interrupt is enabled. When the ALM_INT_FLG bit in the RTCSTAC register transitions from 0 to 1 (see page 17-18), the RTC alarm interrupt latches the INT_FLG bit in the RTCSTAC register to 1 (see page 17-18). If the ALM_INT_FLG bit is 1 when the ALM_INT_ENB bit is set by software, the INT_FLG bit is asserted immediately. The ALM_INT_ENB bit is not modified by any internal RTC functions, but is cleared by an RTC-only reset. The alarm interrupt time is configured with the RTCALMSEC (page 17-5), RTCALMMIN (page 17-7), and RTCALMHR (page 17-9) registers. The ALM_INT_FLG bit in the RTCSTAC register provides latched status for the RTC alarm event (see page 17-18). Before any RTC interrupt is enabled, the RTCMAP register (see page 12-21) must be configured to route the interrupt to the appropriate interrupt request level and priority.

Bit	Name	Function
4	UPD_ INT_ENB	**Update-Ended Interrupt Enable** 0 = No RTC update ended interrupt is generated. 1 = The RTC update ended interrupt is enabled. When the UPD_INT_FLG bit in the RTCSTAC register transitions from 0 to 1 (see page 17-19), the RTC update ended interrupt latches the INT_FLG bit in the RTCSTAC register to 1 (see page 17-18). If the UPD_INT_FLG bit is 1 when the UPD_INT_ENB bit is set by software, the INT_FLG bit is asserted immediately. The UPD_INT_ENB bit is not modified by any internal RTC functions, but is cleared by an RTC-only reset, or by writing the SET bit to 1. The UPD_INT_FLG bit in the RTCSTAC register provides latched status for the RTC update ended event (see page 17-19). Before any RTC interrupt is enabled, the RTCMAP register (see page 12-21) must be configured to route the interrupt to the appropriate interrupt request level and priority.
3	Reserved	**Reserved** This bit field should be written to 0 for normal system operation.
2	DATE_MODE	**Date Mode** The DATE_MODE bit selects whether time and calendar updates are to use binary or binary-coded decimal (BCD) formats. 0 = RTC time and calendar data use BCD encoding. 1 = RTC time and calendar data use binary encoding. Neither internal functions nor RTC-only resets affect this bit. The RTC time and date configuration registers (RTC indexes 0–9h) must be reinitialized after software changes the DATE_MODE bit. Software should set the SET bit to 1 before changing the DATE_MODE bit and then clear the SET bit afterward.
1	HOUR_ MODE_SEL	**12/24-Hour Mode Select** The HOUR_MODE_SEL bit selects whether the hours registers use 12- or 24-hour format. 0 = RTC hours registers use 12-hour format. The AM/PM bit of each hours register represents PM when 1 and AM when 0. 1 = RTC hours registers use 24-hour format. Neither internal functions nor RTC-only resets affect this bit. The RTCCURHR (page 17-8) and RTCALMHR (page 17-9) registers must be reinitialized after software changes the HOUR_MODE_SEL bit. Software should set the SET bit to 1 before changing the HOUR_MODE_SEL bit and then clear the SET bit afterward.
0	DS_ENB	**Daylight Savings Enable** The DS_ENB bit enables special daylight savings time updates. 0 = Special daylight savings time updates do not occur. 1 = Two special time updates occur automatically when this bit is 1. On the first Sunday in April, the time reading that follows 1:59:59 AM is 3:00:00 AM. On the last Sunday in October, the time reading that follows 1:59:59 AM is 1:00:00 AM. Neither internal functions nor RTC-only resets affect this bit. Software should set the SET bit to 1 before changing the DS_ENB bit and then clear the SET bit afterward.

Programming Notes

RTC Status C (RTCSTAC)

I/O Address 70h/71h
RTC Index 0Ch

	7	6	5	4	3	2	1	0
Bit	INT_FLG	PER_ INT_FLG	ALM_ INT_FLG	UPD_ INT_FLG	Reserved			
Reset	0	x	x	x	0	0	0	0
R/W	R!				RSV			

Register Description

The RTC Status C provides RTC interrupt status.

Bit Definitions

Bit	Name	Function
7	INT_FLG	**Interrupt Request Flag** 0 = The RTC interrupt request to the programmable interrupt controller (PIC) is driven inactive. 1 = When this bit transitions from 0 to 1, the RTC interrupt request to the PIC is driven active, which generates a CPU interrupt if the RTC interrupt source is enabled at the PIC. The INT_FLG bit is set to 1 when any one (or more) of the PER_INT_FLG, ALM_INT_FLG, or UPD_INT_FLG bits transition from 0 to 1 while the corresponding enable bit is asserted in the RTCCTLB register (see page 17-16). The INT_FLG bit is also set to 1 if an RTC interrupt source enable bit is written to 1 when the associated flag bit is already asserted. The INT_FLG bit is cleared after read, and is also cleared by an RTC-only reset. If the internal RTC is disabled (via the RTC_DIS bit in the ADDDECCTL register, see page 2-3), the internal signal associated with the INT_FLG status bit is not automatically disconnected from the PIC. If the intent is to use an external RTC to drive the RTC interrupt request, then all internal RTC interrupt enables (bits PER_INT_ENB, ALM_INT_ENB, and UPD_INT_ENB) must be cleared in the RTCCTLB register prior to disabling the internal RTC (see page 17-16).
6	PER_INT_FLG	**Periodic Interrupt Flag** 0 = An RTC periodic event has not occurred since this bit was cleared. This bit is cleared after read and is also cleared by an RTC-only reset. 1 = An RTC periodic event has occurred. This bit is set when an RTC periodic event occurs regardless of the state of its interrupt enable bit (the PER_INT_ENB bit in the RTCCTLB register, page 17-16). The periodic interrupt rate is configured with the RATE_SEL bit field in the RTCCTLA register (see page 17-15).
5	ALM_INT_FLG	**Alarm Interrupt Flag** 0 = An RTC alarm event has not occurred since this bit was cleared. This bit is cleared after read and is also cleared by an RTC-only reset. 1 = An RTC alarm event has occurred. This bit is set when an RTC alarm event occurs regardless of the state of its interrupt enable bit (the ALM_INT_ENB bit in the RTCCTLB register, page 17-16). Alarm events can only occur with a time resolution of one second. An alarm event occurs when the current time contained in the RTCCURSEC, RTCCURMIN, and RTCCURHR registers is equal to the alarm setting configured in the RTCALMSEC, RTCALMMIN, and RTCALMHR registers (see page 17-4 through page 17-9). The alarm time can contain wildcards for hour, minute, or second settings. A wildcard is any value from C0h to FFh.

Bit	Name	Function
4	UPD_INT_FLG	**Update-Ended Interrupt Flag** 0 = An RTC update-ended event has not occurred since this bit was cleared. This bit is cleared after read, and is also cleared by an RTC-only reset. 1 = An RTC update-ended event has occurred. This bit is set upon termination of each time/date update cycle.
3–0	Reserved	**Reserved** This bit field should be written to 0 for normal system operation.

Programming Notes

RTC Status D (RTCSTAD)

I/O Address 70h/71h
RTC Index 0Dh

	7	6	5	4	3	2	1	0
Bit	RTC_VRT	Reserved						
Reset	?	0	0	0	0	0	0	0
R/W	R	RSV						

Register Description

The RTC Status D register provides RTC voltage monitor status.

Bit Definitions

Bit	Name	Function
7	RTC_VRT	**Valid RAM and Time** The Valid RAM and Time (RTC_VRT) bit is used to determine the validity of the RTC time, date and CMOS RAM registers. 0 = Indicates that the RTC backup battery, as sensed by the microcontroller's BBATSEN pin, was below a fixed reference voltage prior to the application of main system power. (See the *Élan™SC520 Microcontroller Data Sheet*, order #22003, for the RTC voltage monitor reference voltage.) Because the RTC reference voltage is too low to keep the RTC logic operational, the RTC date, time, and CMOS RAM are invalid, and the microcontroller performed an RTC-only reset as a result. An RTC-only reset clears this latched status bit. 1 = Reading this bit causes it to be set to 1, and it remains set until an RTC-only reset occurs. Note that this bit is not a real-time indication of the state of the external RTC backup battery, although there can be a relationship. For example, if the RTC backup battery is removed while main system power is still applied, this bit still reads back 1 until main system power is cycled. This bit is always set to 0 upon RTC-only reset. This bit is always set to 1 after an initial read to this register is performed. This bit remains set to 1 until an RTC-only reset occurs. RTC-only reset occurs any time the BBATSEN input is sampled to be below the RTC reference voltage during a power-on reset.
6–0	Reserved	**Reserved** This bit field should be written to 0 for normal system operation.

Programming Notes

The default value for this register (RTCSTAD) depends on whether an RTC-only reset has occurred. The RTC-only reset occurs when the BBATSEN input is sampled to be below the RTC reference voltage prior to a power-on system reset. See the *Élan™SC520 Microcontroller Data Sheet*, order #22003, for the RTC voltage monitor reference voltage.

General-Purpose CMOS RAM (114 bytes) (RTCCMOS)

I/O Address 70h/71h
RTC Indexes 0E–7Fh

	7	6	5	4	3	2	1	0
Bit	RTC_CMOS_REG_X[7–0]							
Reset	x	x	x	x	x	x	x	x
R/W	R/W							

Register Description

These registers are the general-purpose CMOS RAM registers.

Bit Definitions

Bit	Name	Function
7–0	RTC_CMOS_ REG_X[7–0]	**CMOS RAM Location** These are 114 bytes of general-purpose, battery-backed (nonvolatile) CMOS RAM available for use by system software, applications, etc. In a PC/AT-compatible system, many of these bytes can be used by the system BIOS. The number of bytes used and the meaning of data stored in a given CMOS RAM byte location can vary between different BIOS vendors or even between different versions of a single BIOS. Accesses to CMOS RAM locations can be performed without any regard for RTC operations. For example, DATE_MODE bit has no effect on CMOS RAM data. If the RTC is disabled (via the RTC_DIS bit in the ADDDECCTL register, see page 2-3), the CMOS RAM is unavailable, but not lost (unless both main and backup power to the internal RTC is removed). Re-enabling the RTC allows access to the CMOS RAM with its contents intact.

Programming Notes

CHAPTER

18 UART SERIAL PORT REGISTERS

18.1 OVERVIEW

This chapter describes the universal asynchronous receiver/transmitter (UART) registers of the ÉlanSC520 microcontroller.

The ÉlanSC520 microcontroller includes two industry-standard 16550-compatible UARTs.

The UART register set includes two groups of registers:

- Six memory-mapped configuration region (MMCR) registers are used to configure and control UART functions specific to the ÉlanSC520 microcontroller.
- 24 direct-mapped I/O registers are used for industry-standard UART configuration, control, and status functions.

See the *Élan™SC520 Microcontroller User's Manual*, order #22004, for details about the UARTs.

Table 18-1 and Table 18-2 list each type of UART register in offset order, with the corresponding description's page number.

18.2 REGISTERS

Table 18-1 UART MMCR Registers

Register Name	Mnemonic	MMCR Offset	Page Number
UART 1 General Control	UART1CTL	CC0h	page 18-3
UART 1 General Status	UART1STA	CC1h	page 18-4
UART 1 FIFO Control Shadow	UART1FCRSHAD	CC2h	page 18-5
UART 2 General Control	UART2CTL	CC4h	page 18-3
UART 2 General Status	UART2STA	CC5h	page 18-4
UART 2 FIFO Control Shadow	UART2FCRSHAD	CC6h	page 18-5

Table 18-2 UART Direct-Mapped Registers

Register Name	Mnemonic	I/O Address	Page Number
UART 2 Transmit Holding	UART2THR	02F8h	page 18-7
UART 2 Receive Buffer	UART2RBR	02F8h	page 18-8
UART 2 Baud Clock Divisor Latch LSB	UART2BCDL	02F8h	page 18-9
UART 2 Baud Clock Divisor Latch MSB	UART2BCDH	02F9h	page 18-10
UART 2 Interrupt Enable	UART2INTENB	02F9h	page 18-11
UART 2 Interrupt ID	UART2INTID	02FAh	page 18-12
UART 2 FIFO Control	UART2FCR	02FAh	page 18-15
UART 2 Line Control	UART2LCR	02FBh	page 18-17
UART 2 Modem Control	UART2MCR	02FCh	page 18-19
UART 2 Line Status	UART2LSR	02FDh	page 18-21
UART 2 Modem Status	UART2MSR	02FEh	page 18-23

Table 18-2 UART Direct-Mapped Registers (Continued)

Register Name	Mnemonic	I/O Address	Page Number
UART 2 Scratch Pad	UART2SCRATCH	02FFh	page 18-25
UART 1 Transmit Holding	UART1THR	03F8h	page 18-7
UART 1 Receive Buffer	UART1RBR	03F8h	page 18-8
UART 1 Baud Clock Divisor Latch LSB	UART1BCDL	03F8h	page 18-9
UART 1 Baud Clock Divisor Latch MSB	UART1BCDH	03F9h	page 18-10
UART 1 Interrupt Enable	UART1INTENB	03F9h	page 18-11
UART 1 Interrupt ID	UART1INTID	03FAh	page 18-12
UART 1 FIFO Control	UART1FCR	03FAh	page 18-15
UART 1 Line Control	UART1LCR	03FBh	page 18-17
UART 1 Modem Control	UART1MCR	03FCh	page 18-19
UART 1 Line Status	UART1LSR	03FDh	page 18-21
UART 1 Modem Status	UART1MSR	03FEh	page 18-23
UART 1 Scratch Pad	UART1SCRATCH	03FFh	page 18-25

Memory-Mapped

UART 1 General Control (UART1CTL) — MMCR Offset CC0h

UART 2 General Control (UART2CTL) — MMCR Offset CC4h

	7	6	5	4	3	2	1	0
Bit	Reserved					CLK_SRC	RXTC_ENB	TXTC_ENB
Reset	0	0	0	0	0	0	0	0
R/W	RSV					R/W	R/W	R/W

Register Description

This register is used to enable or disable the transmit and receive transfer count (TC) interrupt and also select from two internal baud-rate clock sources.

Bit Definitions

Bit	Name	Function
7–3	Reserved	**Reserved** This bit field should be written to 0 for normal system operation.
2	CLK_SRC	**UART x Clock Source Enable** 0 = 18.432 MHz 1 = 1.8432 MHz
1	RXTC_ENB	**UART x Receive TC Interrupt Enable** 0 = Receive TC interrupt disable 1 = Receive TC interrupt enable
0	TXTC_ENB	**UART x Transmit TC Interrupt Enable** 0 = Transmit TC interrupt disable 1 = Transmit TC interrupt enable

Programming Notes

Each UART can generate an interrupt when the transfer count (GPTC) signal associated with DMA transfers is asserted. Table 18-5 on page 18-14 provides a summary of UART interrupt sources.

Memory-Mapped

UART 1 General Status (UART1STA) — MMCR Offset CC1h

UART 2 General Status (UART2STA) — MMCR Offset CC5h

	7	6	5	4	3	2	1	0
Bit	Reserved						RXTC_DET	TXTC_DET
Reset	0	0	0	0	0	0	0	0
R/W	RSV						R/W!	R/W!

Register Description

This register shows the status of transfer count (TC) interrupt detected for the UART. These bits are cleared by writing a 1 to them.

Bit Definitions

Bit	Name	Function
7–2	Reserved	**Reserved** This bit field should be written to 0 for normal system operation.
1	RXTC_DET	**UART x Receive TC Detected** When reading: 0 = TC not detected 1 = TC detected This bit is cleared by writing a 1.
0	TXTC_DET	**UART x Transmit TC Detected** When reading: 0 = TC not detected 1 = TC detected This bit is cleared by writing a 1.

Programming Notes

Each UART can generate an interrupt when the transfer count (GPTC) signal associated with DMA transfers is asserted. Table 18-5 on page 18-14 provides a summary of UART interrupt sources.

Memory-Mapped

UART 1 FIFO Control Shadow (UART1FCRSHAD) — MMCR Offset CC2h

UART 2 FIFO Control Shadow (UART2FCRSHAD) — MMCR Offset CC6h

	7	6	5	4	3	2	1	0
Bit	RFRT[1–0]		Reserved		DMA_ MODE	TF_CLR	RF_CLR	FIFO_ENB
Reset	0	0	0	0	0	0	0	0
R/W	R		RSV		R	R	R	R

Register Description

This register is provided for reading the information written to the UART x FIFO Control (UARTxFCR) register (see page 18-15), because the UARTxFCR register is write-only. Writing to this shadow register has no effect.

Bit Definitions

Bit	Name	Function
7–6	RFRT[1–0]	**Receiver FIFO Register Trigger Bits** In 16550-compatible mode, this bit field specifies the trigger level at which the INT_ID bit field in the UARTxINTID register (see page 18-13) reports that a received data available interrupt is pending. If received data available interrupts are enabled in the UARTxINTENB register (page 18-11), the system is interrupted when the receive FIFO fills to the trigger level as follows: 00 = 1 byte 01 = 4 bytes 10 = 8 bytes 11 = 14 bytes When the data in the receive FIFO falls below the specified trigger level, the interrupt is cleared.
5–4	Reserved	**Reserved** This bit field should be written to 0 for normal system operation.
3	DMA_MODE	**DMA Mode** This bit is valid only in 16550-compatible mode. In 16450-compatible mode, the DMA operation is defined as if this bit were set to 0. 0 = The internal rxdrq signal to the DMA controller goes High when there is at least one character in the receiver FIFO or the UARTx Receive Buffer register (see page 18-8). The internal txdrq signal goes High when the transmitter FIFO (16550-compatible mode) or the UARTx Transmit Holding register (page 18-7) (16450-compatible mode) is not full. 1 = The internal rxdrq signal goes High when the trigger level or the time-out has been reached, and then it goes inactive when there are no more characters in the FIFO or holding register. For transmit, the internal txdrq signal goes High when the transmitter FIFO is not full and remains High until the transmitter FIFO is completely full.
2	TF_CLR	**Transmitter FIFO Clear** Because the direct-mapped version of this bit is self-clearing, it always reads back 0.
1	RF_CLR	**Receiver FIFO Clear** Because the direct-mapped version of this bit is self-clearing, it always reads back 0.

Bit	Name	Function
0	FIFO_ENB	**FIFO Enabled (16550-Compatible Mode Enabled)** 0 = The UART is in 16450-compatible mode. Accesses to receive and transmit FIFOs and to all FIFO control bits (except FIFO_ENB) in the write-only UART x FIFO Control (UARTxFCR) register are disabled (see page 18-15). 1 = The UART is in 16550-compatible mode. Accesses to receive and transmit FIFOs and to all FIFO control bits in the write-only UARTxFCR register are enabled. This bit must be 1 when other UARTxFCR register bits are written to or they cannot be programmed. *Any* mode switch clears both FIFOs.

Programming Notes

UARTxFCRSHAD is a shadow register for the write-only UART x FIFO Control (UARTxFCR) register (see page 18-15).

Direct-Mapped

UART 2 Transmit Holding (UART2THR) — I/O Address 02F8h

UART 1 Transmit Holding (UART1THR) — I/O Address 03F8h

	7	6	5	4	3	2	1	0
Bit	THR[7–0]							
Reset	0	0	0	0	0	0	0	0
R/W	W!							

Register Description

This is a write-only register used to write data to be transmitted. This register can be accessed only when the DLAB bit is 0 in the UARTxLCR register (see page 18-17).

Bit Definitions

Bit	Name	Function
7–0	THR[7–0]	**UART x Transmit Holding Register** When the DLAB bit is 0 in the UART x Line Control (UARTxLCR) register (see page 18-17) and the THRE bit is 1 in the UART x Line Status (UARTxLSR) register (see page 18-21), writing a byte to this bit field causes the byte to be transmitted.

Programming Notes

When the DLAB bit is 0 in the UARTxLCR register (see page 18-17), reads from this address access the UART x Receive Buffer (UARTxRBR) register (see page 18-8), and writes to this address access the UART x Transmit Holding (UARTxTHR) register.

When the DLAB bit is 1 in the UARTxLCR register, reads from and writes to this address access the UART x Baud Clock Divisor Latch LSB (UARTxBCDL) register (see page 18-9).

Direct-Mapped

UART 2 Receive Buffer (UART2RBR) — I/O Address 02F8h

UART 1 Receive Buffer (UART1RBR) — I/O Address 03F8h

	7	6	5	4	3	2	1	0
Bit	RBR[7–0]							
Reset	0	0	0	0	0	0	0	0
R/W	R!							

Register Description

This is a read-only register used to read received data. This register can be accessed only when the DLAB bit is 0 in the UARTxLCR register (see page 18-17).

Bit Definitions

Bit	Name	Function
7–0	RBR[7–0]	**UART x Receive Buffer** When the DLAB bit is 0 in the UART x Line Control (UARTxLCR) register (see page 18-17) and the DR bit is 1 in the UART x Line Status (UARTxLSR) register (see page 18-21), this bit field contains valid data received over the serial line. Reading this bit field returns the last byte received (in 16450-compatible mode) or the head of the receive FIFO (in 16550-compatible mode).

Programming Notes

When the DLAB bit is 0 in the UARTxLCR register (see page 18-17), reads from this address access the UART x Receive Buffer (UARTxRBR) register, and writes to this address access the UART x Transmit Holding (UARTxTHR) register (see page 18-7).

When the DLAB bit is 1 in the UARTxLCR register, reads from and writes to this address access the UART x Baud Clock Divisor Latch LSB (UARTxBCDL) register (see page 18-9).

Direct-Mapped

UART 2 Baud Clock Divisor Latch LSB (UART2BCDL) — I/O Address 02F8h

UART 1 Baud Clock Divisor Latch LSB (UART1BCDL) — I/O Address 03F8h

	7	6	5	4	3	2	1	0
Bit	DIV[7–0]							
Reset	0	0	0	0	0	0	0	0
R/W	R/W							

Register Description

This register contains bits DIV[7–0] of the16-bit baud-rate clock divisor used to generate the 16x baud clock. The baud-rate clock divisor can only be accessed when the DLAB bit is 1 in the UARTxLCR register (see page 18-17). The DIV[15–8] bits are located in the UARTxBCDH register (see page 18-10).

Bit Definitions

Bit	Name	Function
7–0	DIV[7–0]	**UART x Baud Clock Divisor Latch** When the DLAB bit is 1 in the UARTxLCR register (see page 18-17), this bit field holds the *least* significant byte of a 16-bit baud-rate clock divisor that is used to generate the 16x baud clock.

Programming Notes

Setting the DIV[15–0] bit field to 0000h is not supported.

When the DLAB bit is 0 in the UARTxLCR register (see page 18-17), reads from this address access the UART x Receive Buffer (UARTxRBR) register (see page 18-8), and writes to this address access the UART x Transmit Holding (UARTxTHR) register (see page 18-7).

When the DLAB bit is 1 in the UARTxLCR register, reads from and writes to this address access the UART x Baud Clock Divisor Latch LSB (UARTxBCDL) register.

The clock source frequency is selected by the CLK_SRC bit in the UARTxCTL register (see page 18-3). Table 18-3 lists the divisor value (in decimal and hexadecimal) to use with each clock frequency to achieve common baud rates.

Table 18-3 Baud Rates, Divisors, and Clock Source

	DIV[15–0] (Decimal)		DIV[15–0] (Hexadecimal)	
Baud Rate	**1.8432 MHz**	**18.432 MHz**	**1.8432 MHz**	**18.432 MHz**
300 baud	384d	3840d	0180h	0F00h
600 baud	192d	1920d	00C0h	0780h
2400 baud	48d	480d	0030h	01E0h
4800 baud	24d	240d	0018h	00F0h
7200 baud	16d	160d	000Fh	00A0h
9600 baud	12d	120d	000Ch	0078h
14.4 kbaud	8d	80d	0008h	0050h
19.2 kbaud	6d	60d	0006h	003Ch
57.6 kbaud	2d	20d	0002h	0014h
115.2 kbaud	1d	10d	0001h	000Ah
144 kbaud		8d		0008h
192 kbaud		6d		0006h
288 kbaud		4d		0004h
576 kbaud		2d		0002h
1.152 Mbaud		1d		0001h

Direct-Mapped

UART 2 Baud Clock Divisor Latch MSB (UART2BCDH) I/O Address 02F9h

UART 1 Baud Clock Divisor Latch MSB (UART1BCDH) I/O Address 03F9h

	7	6	5	4	3	2	1	0
Bit	DIV[15–8]							
Reset	0	0	0	0	0	0	0	0
R/W	R/W							

Register Description

This register contains bits DIV[15–8] of the16-bit baud-rate clock divisor used to generate the 16x baud clock. The baud-rate clock divisor can only be accessed when the DLAB bit is 1 in the UARTxLCR register (see page 18-17). The DIV[7–0] bits are located in the UARTxBCDL register (see page 18-9).

Bit Definitions

Bit	Name	Function
7–0	DIV[15–8]	**UART x Baud Clock Divisor Latch** When the DLAB bit is 1 in the UARTxLCR register (see page 18-17), this bit field holds the *most* significant byte of the 16-bit baud-rate clock divisor that is used to generate the 16x baud clock.

Programming Notes

Setting the DIV[15–0] bit field to 0000h in the UARTxBCDL and UARTxBCDH registers is not supported.

When the DLAB bit is 0 in the UARTxLCR register (see page 18-17), reads from and writes to this address access the UART x Interrupt Enable (UARTxINTENB) register (see page 18-11).

When the DLAB bit is 1 in the UARTxLCR register, reads from and writes to this address access the UART x Baud Clock Divisor Latch MSB (UARTxBCDH) register.

The clock source frequency is selected by the CLK_SRC bit in the UARTxCTL register (see page 18-3). Table 18-3 on page 18-9 lists the divisor value (in decimal and hexadecimal) to use with each clock frequency to achieve common baud rates.

Direct-Mapped

UART 2 Interrupt Enable (UART2INTENB) — I/O Address 02F9h

UART 1 Interrupt Enable (UART1INTENB) — I/O Address 03F9h

	7	6	5	4	3	2	1	0
Bit	Reserved				EMSI	ERLSI	ETHREI	ERDAI
Reset	0	0	0	0	0	0	0	0
R/W	RSV				R/W	R/W	R/W	R/W

Register Description

This register enables the following serial port interrupts: modem status, receiver line status, transmitter holding register empty, received data available, and time-out interrupt. This register can be accessed only when the DLAB bit is 0 in the UARTxLCR register (see page 18-17).

Each interrupt can individually activate the interrupt request signal.

Bit Definitions

Bit	Name	Function
7–4	Reserved	**Reserved** This bit field should be written to 0 for normal system operation.
3	EMSI	**Enable Modem Status Interrupt** 0 = Disable modem status interrupt. 1 = Enable modem status interrupt.
2	ERLSI	**Enable Receiver Line Status Interrupt** 0 = Disable receiver line status interrupt. 1 = Enable receiver line status interrupt.
1	ETHREI	**Enable Transmitter Holding Register Empty Interrupt** 0 = Disable transmitter holding register empty interrupt. 1 = Enable transmitter holding register empty interrupt.
0	ERDAI	**Enable Received Data Available Interrupt** 0 = Disable data available interrupt in 16450-compatible mode; in 16550-compatible mode, disable FIFO trigger level reached interrupt and time-out interrupt. 1 = Enable received data available interrupt in 16450-compatible mode; in 16550-compatible mode, this bit also enables FIFO trigger level reached interrupt and time-out interrupt. More detail on time-out interrupts can be found in the UARTxINTID register description (page 18-12).

Programming Notes

Table 18-5 on page 18-14 provides a summary of UART interrupt sources.

When the DLAB bit is 0 in the UARTxLCR register (see page 18-17), reads from and writes to this address access the UART x Interrupt Enable (UARTxINTENB) register.

When the DLAB bit is 1 in the UARTxLCR register, reads from and writes to this address access the UART x Baud Clock Divisor Latch MSB (UARTxBCDH) register (see page 18-10).

Direct-Mapped

UART 2 Interrupt ID (UART2INTID) — I/O Address 02FAh

UART 1 Interrupt ID (UART1INTID) — I/O Address 03FAh

	7	6	5	4	3	2	1	0
Bit	FIFO_MODE[1–0]		Reserved		INT_ID[2–0]			INT_NOT_PND
Reset	0	0	0	0	0	0	0	1
R/W	R	R	RSV		R	R	R	R

Register Description

This read-only register is used to identify UART interrupts and the current FIFO mode. Writes to this address access the UART x FIFO Control (UARTxFCR) register (see page 18-15).

Bit Definitions

Bit	Name	Function
7–6	FIFO_MODE [1–0]	**FIFO Mode Indication** FIFO is only present when 16550-compatible mode is enabled: 00 = 16450-compatible mode is enabled 01 = No significance 10 = No significance 11 = 16550-compatible mode is enabled The FIFO mode is selected by the FIFO_ENB bit in the UARTxFCR register (see page 18-16).
5–4	Reserved	**Reserved** These bits always read back 00b.

Bit	Name	Function
3–1	INT_ID[2–0]	**Interrupt Identification Bit Field** The INT_ID bit field indicates only the highest priority interrupt (as defined in Table 18-4) when two interrupt sources are pending simultaneously. When the interrupt source is cleared, a subsequent read from the INT_ID bit field returns the next highest priority interrupt source. The INT_ID bit field has no meaning if the INT_NOT_PND bit is 1.

Table 18-4 UART Interrupt Identification and Priority

INT_ID Bit Field	Description	Identification Priority
000b	Modem status	Fourth (Lowest)
001b	Transmit holding register empty (16540-compatible mode)/Transmit FIFO empty (16550-compatible mode)	Third
010b	Received data available (16540-compatible mode)/ Receiver FIFO trigger (16550-compatible mode)	Second
011b	Receive line status	First (Highest)
100b	Not used	—
101b	Not used	—
110b	FIFO time-out	Second
111b	Not used	—

In 16450-compatible mode, the INT_ID[2] bit always reads back 0.

A receiver FIFO trigger occurs when the data in the receive FIFO fills to the level set in the RFRT bit field of the UARTxFCR register (see page 18-15).

A FIFO time-out occurs when the receive FIFO is not empty, and more than four continuous character times have elapsed without more data being placed into or read out of the receive FIFO. Reading a character from the receive FIFO clears the time-out interrupt.

See the UARTxLSR and UARTxMSR register descriptions on page 18-21 and page 18-23 for information about the other interrupt events identified by the INT_ID bit field.

Bit	Name	Function
0	INT_NOT_PND	**No Serial Port Interrupt Pending** 0 = Interrupt pending 1 = No Interrupt pending

Programming Notes

Table 18-5 on page 18-14 provides a summary of UART interrupt sources.

Interrupts generated by the UART are cleared in a variety of ways, depending on the source event. For details about clearing a particular event, see the event's status bit description. Table 18-5 on page 18-14 lists interrupt status registers and bits. For additional information, see the *Élan™SC520 Microcontroller User's Manual*, order #22004.

Table 18-5 UART Interrupt Programming Summary

Interrupt Description	Enable Register[1, 2]	Status Register[3]	Source Event	Polled Status Bit
Receive DMA transfer count	UARTxCTL, page 18-3	UARTxSTA, page 18-4	UART x Receive TC Detected	RXTC_DET
Transmit DMA transfer count			UART x Transmit TC Detected	TXTC_DET
Modem status change	UARTxINTENB, page 18-11	UARTxMSR, page 18-23	Delta data carrier detect	DDCD
			Trailing edge ring indicator	TERI
			Delta data set ready	DDSR
			Delta clear to send	DCTS
Receiver line status		UARTxLSR, page 18-21	Break indicator	BI
			Framing error	FE
			Parity error	PE
			Overrun error	OE
Transmitter holding register empty			Transmit holding register (16450-compatible mode) or transmitter FIFO (16550-compatible mode) empty	THRE
Received data available			Data ready (16450-compatible mode)	DR
		—[4]	FIFO trigger level reached (16550-compatible mode)	—
FIFO time-out[5]			FIFO time-out (16550-compatible mode)	—

Notes:

1. Before any UART interrupt is enabled, the corresponding UARTxMAP register (see page 12-21) must be configured to route the interrupt to the appropriate interrupt request level and priority.

2. The OUT2 bit in the UARTxMCR register (page 18-19) is used as a master control for UART interrupts. The OUT2 bit must be set for UART interrupts to be generated. Status bits can be read even when interrupts are disabled.

3. If two of the interrupts enabled in the UARTxINTENB register are pending simultaneously, the highest-priority interrupt is identified in the INT_ID bit field of the UARTxINTID register (see page 18-13).

4. There are no polled-status bits for the FIFO trigger level and FIFO time-out events. These events are indicated by the INT_ID bit field only (see page 18-13).

5. The FIFO time-out interrupt is enabled with the received data available interrupt by the ERDAI bit in the UARTxINTENB register (see page 18-11).

Direct-Mapped

UART 2 FIFO Control (UART2FCR) — I/O Address 02FAh

UART 1 FIFO Control (UART1FCR) — I/O Address 03FAh

	7	6	5	4	3	2	1	0
Bit	RFRT[1–0]		Reserved		DMA_ MODE	TF_CLR	RF_CLR	FIFO_ENB
Reset	0	0	0	0	0	0	0	0
R/W	W	W	RSV		W	W	W	W

Register Description

This is a write-only register used to enable and control the FIFO in 16550-compatible mode. Reads to this address access the UART x Interrupt ID (UARTxINTID) register (see page 18-12).

Bit Definitions

Bit	Name	Function
7–6	RFRT[1–0]	**Receiver FIFO Register Trigger Bits** In 16550-compatible mode, this bit field specifies the trigger level at which the INT_ID bit field in the UARTxINTID register (see page 18-13) reports that a received data available interrupt is pending. If received data available interrupts are enabled in the UARTxINTENB register (page 18-11), the system is interrupted when the receive FIFO fills to the trigger level as follows: 00 = 1 byte 01 = 4 bytes 10 = 8 bytes 11 = 14 bytes When the data in the receive FIFO falls below the specified trigger level, the interrupt is cleared.
5–4	Reserved	**Reserved** These bits should be written to 0 for normal system operation.
3	DMA_MODE	**DMA Mode** This bit is valid only in 16550-compatible mode. In 16450-compatible mode, the DMA operation is defined as if this bit were set to 0. 0 = The internal rxdrq signal to the DMA controller goes High when there is at least one character in the receiver FIFO or the UARTx Receive Buffer register (see page 18-8). The txdrq signal goes High when the transmitter FIFO (16550-compatible mode) or the UARTx Transmit Holding register (page 18-7) (16450-compatible mode) is not full. 1 = The internal rxdrq signal goes High when the trigger level or the time-out has been reached, and then it goes inactive when there are no more characters in the FIFO or holding register. For transmit, the txdrq signal goes High when the transmitter FIFO is not full and remains High until the transmitter FIFO is completely full.
2	TF_CLR	**Transmitter FIFO Clear** 0 = Writing a 0 to this bit has no effect. This bit is self-clearing and does not need to be reset by software. 1 = Writing a 1 to this bit position clears the transmit FIFO and resets the transmit FIFO counter logic. It does not clear the transmitter shift register.
1	RF_CLR	**Receiver FIFO Clear** 0 = Writing a 0 to this bit has no effect. This bit is self-clearing and does not need to be reset by software. 1= Writing a 1 to this bit position clears the receive FIFO and resets the receive FIFO counter logic. It does not clear the receive shift register.

Bit	Name	Function
0	FIFO_ENB	**FIFO Enable** 0 = Causes the UART to enter 16450-compatible mode. Disables accesses to receive and transmit FIFOs and all FIFO control bits, except this bit. 1 = Causes the UART to enter 16550-compatible mode. Enables receive and transmit FIFO buffers, and enables accesses to other FIFO control bits.

Programming Notes

The contents of this write-only register can be read back via the UART x FIFO Control Shadow (UARTxFCRSHAD) register (see page 18-5).

Direct-Mapped

UART 2 Line Control (UART2LCR) — I/O Address 02FBh

UART 1 Line Control (UART1LCR) — I/O Address 03FBh

	7	6	5	4	3	2	1	0
Bit	DLAB	SB	SP	EPS	PENB	STP	WLS[1–0]	
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

Register Description

This register is used to configure the format of the UART frame for data transfer, including character length, stop bits, and parity. The DLAB bit is used to gain access to the baud-rate divisor latches or the UARTxTHR and UARTxRBR registers.

Bit Definitions

Bit	Name	Function
7	DLAB	**Divisor Latch Access** 0 = Software can access the transmit holding, receive buffer, and interrupt enable registers: UARTxTHR (page 18-7), UARTxRBR (page 18-8), and UARTxINTENB (page 18-11). 1 = Software can access the baud rate divisor latch registers: UARTxBCDL (page 18-9) and UARTxBCDH (page 18-10).
6	SB	**Set Break Enable** Setting this bit causes a break condition to be transmitted to the receiving UART. 0 = Disable set break. 1 = Force serial output to spacing state (logic 0) regardless of other transmitter activity. The break control acts on the SOUTx pin only and has no other effect on the transmitter logic.
5	SP	**Stick Parity Enable** Stick parity forces the parity bit to be always 0 or 1. 0 = Stick Parity is disabled. If parity is enabled (by the PENB bit), normal parity is used: the parity bit dynamically changes so the number of 1 bits in the transmitted data is always odd or even (depending on the EPS bit). 1 = Stick Parity is enabled. If parity is enabled, the parity bit is always 0 or 1. If bits SP, EPS, and PENB are 1, the parity bit is generated and checked as 0. If bits SP and PENB are 1 and EPS is 0, the parity bit is generated and checked as 1.
4	EPS	**Even Parity Select** Parity must be enabled via the PENB bit for this bit to have meaning: 0 = Odd parity. The parity bit is manipulated to force an odd number of 1 bits in the transmitted data and the same condition is checked for in the received data. 1 = Even parity. The parity bit is manipulated to force an even number of 1 bits in the transmitted data and the same condition is checked for in the received data. Start and stop bits are not included in the parity generation and checking scheme.
3	PENB	**Parity Enable** 0 = Parity is disabled. 1 = Parity is enabled. A parity bit is generated in the transmitted data and checked in the received data. The parity bit is located between the last data-word bit and the first stop bit in the bit stream.

Bit	Name	Function
2	STP	**Stop Bits** This bit sets the number of stop bits used, based on the character length set in the WSL bit field: If WSL = 00b (5-bit words): 0 = 1 stop bit 1 = 1.5 stop bits If WSL = 01–11b (6-, 7-, or 8-bit words): 0 = 1 stop bit 1 = 2 stop bits
1–0	WLS[1–0]	**Transmit/Receive Word Length Select** This bit field sets the UART data word length. 00 = 5 bits 01 = 6 bits 10 = 7 bits 11 = 8 bits

Programming Notes

Direct-Mapped

UART 2 Modem Control (UART2MCR) — I/O Address 02FCh

UART 1 Modem Control (UART1MCR) — I/O Address 03FCh

	7	6	5	4	3	2	1	0
Bit	Reserved			LOOP	OUT2	OUT1	RTS	DTR
Reset	0	0	0	0	0	0	0	0
R/W	RSV			R/W	R/W	R/W	R/W	R/W

Register Description

This register is used to control the interface with the modem or peripheral device. It is used to enable interrupts from the UART, enable loopback diagnostic mode, or assert $\overline{\text{RTSx}}$ or $\overline{\text{DTRx}}$.

Bit Definitions

Bit	Name	Function
7–5	Reserved	**Reserved** This bit field should be written to 0 for normal system operation.
4	LOOP	**Loopback Mode (Diagnostic Mode) Enable** 0 = Loopback mode is disabled 1 = Loopback mode is enabled The following internal connections are made by setting this diagnostic bit: $\overline{\text{RTSx}}$ is internally connected to $\overline{\text{CTSx}}$. $\overline{\text{DTRx}}$ is internally connected to $\overline{\text{DSRx}}$. $\overline{\text{out1}}$ is internally connected to $\overline{\text{RINx}}$. $\overline{\text{out2}}$ is internally connected to $\overline{\text{DCDx}}$. Also, the SOUTx pin is driven High, the SIN input line is blocked, and interrupt generation to the PIC is disabled. The transmit shift register is directly connected to the receive shift register. In addition, the $\overline{\text{DTRx}}$ and $\overline{\text{RTSx}}$ signals, and the internal $\overline{\text{out1}}$ and $\overline{\text{out2}}$ signals are forced inactive. Modem status events (UARTxMSR register, see page 18-23) can be forced by setting the EMSI bit in the UARTxINTENB register (see page 18-11), and changing one of the bits OUT2, OUT1, RTS, or DTR in this register (UARTxMCR) in loopback mode.
3	OUT2	**Enable UART x Interrupts** This bit controls the internal $\overline{\text{out2}}$ signal, which is used internally as a master enable for UART x interrupts when loopback mode is disabled. 0 = No UART x interrupt requests are sensed at the Programmable Interrupt Controller (PIC). 1 = UART x interrupt requests are enabled. In loopback mode, the internal $\overline{\text{out2}}$ signal is internally connected to the $\overline{\text{DCDx}}$ signal, which can be read via the DCD bit in the UARTxMSR register (see page 18-23). 0 = In loopback mode, the $\overline{\text{out2}}$ signal forces $\overline{\text{DCDx}}$ High (deasserted). 1 = In loopback mode, the $\overline{\text{out2}}$ signal forces $\overline{\text{DCDx}}$ Low (asserted).
2	OUT1	**$\overline{\text{out1}}$ Control** This bit controls the internal $\overline{\text{out1}}$ signal, which is not used when loopback mode is disabled. It is provided for PC/AT compatibility and for use as part of the loopback diagnostics. 0 = In loopback mode, the $\overline{\text{out1}}$ signal forces $\overline{\text{RINx}}$ High (deasserted). 1 = In loopback mode, the $\overline{\text{out1}}$ signal forces $\overline{\text{RINx}}$ Low (asserted). In loopback mode, the internal $\overline{\text{out1}}$ signal is internally connected to the $\overline{\text{RINx}}$ signal, which can be read via the RIN bit in the UARTxMSR register (see page 18-23). Other than that, the OUT1 bit has no effect on system operation and can be used as a scratch pad during normal system operation.

Bit	Name	Function
1	RTS	**Request To Send** In normal operation (loopback mode disabled), this bit is the complement of the $\overline{\text{RTSx}}$ signal. 0 = $\overline{\text{RTSx}}$ is forced High, deasserting the signal. 1 = $\overline{\text{RTSx}}$ is forced Low, asserting the signal. In loopback mode, the $\overline{\text{RTSx}}$ signal is internally connected to the $\overline{\text{CTSx}}$ signal, which can be read via the CTS bit in the UARTxMSR register (see page 18-23).
0	DTR	**Data Terminal Ready** In normal operation (loopback mode disabled), this bit is the complement of the $\overline{\text{DTRx}}$ signal. 0 = $\overline{\text{DTRx}}$ is forced High, deasserting the signal. 1 = $\overline{\text{DTRx}}$ is forced Low, asserting the signal. In loopback mode, the $\overline{\text{DTRx}}$ signal is internally connected to the $\overline{\text{DSRx}}$ signal, which can be read via the DSR bit in the UARTxMSR register (see page 18-23).

Programming Notes

Direct-Mapped

UART 2 Line Status (UART2LSR) — I/O Address 02FDh

UART 1 Line Status (UART1LSR) — I/O Address 03FDh

	7	6	5	4	3	2	1	0
Bit	ERR_IN_FIFO	TEMT	THRE	BI	FE	PE	OE	DR
Reset	0	1	1	0	0	0	0	0
R/W	R!	R!	R!	R!	R!	R!	R!	R!

Register Description

This read-only register shows the status of the data transfer, with indicators for transmitter or transmit holding register empty, break detected, framing error, parity error, overrun error, and received data ready.

Bit Definitions

Bit	Name	Function
7	ERR_IN_FIFO	**16550-Compatible Mode Error** 0 = In 16550-compatible mode, there is *no* parity error, framing error, or break condition in the receive FIFO. In 16450-compatible mode, this bit always reads back 0. 1 = At least one parity error, framing error or break condition is present in the receive FIFO (16550-compatible mode only). This bit is cleared by a read from this register (UARTxLSR) or by a read from the receiver FIFO when there are no more error conditions present in the FIFO.
6	TEMT	**Transmitter Empty Indicator** 0 = The transmit shift register still has data to transmit. 1 = In 16450-compatible mode, both the transmit holding register and the transmit shift register are empty. In 16550-compatible mode, both the transmit FIFO and the transmit shift register are empty.
5	THRE	**Transmit Holding Register (16450-Compatible Mode) or Transmitter FIFO (16550-Compatible Mode) Empty** 0 = The transmitter still has data to place in the transmit shift register. 1 = In 16450-compatible mode, the transmit holding register is ready to accept a new character. In 16550-compatible mode, the transmit FIFO is completely empty. In 16450-compatible mode, this bit is automatically reset by a write to the UARTxTHR register (see page 18-7). In 16550-compatible mode, this interrupt is cleared when the transmit FIFO is written to. This bit can be used to generate an interrupt if programmed to do so via the Interrupt Enable register.
4	BI	**Break Indicator** 0 = There is no break indication associated with the current character. 1 = In 16450-compatible mode, this bit is set when the UART has detected that the sending UART has transmitted a break condition for a period longer than the time it takes to receive start, data, parity and stop bits. In 16550-compatible mode, this bit is set when an entire word (start, data, parity, stop) that was received into the FIFO with break indication present is now at the top of the FIFO. Only one break indication is loaded into the FIFO regardless of the duration of the break condition. A new character is not loaded into the FIFO until the next valid start bit is detected. This latched status bit is automatically cleared by a read from this register (UARTxLSR).

Bit	Name	Function
3	FE	**Framing Error** 0 = No framing error has been reported since line status was last read. 1 = In 16450-compatible mode, this bit is set to indicate that a received character did not have a valid stop bit. In 16550-compatible mode, this bit is set when a character that was received into the FIFO with a framing error is at the top of the receive FIFO. This latched status bit is automatically cleared by a read from this register (UARTxLSR).
2	PE	**Parity Error** 0 = There is no parity error associated with the current character. 1 = In 16450-compatible mode, this bit is set upon receipt of data with incorrect parity. In 16550-compatible mode, this bit is set when a character that was received into the FIFO with bad parity is at the top of the receive FIFO. This latched status bit is automatically cleared by a read from this register (UARTxLSR).
1	OE	**Overrun Error** 0 = No overrun error has been reported since line status was last read. 1 = In 16450-compatible mode, this bit is set if a new character is received into the receiver buffer before a previous character was read, thus resulting in lost data. In 16550-compatible mode, this bit is set if a new character is completely received into the shift register when the FIFO is already 100% full. Data in the FIFO is not overwritten by this overrun. The data in the shift register is lost when the next character is received. This latched status bit is automatically cleared by a read from this register (UARTxLSR).
0	DR	**Data Ready** 0 = There is no received data ready to read. 1 = In 16450-compatible mode, this bit is set when a character has been received and placed in the Receive Buffer register. In 16550-compatible mode, this bit is set when a character has been received and placed in the Receive FIFO. In 16450-compatible mode, this bit is automatically cleared by reading the UARTxRBR register (see page 18-8). In 16550-compatible mode, this bit is automatically cleared by reading the receiver FIFO, assuming that no more data is present in the FIFO.

Programming Notes

When a receiver line status interrupt is enabled and detected, bits BI, FE, PE, and OE in this register indicate the reason for the interrupt.

The status bits are valid even when line status interrupts are not enabled.

Direct-Mapped

UART 2 Modem Status (UART2MSR) — I/O Address 02FEh

UART 1 Modem Status (UART1MSR) — I/O Address 03FEh

	7	6	5	4	3	2	1	0
Bit	DCD	RI	DSR	CTS	DDCD	TERI	DDSR	DCTS
Reset	?	?	?	?	0	0	0	0
R/W	R	R	R	R	R!	R!	R!	R!

Register Description

This read-only register contains both real-time and latched control line status bits for the UART's $\overline{\text{DCDx}}$, $\overline{\text{RINx}}$, $\overline{\text{DSRx}}$ and $\overline{\text{CTSx}}$ input signals.

Bit Definitions

Bit	Name	Function
7	DCD	**Data Carrier Detect** In normal operation (loopback mode disabled), this bit is the complement of the $\overline{\text{DCDx}}$ signal. 0 = $\overline{\text{DCDx}}$ input signal is High (deasserted). 1 = $\overline{\text{DCDx}}$ input signal is Low (asserted). If in loopback mode, this bit tracks the OUT2 bit in the UARTxMCR register (see page 18-19).
6	RI	**Ring Indicator** In normal operation (loopback mode disabled), this bit is the complement of the $\overline{\text{RINx}}$ signal. 0 = $\overline{\text{RINx}}$ input signal is High (deasserted). 1 = $\overline{\text{RINx}}$ input signal is Low (asserted). If in loopback mode, this bit tracks bit OUT1 bit in the UARTxMCR register (see page 18-19).
5	DSR	**Data Set Ready** In normal operation (loopback mode disabled), this bit is the complement of the $\overline{\text{DSRx}}$ signal. 0 = $\overline{\text{DSRx}}$ input signal is High (deasserted). 1 = $\overline{\text{DSRx}}$ input signal is Low (asserted). If in loopback mode, this bit tracks bit DTR bit in the UARTxMCR register (see page 18-20).
4	CTS	**Clear To Send** In normal operation (loopback mode disabled), this bit is the complement of the $\overline{\text{CTSx}}$ signal. 0 = $\overline{\text{CTSx}}$ input signal is High (deasserted). 1 = $\overline{\text{CTSx}}$ input signal is Low (asserted). If in loopback mode, this bit tracks bit RTS bit in the UARTxMCR register (see page 18-20).
3	DDCD	**Delta Data Carrier Detect** 0 = Indicates that the $\overline{\text{DCDx}}$ signal has not changed since this register (UARTxMSR) was last read. 1 = Indicates that the $\overline{\text{DCDx}}$ signal changed since the UARTxMSR register was last read.
2	TERI	**Trailing Edge Ring Indicator** 0 = Indicates that the $\overline{\text{RINx}}$ signal has not changed from an active to an inactive state since this register (UARTxMSR) was last read. 1 = Indicates that the $\overline{\text{RINx}}$ signal changed from an active to an inactive state since the UARTxMSR register was last read.

Bit	Name	Function
1	DDSR	**Delta Data Set Ready** 0 = Indicates that the $\overline{\text{DSRx}}$ signal has not changed since this register (UARTxMSR) was last read. 1 = Indicates that the $\overline{\text{DSRx}}$ signal changed since the UARTxMSR register was last read.
0	DCTS	**Delta Clear To Send** 0 = Indicates that the $\overline{\text{CTSx}}$ signal has not changed since this register (UARTxMSR) was last read. 1 = Indicates that the $\overline{\text{CTSx}}$ signal changed since the UARTxMSR register was last read.

Programming Notes

Bits DCD, RI, DSR, and CTS are real-time status indicators (inverted) for the corresponding UART input signals.

DDCD, TERI, DDSR, and DCTS are latched status bits that generate an interrupt if modem status interrupts are unmasked in the UART x Interrupt Enable (UARTxINTENB) register. Reading this register (UARTxMSR) clears these bits and the associated interrupt.

Direct-Mapped

UART 2 Scratch Pad (UART2SCRATCH) — I/O Address 02FFh

UART 1 Scratch Pad (UART1SCRATCH) — I/O Address 03FFh

	7	6	5	4	3	2	1	0
Bit	SCRATCH[7–0]							
Reset	0	0	0	0	0	0	0	0
R/W	R/W							

Register Description

This location can be used to hold temporary data and is not required for serial data transfer.

Bit Definitions

Bit	Name	Function
7–0	SCRATCH [7–0]	**Scratch Bits** General-purpose I/O location, not required for serial data transfer.

Programming Notes

CHAPTER

19 SYNCHRONOUS SERIAL INTERFACE REGISTERS

19.1 OVERVIEW

This chapter describes the synchronous serial interface (SSI) registers of the ÉlanSC520 microcontroller.

The SSI provides efficient full-duplex or half-duplex bidirectional communication with peripheral devices that use a 4-pin or 3-pin serial interface.

The SSI register set consists of five memory-mapped configuration region (MMCR) registers used for SSI control, transmit, command, status, and receive functions. See the *Élan™SC520 Microcontroller User's Manual*, order #22004, for details about the SSI.

Table 19-1 lists the SSI registers in offset order, with the corresponding description's page number.

19.2 REGISTERS

Table 19-1 SSI MMCR Registers

Register Name	Mnemonic	MMCR Offset	Page Number
SSI Control	SSICTL	CD0h	page 19-2
SSI Transmit	SSIXMIT	CD1h	page 19-4
SSI Command	SSICMD	CD2h	page 19-5
SSI Status	SSISTA	CD3h	page 19-6
SSI Receive	SSIRCV	CD4h	page 19-7

SSI Control (SSICTL)

Memory-Mapped
MMCR Offset CD0h

	7	6	5	4	3	2	1	0
Bit	Reserved	CLK_SEL[2–0]			TC_INT_ ENB	PHS_INV_ ENB	CLK_INV_ ENB	MSBF_ENB
Reset	0	0	0	0	0	0	0	0
R/W	RSV	R/W			R/W	R/W	R/W	R/W

Register Description

This register controls the bit order, clock idle state, clock phase (for data drive and latch), interrupt enable, and clock speed of the SSI.

Bit Definitions

Bit — **Name** — **Function**

7 — Reserved

Reserved
This bit field should be written to 0 for normal system operation.

6–4 — CLK_SEL[2–0]

SSI Clock Speed Select
The SSI clock (SSI_CLK pin) frequency is derived from the system clock. The CLK_SEL bit field selects the frequency of the SSI clock as shown in Table 19-2.

Table 19-2 SSI Clock Speed Selections

CLK_SEL Bit Field	Selected Divisor	Nominal Bit Rate	Actual Bit Rate with a 33.000-MHz System Clock	Actual Bit Rate with a 33.333-MHz System CLock
000b	4d	8 MHz	8.250 MHz	8.333 MHz
001b	8d	4 MHz	4.125 MHz	4.167 MHz
010b	16d	2 MHz	2.063 MHz	2.083 MHz
011b	32d	1 MHz	1.031 MHz	1.042 MHz
100b	64d	512 kHz	516.6 kHz	520.8 kHz
101b	128d	256 kHz	257.8 kHz	260.4 kHz
110b	256d	128 kHz	128.9 kHz	130.2 kHz
111b	512d	64 kHz	64.5 kHz	65.1 kHz

3 — TC_INT_ENB

Transaction Complete Interrupt Enable
This bit is used to enable the TC_INT bit of the SSISTA register (see page 19-6) to generate an interrupt request.

0 = No interrupt is generated when the transaction is complete.

1 = The SSI issues an interrupt request when the transaction is complete.

If the interrupt is disabled, software should poll the BSY bit in the SSISTA register (see page 19-6).

Before the SSI interrupt is enabled, the SSIMAP register (see page 12-21) must be configured to route the interrupt to the appropriate interrupt request level and priority.

Bit	Name	Function
2	PHS_INV_ENB	**SSI Inverted Phase Mode Enable** This bit configures the SSI clock phase relationship with the incoming and outgoing data. 0 = Non-inverted phase mode: data is driven on odd edges of the SSI_CLK signal and latched on even edges. When writing data, the first bit of a transaction is shifted out on the SSI_DO pin on the first (odd) SSI clock edge, and the remaining bits are shifted out on odd clock edges. When reading data, the SSI_DI pin state is latched on the eight even edges (2,4,6,8,10,12,14,16) of the SSI clock. The transaction is complete one half of an SSI_CLK period after the last clock transition. 1 = Inverted-phase mode: data is driven on even edges of the SSI_CLK signal and latched on odd edges. When writing data, the first bit of a transaction is shifted out on the SSI_DO pin one half of an SSI_CLK period *before* the first (odd) SSI clock edge, and the remaining bits are shifted out on even clock edges. When reading data, the SSI_DI pin state is latched on the eight odd edges (1, 3, 5, 7, 9, 11, 13, 15) of the SSI clock. The transaction is complete on the last transition of the SSI clock.
1	CLK_INV_ENB	**SSI Inverted Clock Mode Enable** This bit controls the idle state of the SSI clock. The clock idle state is independent of the phase of the SSI clock. 0 = The clock is not inverted, and the idle state is High. The SSI clock pulses Low eight times. 1 = The clock is inverted, and the idle state is Low. The SSI clock pulses High eight times.
0	MSBF_ENB	**SSI Most Significant Bit First Mode Enable** This bit controls the bit order of data transfers. 0 = Bits are transmitted and received least significant bit (LSB) first. In this mode, the SSI shifts out the LSB of the transmit byte first. The first data bit received is stored in the LSB of the receive register and the last data bit received is stored in the most significant bit of the receive register. 1 = Bits are transmitted and received most significant bit (MSB) first. In this mode, the SSI shifts out the MSB of the transmit byte first. The first data bit received is stored in the MSB of the receive register and the last data bit received is stored in the LSB of the receive register. The configuration applies to both transmit and receive operations.

Programming Notes

This register should not be written while the BSY bit is set in the SSISTA register (see page 19-6).

SSI Transmit (SSIXMIT)

Memory-Mapped
MMCR Offset CD1h

	7	6	5	4	3	2	1	0
Bit	DAT_OUT[7–0]							
Reset	0	0	0	0	0	0	0	0
R/W	R/W							

Register Description

This register is used to write data to be transmitted.

Bit Definitions

Bit	Name	Function
7–0	DAT_OUT[7–0]	**SSI Data Out** Software uses this bit field to write eight bits to be shifted out on the SSI_DO pin. After writing to this bit field, software must write a Transmit command or a Simultaneous Transmit/Receive command to the SSICMD register (see page 19-5) to send the data. The contents of this bit field (DAT_OUT) are not destroyed when transmitted, so the user can repeatedly transmit the same data by writing the appropriate transmit command to the SSICMD register.

Programming Notes

This register should not be written while the BSY bit is set in the SSISTA register (see page 19-6).

SSI Command (SSICMD)

Memory-Mapped
MMCR Offset CD2h

	7	6	5	4	3	2	1	0
Bit	Reserved						CMD_SEL[1–0]	
Reset	0	0	0	0	0	0	0	0
R/W	RSV						R/W	

Register Description

This register is used to write the transfer command to be executed.

Bit Definitions

Bit	Name	Function
7–2	Reserved	**Reserved** This bit field should be written to 0 for normal system operation.
1–0	CMD_SEL[1–0]	**SSI Command Select** Three commands are available to initiate an SSI transaction. 00 = Reserved 01 = Transmit command: initiate a transaction in which the contents of the SSIXMIT register are shifted out (see page 19-4). 10 = Receive command: initiate a transaction in which data is shifted in to the SSIRCV register (see page 19-7). 11 = Simultaneous Transmit/Receive command: initiate a transaction in which both transmit and receive happen simultaneously. When read, this bit field returns the last command written to it.

Programming Notes

This register should not be written while the BSY bit is set in the SSISTA register (see page 19-6).

The CMD_SEL bit field is decoded and the command executed after the command is written. Software should load the SSIXMIT register (if necessary) before writing the command. The SSIRCV register can be read after the transaction is complete.

There is at least one 33-MHz clock period idle time between transactions.

A slave device should be enabled (if necessary) before a transmit or receive transaction is initiated, and the device should be disabled (if necessary) after the transaction is complete. Software can use programmable I/O (PIO) pins to implement device enable signals. See Chapter 20, “Programmable Input/Output Registers” for details about configuring PIO pins.

SSI Status (SSISTA)

Memory-Mapped
MMCR Offset CD3h

	7	6	5	4	3	2	1	0
Bit	Reserved						BSY	TC_INT
Reset	0	0	0	0	0	0	0	0
R/W	RSV						R	R/W!

Register Description

This register reports SSI port busy status and a latched transaction complete status.

Bit Definitions

Bit	Name	Function
7–2	Reserved	**Reserved** This bit field should be written to 0 for normal system operation.
1	BSY	**SSI Busy** This bit reports SSI activity status. 0 = The port is not busy. 1 = The port is busy. The port is busy (active) while a receive or transmit operation is in progress. This bit is set by the SSI hardware after a command is written, and cleared by hardware when the transaction is complete. Transaction complete status is also indicated by the TC_INT bit. Writes to the BSY bit have no effect.
0	TC_INT	**SSI Transaction Complete Interrupt** This bit indicates transaction complete status. 0 = No transaction has completed since software cleared this bit. 1 = A transaction has completed since software cleared this bit. Software must clear this bit by writing a 1. The TC_INT bit remains set until software acknowledges this completion by writing a 1. When the TC_INT bit is set, an interrupt request is generated if enabled via the TC_INT_ENB bit in the SSICTL register (see page 19-2). SSI activity is also indicated by the BSY bit.

Programming Notes

This register (SSISTA) should not be written while the BSY bit is set. Also, unreliable operation occurs if the SSICTL, SSIXMIT, or SSICMD registers are written, or if the SSIRCV register is read, while the BSY bit is set.

Hardware updates the BSY and TC_INT bits to indicate non-busy status (transaction complete) one half SSI_CLK period after the last edge of a receive transaction, or one full SSI_CLK period after the last edge of a transmit transaction.

The TC_INT bit should be used for interrupt operation, i.e., if the TC_INT_ENB bit is enabled in the SSICTL register (see page 19-2). Software must acknowledge (clear) the TC_INT bit by writing a 1 to it after each SSI transaction.

The BSY bit should be used for polled operation. Although the TC_INT bit can be used for polled operation, using the BSY bit is more efficient because it does not need to be cleared by software after each transaction.

Note that if only the polled operation is used, the TC_INT_ENB bit should not be enabled in the SSICTL register (see page 19-2).

SSI Receive (SSIRCV)

Memory-Mapped
MMCR Offset CD4h

	7	6	5	4	3	2	1	0
Bit	DAT_IN[7–0]							
Reset	0	0	0	0	0	0	0	0
R/W	R							

Register Description

This register is used to read data received from a peripheral device.

Bit Definitions

Bit	Name	Function
7–0	DAT_IN[7–0]	**SSI Data IN** Software uses this bit field to read the eight bits shifted in from the SSI_DI pin by the last Receive command or Simultaneous Transmit/Receive command that was issued via the SSICMD register (see page 19-5). After writing a receive command, software must wait until the receive transaction is complete before reading this bit field. Transaction complete status is indicated by the BSY and TC_INT bits in the SSISTA register (see page 19-6), or by an interrupt if enabled via the TC_INT_ENB bit in the SSICTL register (see page 19-2). Writes to this bit field (DAT_IN) have no effect.

Programming Notes

This register should not be read while the BSY bit is set in the SSISTA register (see page 19-6).

CHAPTER

20 PROGRAMMABLE INPUT/OUTPUT REGISTERS

20.1 OVERVIEW

This chapter describes the programmable input/output (PIO) pins and other multiplexed or configurable pins of the ÉlanSC520 microcontroller.

The PIO register set consists of 13 memory-mapped configuration region (MMCR) registers used to configure the programmable I/O pins (PIO31–PIO0), to enable interface functions for PIO pins and other multiplexed pins, and to adjust the drive strength of SDRAM interface pins. See the *Élan™SC520 Microcontroller User's Manual*, order #22004, for details about PIO pins.

Table 20-1 lists the PIO registers in offset order, with the corresponding description's page number.

Table 20-2 on page 20-2 provides an overview of how bits in the PIO pin configuration registers are used to control individual PIO pins.

20.2 REGISTERS

Table 20-1 Programmable I/O MMCR Registers

Register Name	Mnemonic	MMCR Offset	Page Number
PIO15–PIO0 Pin Function Select	PIOPFS15_0	C20h	page 20-3
PIO31–PIO16 Pin Function Select	PIOPFS31_16	C22h	page 20-5
Chip Select Pin Function Select	CSPFS	C24h	page 20-7
Clock Select	CLKSEL	C26h	page 20-9
Drive Strength Control	DSCTL	C28h	page 20-10
PIO15–PIO0 Direction	PIODIR15_0	C2Ah	page 20-12
PIO31–PIO16 Direction	PIODIR31_16	C2Ch	page 20-14
PIO15–PIO0 Data	PIODATA15_0	C30h	page 20-16
PIO31–PIO16 Data	PIODATA31_16	C32h	page 20-18
PIO15–PIO0 Set	PIOSET15_0	C34h	page 20-20
PIO31–PIO16 Set	PIOSET31_16	C36h	page 20-22
PIO15–PIO0 Clear	PIOCLR15_0	C38h	page 20-24
PIO31–PIO16 Clear	PIOCLR31_16	C3Ah	page 20-26

Table 20-2 PIO Register Programming Summary

Function Select Register Bit	Direction Register Bit	Data Register Bit (Writes)	Set Register Bit	Clear Register Bit	Data Register Bit (Reads)[1]	Resulting Programmable I/O Pin Function
1	X[2]	X	X	X	?[3]	The pin is not a PIO; it uses its interface function. The value of the pin can be read at the Data bit, but writes to the Direction, Data, Set, and Clear bits have no effect.
0	0	X	X	X	?	The PIO is an input. The state of the pin can be read at the Data bit. Writes to the Data, Set and Clear bits have no effect.
0	1	X	X	1[4]	0	The PIO is an output. The 1 that is written to the Clear bit causes this PIO pin to be driven Low. The state of the pin can be read at the Data bit, (in this case the pin is Low).
0	1	X	1	X	1	The PIO is an output. The 1 that is written to the Set bit causes this PIO pin to be driven High. The state of the pin can be read at the Data bit, (in this case the pin is High).
0	1	0	X	X	0	The PIO is an output. The 0 that is written to the Data bit causes this PIO pin to be driven Low. The state of the pin can be read at the Data bit, (in this case the pin is Low).
0	1	1	X	X	1	The PIO is an output. The 1 that is written to the Data bit causes this PIO pin to be driven High. The state of the pin can be read at the Data bit, (in this case the pin is High).

Notes:

1. The Data Register Bit (Reads) column shows the resulting state of the Data register bit and the corresponding PIO pin.
2. X = Not used in this operation.
3. ? = Input value. (The Data register bit state always reflects the corresponding pin state, whether input or output.)
4. For a particular PIO output operation, only one of the pin's Data, Set, or Clear bits can be used. The state of the unused bits is not important, but subsequent writes to these bits can change the PIO pin state.

PIO15–PIO0 Pin Function Select (PIOPFS15_0)

Memory-Mapped
MMCR Offset C20h

	15	14	13	12	11	10	9	8
Bit	PIO15_ FNC	PIO14_ FNC	PIO13_ FNC	PIO12_ FNC	PIO11_ FNC	PIO10_ FNC	PIO9_ FNC	PIO8_ FNC
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

	7	6	5	4	3	2	1	0
Bit	PIO7_ FNC	PIO6_ FNC	PIO5_ FNC	PIO4_ FNC	PIO3_ FNC	PIO2_ FNC	PIO1_ FNC	PIO0_ FNC
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Register Description

This register allows the programmer to choose the functionality of programmable I/O pins PIO15-PIO0.

Bit Definitions

Bit	Name	Function
15	PIO15_FNC	**PIO15 or GPIRQ8 Function Select** This bit is used to select the functionality of the PIO15 pin. 0 = The pin is PIO15. 1 = The pin is GPIRQ8.
14	PIO14_FNC	**PIO14 or GPIRQ9 Function Select** This bit is used to select the functionality of the PIO14 pin. 0 = The pin is PIO14. 1 = The pin is GPIRQ9.
13	PIO13_FNC	**PIO13 or GPIRQ10 Function Select** This bit is used to select the functionality of the PIO13 pin. 0 = The pin is PIO13. 1 = The pin is GPIRQ10.
12	PIO12_FNC	**PIO12 or $\overline{\text{GPDACK0}}$ Function Select** This bit is used to select the functionality of the PIO12 pin. 0 = The pin is PIO12. 1 = The pin is $\overline{\text{GPDACK0}}$.
11	PIO11_FNC	**PIO11 or $\overline{\text{GPDACK1}}$ Function Select** This bit is used to select the functionality of the PIO11 pin. 0 = The pin is PIO11. 1 = The pin is $\overline{\text{GPDACK1}}$.
10	PIO10_FNC	**PIO10 or $\overline{\text{GPDACK2}}$ Function Select** This bit is used to select the functionality of the PIO10 pin. 0 = The pin is PIO10. 1 = The pin is $\overline{\text{GPDACK2}}$.
9	PIO9_FNC	**PIO9 or $\overline{\text{GPDACK3}}$ Function Select** This bit is used to select the functionality of the PIO9 pin. 0 = The pin is PIO9. 1 = The pin is $\overline{\text{GPDACK3}}$.

Bit	Name	Function
8	PIO8_FNC	**PIO8 or GPDRQ0 Function Select** This bit is used to select the functionality of the PIO8 pin. 0 = The pin is PIO8. 1 = The pin is GPDRQ0.
7	PIO7_FNC	**PIO7 or GPDRQ1 Function Select** This bit is used to select the functionality of the PIO7 pin. 0 = The pin is PIO7. 1 = The pin is GPDRQ1.
6	PIO6_FNC	**PIO6 or GPDRQ2 Function Select** This bit is used to select the functionality of the PIO6 pin. 0 = The pin is PIO6. 1 = The pin is GPDRQ2.
5	PIO5_FNC	**PIO5 or GPDRQ3 Function Select** This bit is used to select the functionality of the PIO5 pin. 0 = The pin is PIO5. 1 = The pin is GPDRQ3.
4	PIO4_FNC	**PIO4 or GPTC Function Select** This bit is used to select the functionality of the PIO4 pin. 0 = The pin is PIO4. 1 = The pin is GPTC.
3	PIO3_FNC	**PIO3 or GPAEN Function Select** This bit is used to select the functionality of the PIO3 pin. 0 = The pin is PIO3. 1 = The pin is GPAEN.
2	PIO2_FNC	**PIO2 or GPRDY Function Select** This bit is used to select the functionality of the PIO2 pin. 0 = The pin is PIO2. 1 = The pin is GPRDY.
1	PIO1_FNC	**PIO1 or $\overline{\text{GPBHE}}$ Function Select** This bit is used to select the functionality of the PIO1 pin. 0 = The pin is PIO1. 1 = The pin is $\overline{\text{GPBHE}}$.
0	PIO0_FNC	**PIO0 or GPALE Function Select** This bit is used to select the functionality of the PIO0 pin. 0 = The pin is PIO0. 1 = The pin is GPALE.

Programming Notes

This register (PIOPFS15_0) should be written early in the microcontroller's initialization routine. The bit values to write depend on which pins are to be used for PIO functions, as opposed to interface functions. This depends on how the microcontroller is used in each particular system design.

On reset, each PIO pin is an input with a pullup or pulldown resistance for termination. See the pin list summary table in the *Élan™SC520 Microcontroller Data Sheet*, order #22003. Software writes a 1 to the corresponding bit in this register to change a pin to its interface function. For example, PIO2 shares a pin with the GP bus GPRDY signal, so before GPRDY can be used, a 1 must be written to the PIO2_FNC bit. To summarize:

- A bit must be cleared to use the corresponding pin as a programmable I/O pin.
- A bit must be set to 1 to use the corresponding pin for its interface function.

Although software can perform a 32-bit access of MMCR offset C20h to select all 32 PIO pin functions with a single instruction, the 32-bit access is split into two separate 16-bit accesses, with the PIOPFS15_0 register being accessed prior to the PIOPFS31_16 register. The two accesses are not simultaneous.

PIO31–PIO16 Pin Function Select (PIOPFS31_16)

Memory-Mapped
MMCR Offset C22h

	15	14	13	12	11	10	9	8
Bit	PIO31_ FNC	PIO30_ FNC	PIO29_ FNC	PIO28_ FNC	PIO27_ FNC	PIO26_ FNC	PIO25_ FNC	PIO24_ FNC
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

	7	6	5	4	3	2	1	0
Bit	PIO23_ FNC	PIO22_ FNC	PIO21_ FNC	PIO20_ FNC	PIO19_ FNC	PIO18_ FNC	PIO17_ FNC	PIO16_ FNC
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Register Description

This register allows the programmer to choose the functionality of programmable I/O pins PIO31–PIO16.

Bit Definitions

Bit	Name	Function
15	PIO31_FNC	**PIO31 or $\overline{\text{RIN2}}$ Function Select** This bit is used to select the functionality of the PIO31 pin. 0 = The pin is PIO31. 1 = The pin is $\overline{\text{RIN2}}$.
14	PIO30_FNC	**PIO30 or $\overline{\text{DCD2}}$ Function Select** This bit is used to select the functionality of the PIO30 pin. 0 = The pin is PIO30. 1 = The pin is $\overline{\text{DCD2}}$.
13	PIO29_FNC	**PIO29 or $\overline{\text{DSR2}}$ Function Select** This bit is used to select the functionality of the PIO29 pin. 0 = The pin is PIO29. 1 = The pin is $\overline{\text{DSR2}}$.
12	PIO28_FNC	**PIO28 or $\overline{\text{CTS2}}$ Function Select** This bit is used to select the functionality of the PIO28 pin. 0 = The pin is PIO28. 1 = The pin is $\overline{\text{CTS2}}$.
11	PIO27_FNC	**PIO27 or $\overline{\text{GPCS0}}$ Function Select** This bit is used to select the functionality of the PIO27 pin. 0 = The pin is PIO27. 1 = The pin is $\overline{\text{GPCS0}}$.
10	PIO26_FNC	**PIO26 or $\overline{\text{GPMEMCS16}}$ Function Select** This bit is used to select the functionality of the PIO26 pin. 0 = The pin is PIO26. 1 = The pin is $\overline{\text{GPMEMCS16}}$.
9	PIO25_FNC	**PIO25 or $\overline{\text{GPIOCS16}}$ Function Select** This bit is used to select the functionality of the PIO25 pin. 0 = The pin is PIO25. 1 = The pin is $\overline{\text{GPIOCS16}}$.

Bit	Name	Function
8	PIO24_FNC	**PIO24 or $\overline{\text{GPDBUFOE}}$ Function Select** This bit is used to select the functionality of the PIO24 pin. 0 = The pin is PIO24. 1 = The pin is $\overline{\text{GPDBUFOE}}$.
7	PIO23_FNC	**PIO23 or GPIRQ0 Function Select** This bit is used to select the functionality of the PIO23 pin. 0 = The pin is PIO23. 1 = The pin is GPIRQ0.
6	PIO22_FNC	**PIO22 or GPIRQ1 Function Select** This bit is used to select the functionality of the PIO22 pin. 0 = The pin is PIO22. 1 = The pin is GPIRQ1.
5	PIO21_FNC	**PIO21 or GPIRQ2 Function Select** This bit is used to select the functionality of the PIO21 pin. 0 = The pin is PIO21. 1 = The pin is GPIRQ2.
4	PIO20_FNC	**PIO20 or GPIRQ3 Function Select** This bit is used to select the functionality of the PIO20 pin. 0 = The pin is PIO20. 1 = The pin is GPIRQ3.
3	PIO19_FNC	**PIO19 or GPIRQ4 Function Select** This bit is used to select the functionality of the PIO19 pin. 0 = The pin is PIO19. 1 = The pin is GPIRQ4.
2	PIO18_FNC	**PIO18 or GPIRQ5 Function Select** This bit is used to select the functionality of the PIO18 pin. 0 = The pin is PIO18. 1 = The pin is GPIRQ5.
1	PIO17_FNC	**PIO17 or GPIRQ6 Function Select** This bit is used to select the functionality of the PIO17 pin. 0 = The pin is PIO17. 1 = The pin is GPIRQ6.
0	PIO16_FNC	**PIO16 or GPIRQ7 Function Select** This bit is used to select the functionality of the PIO16 pin. 0 = The pin is PIO16. 1 = The pin is GPIRQ7.

Programming Notes

This register (PIOPFS31_16) should be written early in the microcontroller's initialization routine. The bit values to write depend on which pins are to be used for PIO functions, as opposed to interface functions. This depends on how the microcontroller is used in each particular system design.

On reset, each PIO pin is an input with a pullup or pulldown resistance for termination. See the pin list summary table in the *Élan™SC520 Microcontroller Data Sheet*, order #22003. Software writes a 1 to the corresponding bit in this register to change a pin to its interface function. For example, PIO18 shares a pin with the GP bus GPIRQ5 signal, so before GPIRQ5 can be used, a 1 must be written to the PIO18_FNC bit. To summarize:

- A bit must be cleared to use the corresponding pin as a programmable I/O pin.
- A bit must be set to 1 to use the corresponding pin for its interface function.

Although software can perform a 32-bit access of MMCR offset C20h to select all 32 PIO pin functions with a single instruction, the 32-bit access is split into two separate 16-bit accesses, with the PIOPFS15_0 register being accessed prior to the PIOPFS31_16 register. The two accesses are not simultaneous.

Chip Select Pin Function Select (CSPFS)

Memory-Mapped
MMCR Offset C24h

	7	6	5	4	3	2	1	0
Bit	GPCS7_ SEL	GPCS6_ SEL	GPCS5_ SEL	GPCS4_ SEL	GPCS3_ SEL	GPCS2_ SEL	GPCS1_ SEL	Reserved
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	RSV

Register Description

This register selects the pin functionality for pins that have general-purpose chip selects ($\overline{\text{GPCSx}}$) as their alternate function.

Bit Definitions

Bit	Name	Function
7	GPCS7_SEL	**TMROUT0 or $\overline{\text{GPCS7}}$ Function Select** This bit is used to make either the TMROUT0 signal or the $\overline{\text{GPCS7}}$ signal available on the pin. The default on reset is TMROUT0. 0 = The pin is TMROUT0. 1 = The pin is $\overline{\text{GPCS7}}$.
6	GPCS6_SEL	**TMROUT1 or $\overline{\text{GPCS6}}$ Function Select** This bit is used to make either the TMROUT1 signal or the $\overline{\text{GPCS6}}$ signal available on the pin. The default on reset is TMROUT1. 0 = The pin is TMROUT1. 1 = The pin is $\overline{\text{GPCS6}}$.
5	GPCS5_SEL	**TMRIN0 or $\overline{\text{GPCS5}}$ Function Select** This bit is used to make either the TMRIN0 signal or the $\overline{\text{GPCS5}}$ signal available on the pin. The default on reset is TMRIN0. 0 = The pin is TMRIN0. 1 = The pin is $\overline{\text{GPCS5}}$.
4	GPCS4_SEL	**TMRIN1 or $\overline{\text{GPCS4}}$ Function Select** This bit is used to make either the TMRIN1 signal or the $\overline{\text{GPCS4}}$ signal available on the pin. The default on reset is TMRIN1. 0 = The pin is TMRIN1. 1 = The pin is $\overline{\text{GPCS4}}$.
3	GPCS3_SEL	**PITGATE2 or $\overline{\text{GPCS3}}$ Function Select** This bit is used to make either the PITGATE2 signal or the $\overline{\text{GPCS3}}$ signal available on the pin. The default on reset is PITGATE2. 0 = The pin is PITGATE2. 1 = The pin is $\overline{\text{GPCS3}}$.
2	GPCS2_SEL	**$\overline{\text{ROMCS2}}$ or $\overline{\text{GPCS2}}$ Function Select** This bit is used to make either the $\overline{\text{ROMCS2}}$ signal or the $\overline{\text{GPCS2}}$ signal available on the pin. The default on reset is $\overline{\text{ROMCS2}}$. 0 = The pin is $\overline{\text{ROMCS2}}$. 1 = The pin is $\overline{\text{GPCS2}}$.

Bit	Name	Function
1	GPCS1_SEL	**$\overline{\text{ROMCS1}}$ or $\overline{\text{GPCS1}}$ Function Select** This bit is used to make either the $\overline{\text{ROMCS1}}$ signal or the $\overline{\text{GPCS1}}$ signal available on the pin. The default on reset is $\overline{\text{ROMCS1}}$. 0 = The pin is $\overline{\text{ROMCS1}}$. 1 = The pin is $\overline{\text{GPCS1}}$.
0	Reserved	**Reserved** This bit field should be written to 0 for normal system operation.

Programming Notes

The $\overline{\text{GPCS0}}$ signal is shared with the PIO27 signal on one pin, so it is selected through the PIOPFS31_16 register (see page 20-5).

This register (CSPFS) should be written early in the microcontroller's initialization routine. The bit values to write depend on which function is to be used for each pin. This depends on how the microcontroller is used in each particular system design.

On reset, each pin's primary function is selected. See the pin list summary table in the *Élan™SC520 Microcontroller Data Sheet*, order #22003. Software writes a 1 to the corresponding bit in this register to change a pin to its alternate function. For example, TMROUT0 shares a pin with the GP bus $\overline{\text{GPCS7}}$ signal, so before $\overline{\text{GPCS7}}$ can be used, a 1 must be written to the GPCS7_SEL bit. To summarize:

- A bit must be cleared to use the corresponding pin for its primary function.
- A bit must be set to 1 to use the corresponding pin for its alternate function.

Clock Select (CLKSEL)

Memory-Mapped
MMCR Offset C26h

	7	6	5	4	3	2	1	0
Bit	Reserved	CLK_TST_SEL[2–0]			Reserved		CLK_PIN_DIR	CLK_PIN_ENB
Reset	0	1	1	1	0	0	0	0
R/W	RSV	R/W			RSV		R/W	R/W

Register Description

This register is used to set up the CLKTIMER[CLKTEST] pin.

Bit Definitions

Bit	Name	Function
7	Reserved	**Reserved** This bit field should be written to 0 for normal system operation.
6–4	CLK_TST_SEL[2–0]	**CLKTEST Pin Output Clock Select** 000 = 32.768 kHz (RTC clock) 001 = 1.8432 MHz (UART clock) 010 = 18.432 MHz (UART clock) 011 = 1.1892 MHz (PIT clock) 100 = 1.47456 MHz (PLL1 output) 101 = 36.864 MHz (PLL2 output) 110–111 = Disabled (pin stays Low)
3–2	Reserved	**Reserved** This bit field should be written to 0 for normal system operation.
1	CLK_PIN_DIR	**CLKTIMER[CLKTEST] Pin Direction** This bit determines whether the CLKTIMER[CLKTEST] pin is an input (CLKTIMER) or output (CLKTEST). As an input this pin provides the clock for the programmable interval timer (PIT). As an output, this pin drives the clock selected by the CLK_TST_SEL bit field externally. 0 = Input (CLKTIMER) 1 = Output (CLKTEST)
0	CLK_PIN_ENB	**CLKTIMER[CLKTEST] Pin Enable** The CLKTIMER[CLKTEST] pin is disabled on reset and must be enabled to function. 0 = Disabled 1 = Enabled

Programming Notes

The CLKTIMER[CLKTEST] pin can be configured as an output (CLKTEST) to drive any of the several internal clocks externally for testing, or to drive an external device. Caution should be exercised because there is no logic to avoid spurious pulses while enabling this pin as an output or changing clock frequencies. The target device should be held in reset while the CLK_TST_SEL, CLK_PIN_DIR, and CLK_PIN_ENB bit fields are configured to enable the pin as an output with the desired frequency, then the target device can be released from reset.

The CLKTIMER[CLKTEST] pin can be enabled as an input (CLKTIMER) if an external oscillator is to be used for the programmable interval timer (PIT). For example, a 1.19318-MHz CLKTIMER input can be used to provide PC/AT-compatible time-of-day operation without changing PIT counter values. (See the PIT chapter of the *Élan™SC520 Microcontroller User's Manual*, order #22004, for details.) While the pin is being enabled as an input, it is synchronized to the CPU clock to prevent spurious pulses from occurring in the PIT.

Drive Strength Control (DSCTL)

Memory-Mapped
MMCR Offset C28h

	15	14	13	12	11	10	9	8
Bit	Reserved						SCS_DRIVE[1–0]	
Reset	0	0	0	0	0	0	1	0
R/W	RSV						R/W	

	7	6	5	4	3	2	1	0
Bit	SRCW_DRIVE[1–0]		SDQM_DRIVE[1–0]		MA_DRIVE[1–0]		DATA_DRIVE[1–0]	
Reset	0	0	0	0	0	0	0	0
R/W	R/W		R/W		R/W		R/W	

Register Description

This register controls the drive strengths for the SDRAM interface signals. Independent drive strength control is provided for the address bus (MA12–MA0), bank select bus (BA1–BA0), SDQM3–SDQM0 bus, and $\overline{\text{SCS3}}$–$\overline{\text{SCS0}}$. The $\overline{\text{SRASA}}$–$\overline{\text{SRASB}}$, $\overline{\text{SCASA}}$–$\overline{\text{SCASB}}$ and $\overline{\text{SWEA}}$–$\overline{\text{SWEB}}$ control signals are grouped into a single drive strength control. The data bus (MD31–MD0) and ECC bus (MECC6–MECC0) are also grouped into a single drive strength control.

Note: *A programmable reset preserves this register's state. See the PRG_RST_ENB bit description on page 3-3.*

Bit Definitions

Bit	Name	Function
15–10	Reserved	**Reserved** This bit field should be written to 0 for normal system operation.
9–8	SCS_DRIVE [1–0]	**I/O Pad Drive Strength for $\overline{\text{SCS3}}$–$\overline{\text{SCS0}}$** These bits select the drive strength of I/O pads for the $\overline{\text{SCS3}}$–$\overline{\text{SCS0}}$ signals. 00 = Reserved 01 = 18-mA pads 10 = 12-mA pads (default) 11 = Reserved ***Note:*** *Default state of $\overline{\text{SCS3}}$–$\overline{\text{SCS0}}$ drive strength is 12 mA.*
7–6	SRCW_DRIVE [1–0]	**I/O Pad Drive Strength for $\overline{\text{SRASA}}$–$\overline{\text{SRASB}}$, $\overline{\text{SCASA}}$–$\overline{\text{SCASB}}$ and $\overline{\text{SWEA}}$–$\overline{\text{SWEB}}$** These bits select the drive strength of I/O pads for the $\overline{\text{SRASA}}$–$\overline{\text{SRASB}}$, $\overline{\text{SCASA}}$–$\overline{\text{SCASB}}$, and $\overline{\text{SWEA}}$–$\overline{\text{SWEB}}$ signals. 00 = 24-mA pads (default) 01 = 18-mA pads 10 = 12-mA pads 11 = Reserved
5–4	SDQM_DRIVE [1–0]	**I/O Pad Drive Strength for SDQM3–SDQM0** These bits select the drive strength of I/O pads for the SDQM3–SDQM0 signals. 00 = 24-mA pads (default) 01 = 18-mA pads 10 = 12-mA pads 11 = Reserved

3–2	MA_DRIVE [1–0]	**I/O Pad Drive Strength for MA12–MA0 and BA1–BA0** These bits select the drive strength of I/O pads for the MA12–MA0 and BA1–BA0 signals. 00 = 24-mA pads (default) 01 = 18-mA pads 10 = 12-mA pads 11 = Reserved
1–0	DATA_DRIVE [1–0]	**I/O Pad Drive Strength for MD31–MD0 and MECC6–MECC0** These bits select the drive strength of I/O pads for the MD31–MD0 and MECC6–MECC0 signals. 00 = 24-mA pads (default) 01 = 18-mA pads 10 = 12-mA pads 11 = Reserved

Programming Notes

PIO15–PIO0 Direction (PIODIR15_0)

Memory-Mapped
MMCR Offset C2Ah

	15	14	13	12	11	10	9	8
Bit	PIO15_ DIR	PIO14_ DIR	PIO13_ DIR	PIO12_ DIR	PIO11_ DIR	PIO10_ DIR	PIO9_ DIR	PIO8_ DIR
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

	7	6	5	4	3	2	1	0
Bit	PIO7_ DIR	PIO6_ DIR	PIO5_ DIR	PIO4_ DIR	PIO3_ DIR	PIO2_ DIR	PIO1_ DIR	PIO0_ DIR
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Register Description

This register contains the direction bits for pins PIO15–PIO0.

Bit Definitions

Bit	Name	Function
15	PIO15_DIR	**PIO15 Input or Output Select** This bit programs PIO15 as an input or output. 0 = Input 1 = Output
14	PIO14_DIR	**PIO14 Input or Output Select** This bit programs PIO14 as an input or output. 0 = Input 1 = Output
13	PIO13_DIR	**PIO13 Input or Output Select** This bit programs PIO13 as an input or output. 0 = Input 1 = Output
12	PIO12_DIR	**PIO12 Input or Output Select** This bit programs PIO12 as an input or output. 0 = Input 1 = Output
11	PIO11_DIR	**PIO11 Input or Output Select** This bit programs PIO11 as an input or output. 0 = Input 1 = Output
10	PIO10_DIR	**PIO10 Input or Output Select** This bit programs PIO10 as an input or output. 0 = Input 1 = Output

Bit	Name	Function
9	PIO9_DIR	**PIO9 Input or Output Select** This bit programs PIO9 as an input or output. 0 = Input 1 = Output
8	PIO8_DIR	**PIO8 Input or Output Select** This bit programs PIO8 as an input or output. 0 = Input 1 = Output
7	PIO7_DIR	**PIO7 Input or Output Select** This bit programs PIO7 as an input or output. 0 = Input 1 = Output
6	PIO6_DIR	**PIO6 Input or Output Select** This bit programs PIO6 as an input or output. 0 = Input 1 = Output
5	PIO5_DIR	**PIO5 Input or Output Select** This bit programs PIO5 as an input or output. 0 = Input 1 = Output
4	PIO4_DIR	**PIO4 Input or Output Select** This bit programs PIO4 as an input or output. 0 = Input 1 = Output
3	PIO3_DIR	**PIO3 Input or Output Select** This bit programs PIO3 as an input or output. 0 = Input 1 = Output
2	PIO2_DIR	**PIO2 Input or Output Select** This bit programs PIO2 as an input or output. 0 = Input 1 = Output
1	PIO1_DIR	**PIO1 Input or Output Select** This bit programs PIO1 as an input or output. 0 = Input 1 = Output
0	PIO0_DIR	**PIO0 Input or Output Select** This bit programs PIO0 as an input or output. 0 = Input 1 = Output

Programming Notes

The PIOx_DIR bit for each PIO pin chooses if the pin is an input or output. After reset, all of the PIO signals are inputs with pullup or pulldown termination. Before any PIO can be used as an output, this register (PIODIR15_0) must be programmed to change the PIO from an input to an output.

The PIOx_DIR bit for a pin has no effect if the corresponding PIOx_FNC bit is set in the PIOPFS15_0 register (see page 20-3). If the PIOx_FNC bit is set, the corresponding pin is assigned its interface function, not its PIO function.

Although software can perform a 32-bit access of MMCR offset C2Ah to set the direction for all 32 PIO pins with a single instruction, the 32-bit access is split into two separate 16-bit accesses, with the PIODIR15_0 register being accessed prior to the PIODIR31_16 register. The two accesses are not simultaneous.

PIO31–PIO16 Direction (PIODIR31_16)

Memory-Mapped
MMCR Offset C2Ch

	15	14	13	12	11	10	9	8
Bit	PIO31_ DIR	PIO30_ DIR	PIO29_ DIR	PIO28_ DIR	PIO27_ DIR	PIO26_ DIR	PIO25_ DIR	PIO24_ DIR
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

	7	6	5	4	3	2	1	0
Bit	PIO23_ DIR	PIO22_ DIR	PIO21_ DIR	PIO20_ DIR	PIO19_ DIR	PIO18_ DIR	PIO17_ DIR	PIO16_ DIR
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Register Description

This register contains the direction bits for pins PIO31–PIO16.

Bit Definitions

Bit	Name	Function
15	PIO31_DIR	**PIO31 Input or Output Select** This bit programs PIO31 as an input or output. 0 = Input 1 = Output
14	PIO30_DIR	**PIO30 Input or Output Select** This bit programs PIO30 as an input or output. 0 = Input 1 = Output
13	PIO29_DIR	**PIO29 Input or Output Select** This bit programs PIO29 as an input or output. 0 = Input 1 = Output
12	PIO28_DIR	**PIO28 Input or Output Select** This bit programs PIO28 as an input or output. 0 = Input 1 = Output
11	PIO27_DIR	**PIO27 Input or Output Select** This bit programs PIO27 as an input or output. 0 = Input 1 = Output
10	PIO26_DIR	**PIO26 Input or Output Select** This bit programs PIO26 as an input or output. 0 = Input 1 = Output

Bit	Name	Function
9	PIO25_DIR	**PIO25 Input or Output Select** This bit programs PIO25 as an input or output. 0 = Input 1 = Output
8	PIO24_DIR	**PIO24 Input or Output Select** This bit programs PIO24 as an input or output. 0 = Input 1 = Output
7	PIO23_DIR	**PIO23 Input or Output Select** This bit programs PIO23 as an input or output. 0 = Input 1 = Output
6	PIO22_DIR	**PIO22 Input or Output Select** This bit programs PIO22 as an input or output. 0 = Input 1 = Output
5	PIO21_DIR	**PIO21 Input or Output Select** This bit programs PIO21 as an input or output. 0 = Input 1 = Output
4	PIO20_DIR	**PIO20 Input or Output Select** This bit programs PIO20 as an input or output. 0 = Input 1 = Output
3	PIO19_DIR	**PIO19 Input or Output Select** This bit programs PIO19 as an input or output. 0 = Input 1 = Output
2	PIO18_DIR	**PIO18 Input or Output Select** This bit programs PIO18 as an input or output. 0 = Input 1 = Output
1	PIO17_DIR	**PIO17 Input or Output Select** This bit programs PIO17 as an input or output. 0 = Input 1 = Output
0	PIO16_DIR	**PIO16 Input or Output Select** This bit programs PIO16 as an input or output. 0 = Input 1 = Output

Programming Notes

The PIOx_DIR bit for each PIO pin chooses if the pin is an input or output. After reset, all of the PIO signals are inputs with pullup or pulldown termination. Before any PIO can be used as an output, this register (PIODIR31_16) must be programmed to change the PIO from an input to an output.

The PIOx_DIR bit for a pin has no effect if the corresponding PIOx_FNC bit is set in the PIOPFS31_16 register (see page 20-5). If the PIOx_FNC bit is set, the corresponding pin is assigned its interface function, not its PIO function.

Although software can perform a 32-bit access of MMCR offset C2Ah to set the direction for all 32 PIO pins with a single instruction, the 32-bit access is split into two separate 16-bit accesses, with the PIODIR15_0 register being accessed prior to the PIODIR31_16 register. The two accesses are not simultaneous.

PIO15–PIO0 Data (PIODATA15_0)

Memory-Mapped
MMCR Offset C30h

	15	14	13	12	11	10	9	8
Bit	PIO15_ DATA	PIO14_ DATA	PIO13_ DATA	PIO12_ DATA	PIO11_ DATA	PIO10_ DATA	PIO9_ DATA	PIO8_ DATA
Reset	?	?	?	?	?	?	?	?
R/W	R/W!	R/W!	R/W!	R/W!	R/W!	R/W!	R/W!	R/W!

	7	6	5	4	3	2	1	0
Bit	PIO7_ DATA	PIO6_ DATA	PIO5_ DATA	PIO4_ DATA	PIO3_ DATA	PIO2_ DATA	PIO1_ DATA	PIO0_ DATA
Reset	?	?	?	?	?	?	?	?
R/W	R/W!	R/W!	R/W!	R/W!	R/W!	R/W!	R/W!	R/W!

Register Description

This register is used to read or write the value for pins PIO15–PIO0.

Bit Definitions

Bit	Name	Function
15	PIO15_DATA	**Read or Write the PIO15 Pin** 0 = PIO15 is Low 1 = PIO15 is High
14	PIO14_DATA	**Read or Write the PIO14 Pin** 0 = PIO14 is Low 1 = PIO14 is High
13	PIO13_DATA	**Read or Write the PIO13 Pin** 0 = PIO13 is Low 1 = PIO13 is High
12	PIO12_DATA	**Read or Write the PIO12 Pin** 0 = PIO12 is Low 1 = PIO12 is High
11	PIO11_DATA	**Read or Write the PIO11 Pin** 0 = PIO11 is Low 1 = PIO11 is High
10	PIO10_DATA	**Read or Write the PIO10 Pin** 0 = PIO10 is Low 1 = PIO10 is High
9	PIO9_DATA	**Read or Write the PIO9 Pin** 0 = PIO9 is Low 1 = PIO9 is High
8	PIO8_DATA	**Read or Write the PIO8 Pin** 0 = PIO8 is Low 1 = PIO8 is High

Bit	Name	Function
7	PIO7_DATA	**Read or Write the PIO7 Pin** 0 = PIO7 is Low 1 = PIO7 is High
6	PIO6_DATA	**Read or Write the PIO6 Pin** 0 = PIO6 is Low 1 = PIO6 is High
5	PIO5_DATA	**Read or Write the PIO5 Pin** 0 = PIO5 is Low 1 = PIO5 is High
4	PIO4_DATA	**Read or Write the PIO4 Pin** 0 = PIO4 is Low 1 = PIO4 is High
3	PIO3_DATA	**Read or Write the PIO3 Pin** 0 = PIO3 is Low 1 = PIO3 is High
2	PIO2_DATA	**Read or Write the PIO2 Pin** 0 = PIO2 is Low 1 = PIO2 is High
1	PIO1_DATA	**Read or Write the PIO1 Pin** 0 = PIO1 is Low 1 = PIO1 is High
0	PIO0_DATA	**Read or Write the PIO0 Pin** 0 = PIO0 is Low 1 = PIO0 is High

Programming Notes

Each PIOx_DATA bit is used to read or write the value of the corresponding pin. If the pin is configured as a PIO output, then writing to this register selects the output level of the pin.

Note that the output state of a pin programmed to be a PIO can also be controlled via the PIOSET15_0 and PIOCLR15_0 registers (see page 20-20 and page 20-24).

Reading a pin's PIOx_DATA bit when the PIO is an output returns the state the pin was programmed for (High or Low). Reading a pin's PIOx_DATA bit when the pin's interface function is selected (via the corresponding PIOx_FCN bit in the PIOPFS15_0 register, see page 20-3) returns the state of the pin. In other words, reading a PIOx_DATA bit returns the state of the corresponding PIO pin regardless of how the other PIO registers are programmed.

Although software can perform a 32-bit access of MMCR offset C30h to read or write all 32 PIO pins with a single instruction, the 32-bit access is split into two separate 16-bit accesses, with the PIODATA15_0 register being accessed prior to the PIODATA31_16 register. The two accesses are not simultaneous.

PIO31–PIO16 Data (PIODATA31_16)

Memory-Mapped
MMCR Offset C32h

	15	14	13	12	11	10	9	8
Bit	PIO31_ DATA	PIO30_ DATA	PIO29_ DATA	PIO28_ DATA	PIO27_ DATA	PIO26_ DATA	PIO25_ DATA	PIO24_ DATA
Reset	?	?	?	?	?	?	?	?
R/W	R/W!	R/W!	R/W!	R/W!	R/W!	R/W!	R/W!	R/W!

	7	6	5	4	3	2	1	0
Bit	PIO23_ DATA	PIO22_ DATA	PIO21_ DATA	PIO20_ DATA	PIO19_ DATA	PIO18_ DATA	PIO17_ DATA	PIO16_ DATA
Reset	?	?	?	?	?	?	?	?
R/W	R/W!	R/W!	R/W!	R/W!	R/W!	R/W!	R/W!	R/W!

Register Description

This register is used to read or write the value for pins PIO31–PIO16.

Bit Definitions

Bit	Name	Function
15	PIO31_DATA	**Read or Write the PIO31 Pin** 0 = PIO31 is Low 1 = PIO31 is High
14	PIO30_DATA	**Read or Write the PIO30 Pin** 0 = PIO30 is Low 1 = PIO30 is High
13	PIO29_DATA	**Read or Write the PIO29 Pin** 0 = PIO29 is Low 1 = PIO29 is High
12	PIO28_DATA	**Read or Write the PIO28 Pin** 0 = PIO28 is Low 1 = PIO28 is High
11	PIO27_DATA	**Read or Write the PIO27 Pin** 0 = PIO27 is Low 1 = PIO27 is High
10	PIO26_DATA	**Read or Write the PIO26 Pin** 0 = PIO26 is Low 1 = PIO26 is High
9	PIO25_DATA	**Read or Write the PIO25 Pin** 0 = PIO25 is Low 1 = PIO25 is High
8	PIO24_DATA	**Read or Write the PIO24 Pin** 0 = PIO24 is Low 1 = PIO24 is High

Bit	Name	Function
7	PIO23_DATA	**Read or Write the PIO23 Pin** 0 = PIO23 is Low 1 = PIO23 is High
6	PIO22_DATA	**Read or Write the PIO22 Pin** 0 = PIO22 is Low 1 = PIO22 is High
5	PIO21_DATA	**Read or Write the PIO21 Pin** 0 = PIO21 is Low 1 = PIO21 is High
4	PIO20_DATA	**Read or Write the PIO20 Pin** 0 = PIO20 is Low 1 = PIO20 is High
3	PIO19_DATA	**Read or Write the PIO19 Pin** 0 = PIO19 is Low 1 = PIO19 is High
2	PIO18_DATA	**Read or Write the PIO18 Pin** 0 = PIO18 is Low 1 = PIO18 is High
1	PIO17_DATA	**Read or Write the PIO17 Pin** 0 = PIO17 is Low 1 = PIO17 is High
0	PIO16_DATA	**Read or Write the PIO16 Pin** 0 = PIO16 is Low 1 = PIO16 is High

Programming Notes

Each PIOx_DATA bit is used to read or write the value of the corresponding pin. If the pin is configured as a PIO output, then writing to this register selects the output level of the pin.

Note that the output state of a pin programmed to be a PIO can also be controlled via the PIOSET31_16 and PIOCLR31_16 registers (see page 20-22 and page 20-26).

Reading a pin's PIOx_DATA bit when the PIO is an output returns the state the pin was programmed for (High or Low). Reading a pin's PIOx_DATA bit when the pin's interface function is selected (via the corresponding PIOx_FCN bit in the PIOPFS31_16 register, see page 20-5) returns the state of the pin. In other words, reading a PIOx_DATA bit returns the state of the corresponding PIO pin regardless of how the other PIO registers are programmed.

Although software can perform a 32-bit access of MMCR offset C30h to read or write all 32 PIO pins with a single instruction, the 32-bit access is split into two separate 16-bit accesses, with the PIODATA15_0 register being accessed prior to the PIODATA31_16 register. The two accesses are not simultaneous.

PIO15–PIO0 Set (PIOSET15_0)

Memory-Mapped
MMCR Offset C34h

	15	14	13	12	11	10	9	8
Bit	PIO15_ SET	PIO14_ SET	PIO13_ SET	PIO12_ SET	PIO11_ SET	PIO10_ SET	PIO9_ SET	PIO8_ SET
Reset	x	x	x	x	x	x	x	x
R/W	W!	W!	W!	W!	W!	W!	W!	W!

	7	6	5	4	3	2	1	0
Bit	PIO7_ SET	PIO6_ SET	PIO5_ SET	PIO4_ SET	PIO3_ SET	PIO2_ SET	PIO1_ SET	PIO0_ SET
Reset	x	x	x	x	x	x	x	x
R/W	W!	W!	W!	W!	W!	W!	W!	W!

Register Description

This register is used to make the output level High selectively for pins PIO15–PIO0.

Bit Definitions

Bit	Name	Function
15	PIO15_SET	**PIO15 Set** 0 = No effect. 1 = Set thePIO15 signal High.
14	PIO14_SET	**PIO14 Set** 0 = No effect. 1 = Set the PIO14 signal High.
13	PIO13_SET	**PIO13 Set** 0 = No effect. 1 = Set the PIO13 signal High.
12	PIO12_SET	**PIO12 Set** 0 = No effect. 1 = Set the PIO12 signal High.
11	PIO11_SET	**PIO11 Set** 0 = No effect. 1 = Set the PIO11 signal High.
10	PIO10_SET	**PIO10 Set** 0 = No effect. 1 = Set the PIO10 signal High.
9	PIO9_SET	**PIO9 Set** 0 = No effect. 1 = Set the PIO9 signal High.
8	PIO8_SET	**PIO8 Set** 0 = No effect. 1 = Set the PIO8 signal High.

Bit	Name	Function
7	PIO7_SET	**PIO7 Set** 0 = No effect. 1 = Set the PIO7 signal High.
6	PIO6_SET	**PIO6 Set** 0 = No effect. 1 = Set the PIO6 signal High.
5	PIO5_SET	**PIO5 Set** 0 = No effect. 1 = Set the PIO5 signal High.
4	PIO4_SET	**PIO4 Set** 0 = No effect. 1 = Set the PIO4 signal High.
3	PIO3_SET	**PIO3 Set** 0 = No effect. 1 = Set the PIO3 signal High.
2	PIO2_SET	**PIO2 Set** 0 = No effect. 1 = Set the PIO2 signal High.
1	PIO1_SET	**PIO1 Set** 0 = No effect. 1 = Set the PIO1 signal High.
0	PIO0_SET	**PIO0 Set** 0 = No effect. 1 = Set the PIO0 signal High.

Programming Notes

Each PIOx_SET bit is used to drive the corresponding PIO output High. Writing a 1 to any bit of this register causes the corresponding PIO pin to be driven High if it is programmed to be an output (via the corresponding PIOx_DIR bit in the PIODIR15_0 register, see page 20-12).

Writing 1 to a pin's PIOx_SET bit overrides any previous write to the pin's PIOx_DATA or PIOx_CLR bit (see page 20-16 and page 20-24). Writing 0 to any bit in this register has no effect.

If a PIO pin is programmed to be an input, or if the pin is programmed for its interface function (via the corresponding PIOx_FNC bit in the PIOPFS15_0 register, see page 20-3), then writing to the pin's PIOx_SET bit has no effect.

Although software can perform a 32-bit access of MMCR offset C34h to set bits across all 32 PIO pins with a single instruction, the 32-bit access is split into two separate 16-bit accesses, with the PIOSET15_0 register being accessed prior to the PIOSET31_16 register. The two writes are not simultaneous.

PIO31–PIO16 Set (PIOSET31_16)

Memory-Mapped
MMCR Offset C36h

	15	14	13	12	11	10	9	8
Bit	PIO31_ SET	PIO30_ SET	PIO29_ SET	PIO28_ SET	PIO27_ SET	PIO26_ SET	PIO25_ SET	PIO24_ SET
Reset	x	x	x	x	x	x	x	x
R/W	W!	W!	W!	W!	W!	W!	W!	W!

	7	6	5	4	3	2	1	0
Bit	PIO23_ SET	PIO22_ SET	PIO21_ SET	PIO20_ SET	PIO19_ SET	PIO18_ SET	PIO17_ SET	PIO16_ SET
Reset	x	x	x	x	x	x	x	x
R/W	W!	W!	W!	W!	W!	W!	W!	W!

Register Description

This register is used to make the output level High selectively for pins PIO31–PIO16.

Bit Definitions

Bit	Name	Function
15	PIO31_SET	**PIO31 Set** 0 = No effect. 1 = Set the PIO31 signal High.
14	PIO30_SET	**PIO30 Set** 0 = No effect. 1 = Set the PIO30 signal High.
13	PIO29_SET	**PIO29 Set** 0 = No effect. 1 = Set the PIO29 signal High.
12	PIO28_SET	**PIO28 Set** 0 = No effect. 1 = Set the PIO28 signal High.
11	PIO27_SET	**PIO27 Set** 0 = No effect. 1 = Set the PIO27 signal High.
10	PIO26_SET	**PIO26 Set** 0 = No effect. 1 = Set the PIO26 signal High.
9	PIO25_SET	**PIO25 Set** 0 = No effect. 1 = Set the PIO25 signal High.
8	PIO24_SET	**PIO24 Set** 0 = No effect. 1 = Set the PIO24 signal High.

Bit	Name	Function
7	PIO23_SET	**PIO23 Set** 0 = No effect. 1 = Set the PIO23 signal High.
6	PIO22_SET	**PIO22 Set** 0 = No effect. 1 = Set the PIO22 signal High.
5	PIO21_SET	**PIO21 Set** 0 = No effect. 1 = Set the PIO21 signal High.
4	PIO20_SET	**PIO20 Set** 0 = No effect. 1 = Set the PIO20 signal High.
3	PIO19_SET	**PIO19 Set** 0 = No effect. 1 = Set the PIO19 signal High.
2	PIO18_SET	**PIO18 Set** 0 = No effect. 1 = Set the PIO18 signal High.
1	PIO17_SET	**PIO17 Set** 0 = No effect. 1 = Set the PIO17 signal High.
0	PIO16_SET	**PIO16 Set** 0 = No effect. 1 = Set the PIO16 signal High.

Programming Notes

Each PIOx_SET bit is used to drive the corresponding PIO output High. Writing a 1 to any bit of this register causes the corresponding PIO pin to be driven High if it is programmed to be an output (via the corresponding PIOx_DIR bit in the PIODIR31_16 register, see page 20-14).

Writing 1 to a pin's PIOx_SET bit overrides any previous write to the pin's PIOx_DATA or PIOx_CLR bit (see page 20-18 and page 20-26). Writing 0 to any bit in this register has no effect.

If a PIO pin is programmed to be an input, or if the pin is programmed for its interface function (via the corresponding PIOx_FNC bit in the PIOPFS31_16 register, see page 20-5), then writing to the pin's PIOx_SET bit has no effect.

Although software can perform a 32-bit access of MMCR offset C34h to set bits across all 32 PIO pins with a single instruction, the 32-bit access is split into two separate 16-bit accesses, with the PIOSET15_0 register being accessed prior to the PIOSET31_16 register. The two writes are not simultaneous.

PIO15–PIO0 Clear (PIOCLR15_0)

Memory-Mapped
MMCR Offset C38h

	15	14	13	12	11	10	9	8
Bit	PIO15_ CLR	PIO14_ CLR	PIO13_ CLR	PIO12_ CLR	PIO11_ CLR	PIO10_ CLR	PIO9_ CLR	PIO8_ CLR
Reset	x	x	x	x	x	x	x	x
R/W	W!	W!	W!	W!	W!	W!	W!	W!

	7	6	5	4	3	2	1	0
Bit	PIO7_ CLR	PIO6_ CLR	PIO5_ CLR	PIO4_ CLR	PIO3_ CLR	PIO2_ CLR	PIO1_ CLR	PIO0_ CLR
Reset	x	x	x	x	x	x	x	x
R/W	W!	W!	W!	W!	W!	W!	W!	W!

Register Description

This register is used to make the output level Low selectively for pins PIO15–PIO0.

Bit Definitions

Bit	Name	Function
15	PIO15_CLR	**PIO15 Clear** 0 = No effect. 1 = Drive the PIO15 signal Low.
14	PIO14_CLR	**PIO14 Clear** 0 = No effect. 1 = Drive the PIO14 signal Low.
13	PIO13_CLR	**PIO13 Clear** 0 = No effect. 1 = Drive the PIO13 signal Low.
12	PIO12_CLR	**PIO12 Clear** 0 = No effect. 1 = Drive the PIO12 signal Low.
11	PIO11_CLR	**PIO11 Clear** 0 = No effect. 1 = Drive the PIO11 signal Low.
10	PIO10_CLR	**PIO10 Clear** 0 = No effect. 1 = Drive the PIO10 signal Low.
9	PIO9_CLR	**PIO9 Clear** 0 = No effect. 1 = Drive the PIO9 signal Low.
8	PIO8_CLR	**PIO8 Clear** 0 = No effect. 1 = Drive the PIO8 signal Low.

Bit	Name	Function
7	PIO7_CLR	**PIO7 Clear** 0 = No effect. 1 = Drive the PIO7 signal Low.
6	PIO6_CLR	**PIO6 Clear** 0 = No effect. 1 = Drive the PIO6 signal Low.
5	PIO5_CLR	**PIO5 Clear** 0 = No effect. 1 = Drive the PIO5 signal Low.
4	PIO4_CLR	**PIO4 Clear** 0 = No effect. 1 = Drive the PIO4 signal Low.
3	PIO3_CLR	**PIO3 Clear** 0 = No effect. 1 = Drive the PIO3 signal Low.
2	PIO2_CLR	**PIO2 Clear** 0 = No effect. 1 = Drive the PIO2 signal Low.
1	PIO1_CLR	**PIO1 Clear** 0 = No effect. 1 = Drive the PIO1 signal Low.
0	PIO0_CLR	**PIO0 Clear** 0 = No effect. 1 = Drive the PIO0 signal Low.

Programming Notes

Each PIOx_CLR bit is used to drive the corresponding PIO output Low. Writing a 1 to any bit of this register causes the corresponding PIO pin to be driven Low if it is programmed to be an output (via the corresponding PIOx_DIR bit in the PIODIR15_0 register, see page 20-12).

Writing 1 to a pin's PIOx_CLR bit overrides any previous write to the pin's PIOx_DATA or SET bit (see page 20-16 and page 20-20). Writing 0 to any bit in this register has no effect.

If a PIO pin is programmed to be an input, or if the pin is programmed for its interface function (via the corresponding PIOx_FNC bit in the PIOPFS15_0 register, see page 20-3), then writing to the pin's PIOx_CLR bit has no effect.

Although software can perform a 32-bit access of MMCR offset C38h to clear bits across all 32 PIO pins with a single instruction, the 32-bit access is split into two separate 16-bit accesses, with the PIOCLR15_0 register being accessed prior to the PIOCLR31_16 register. The two writes are not simultaneous.

PIO31–PIO16 Clear (PIOCLR31_16)

Memory-Mapped
MMCR Offset C3Ah

	15	14	13	12	11	10	9	8
Bit	PIO31_ CLR	PIO30_ CLR	PIO29_ CLR	PIO28_ CLR	PIO27_ CLR	PIO26_ CLR	PIO25_ CLR	PIO24_ CLR
Reset	x	x	x	x	x	x	x	x
R/W	W!	W!	W!	W!	W!	W!	W!	W!

	7	6	5	4	3	2	1	0
Bit	PIO23_ CLR	PIO22_ CLR	PIO21_ CLR	PIO20_ CLR	PIO19_ CLR	PIO18_ CLR	PIO17_ CLR	PIO16_ CLR
Reset	x	x	x	x	x	x	x	x
R/W	W!	W!	W!	W!	W!	W!	W!	W!

Register Description

This register is used to make the output level Low selectively for pins PIO31–PIO16.

Bit Definitions

Bit	Name	Function
15	PIO31_CLR	**PIO31 Clear** 0 = No effect. 1 = Drive the PIO31 signal Low.
14	PIO30_CLR	**PIO30 Clear** 0 = No effect. 1 = Drive the PIO30 signal Low.
13	PIO29_CLR	**PIO29 Clear** 0 = No effect. 1 = Drive the PIO29 signal Low.
12	PIO28_CLR	**PIO28 Clear** 0 = No effect. 1 = Drive the PIO28 signal Low.
11	PIO27_CLR	**PIO27 Clear** 0 = No effect. 1 = Drive the PIO27 signal Low.
10	PIO26_CLR	**PIO26 Clear** 0 = No effect. 1 = Drive the PIO26 signal Low.
9	PIO25_CLR	**PIO25 Clear** 0 = No effect. 1 = Drive the PIO25 signal Low.
8	PIO24_CLR	**PIO24 Clear** 0 = No effect. 1 = Drive the PIO24 signal Low.

Bit	Name	Function
7	PIO23_CLR	**PIO23 Clear** 0 = No effect. 1 = Drive the PIO23 signal Low.
6	PIO22_CLR	**PIO22 Clear** 0 = No effect. 1 = Drive the PIO22 signal Low.
5	PIO21_CLR	**PIO21 Clear** 0 = No effect. 1 = Drive the PIO21 signal Low.
4	PIO20_CLR	**PIO20 Clear** 0 = No effect. 1 = Drive the PIO20 signal Low.
3	PIO19_CLR	**PIO19 Clear** 0 = No effect. 1 = Drive the PIO19 signal Low.
2	PIO18_CLR	**PIO18 Clear** 0 = No effect. 1 = Drive the PIO18 signal Low.
1	PIO17_CLR	**PIO17 Clear** 0 = No effect. 1 = Drive the PIO17 signal Low.
0	PIO16_CLR	**PIO16 Clear** 0 = No effect. 1 = Drive the PIO16 signal Low.

Programming Notes

Each PIOx_CLR bit is used to drive the corresponding PIO output Low. Writing a 1 to any bit of this register causes the corresponding PIO pin to be driven Low if it is programmed to be an output (via the corresponding PIOx_DIR bit in the PIODIR31_16 register, see page 20-14).

Writing 1 to a pin's PIOx_CLR bit overrides any previous write to the pin's PIOx_DATA or SET bit (see page 20-18 and page 20-22). Writing 0 to any bit in this register has no effect.

If a PIO pin is programmed to be an input, or if the pin is programmed for its interface function (via the corresponding PIOx_FNC bit in the PIOPFS31_16 register, see page 20-5), then writing to the pin's PIOx_CLR bit has no effect.

Although software can perform a 32-bit access of MMCR offset C38h to clear bits across all 32 PIO pins with a single instruction, the 32-bit access is split into two separate 16-bit accesses, with the PIOCLR15_0 register being accessed prior to the PIOCLR31_16 register. The two writes are not simultaneous.

INDEX

Numerics

A

B

C

D

E

F

G

H

I

L

M

N

O

P

R

S

T

U

V

W

X

Y