

# Élan™ SC300 Microcontroller

## Programmer's Reference Manual

**Rev. B, January 1996**

© 1995 by Advanced Micro Devices, Inc.

Advanced Micro Devices reserves the right to make changes in its products without notice in order to improve design or performance characteristics.

This publication neither states nor implies any warranty of any kind, including but not limited to implied warrants of merchantability or fitness for a particular application. AMD assumes no responsibility for the use of any circuitry other than the circuitry in an AMD product.

The information in this publication is believed to be accurate in all respects at the time of publication, but is subject to change without notice. AMD assumes no responsibility for any errors or omissions, and disclaims responsibility for any consequences resulting from the use of the information included herein. Additionally, AMD assumes no responsibility for the functioning of undescribed features or parameters.

### **Trademarks**

AMD, the AMD logo, Am386, Am486, and FusionE86 are registered trademarks of Advanced Micro Devices, Inc.

Élan is a trademark of Advanced Micro Devices, Inc.

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

---

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

### Customer Service

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

### Hotline, E-Mail, and Bulletin Board Support

For answers to technical questions, AMD provides a toll-free number for direct access to our engineering support staff. For overseas customers, the easiest way to reach the engineering support staff with your questions is via fax with a short description of your question. AMD E86 family customers also receive technical support through electronic mail. This worldwide service is available to E86 family product users via the international UNIX E-mail service. Also available is the AMD bulletin board service, which provides the latest E86 family product information, including technical information and data on upcoming product releases.

### Engineering Support Staff

(800) 292-9263, extension 2	Toll-free for U.S.
0031-11-1163	Toll-free for Japan
(512) 602-4118	Direct dial worldwide (toll applies)
+44-(0)256-811101	U.K. and Europe hotline (toll applies)
(512) 602-5031	Fax (toll applies)
epd.support@amd.com	E-mail

### Bulletin Board Service

(800) 292-9263, extension 1	Toll-free for U.S.
(512) 602-7604	Direct dial worldwide (toll applies)

### Documentation and Literature

A simple phone call gets you free printed publications, such as data books, user's manuals, data sheets, application notes, the FusionE86® Partner Solutions Catalog and Newsletter, and other literature. For electronic copies of the most current E86 family product information and publications, visit our worldwide web site on the Internet.

### Literature Fulfillment

(800) 292-9263, extension 3	Toll-free for U.S.
(512) 602-5651	Direct dial worldwide (toll applies)
(512) 602-7639	Fax for U.S. (toll applies)
<a href="http://www.amd.com">http://www.amd.com</a>	World Wide Web





# TABLE OF CONTENTS

## INTRODUCTION

PURPOSE OF THIS MANUAL .....	xvii
IN THIS MANUAL .....	xviii
RELATED AMD PUBLICATIONS .....	xviii

## Chapter 1 POWER MANAGEMENT

1.1	POWER MANAGEMENT UNIT .....	1-2
1.1.1	Power-Management Modes .....	1-3
1.1.1.1	High-Speed PLL Mode .....	1-4
1.1.1.2	Low-Speed PLL Mode .....	1-5
1.1.1.3	Doze Mode .....	1-5
1.1.1.4	Sleep Mode .....	1-7
1.1.1.5	Suspend Mode .....	1-7
1.1.1.6	Off Mode .....	1-7
1.1.2	PMU Operating-Mode Transitions .....	1-8
1.1.3	PMU Clock Sources .....	1-9
1.1.4	Reading the PMU Mode .....	1-9
1.1.5	Merging of PMU Modes .....	1-9
1.1.6	Programming Example: Power-Management Setup .....	1-10
1.1.7	Programming Example: Peripheral-Device Power .....	1-11
1.2	EXTERNAL-DEVICE CONTROL INTERFACE .....	1-14
1.2.1	Power-Management Control Pins .....	1-14
1.2.1.1	General-Purpose Control Using the PMU State Machine .....	1-14
1.2.1.2	Timer-Controlled Shutdown Using the SMI Interface .....	1-14
1.2.2	Programmable General-Purpose Pins 2 and 3 .....	1-15
1.2.3	LCD-Panel Voltage-Control Pins .....	1-15
1.2.4	Latched Power Pin .....	1-16
1.3	CLOCK-SWITCHING LOGIC .....	1-16
1.3.1	CPU/Memory Clock Switching .....	1-16
1.3.2	Clock Startup and Shutdown Logic .....	1-17
1.4	ACTIVITY MONITORS .....	1-19
1.5	STATE-TRANSITION TIMER .....	1-22
1.6	WAKE-UP LOGIC .....	1-23
1.7	SMI AND NMI CONTROL .....	1-25
1.7.1	Temporary-On Mode .....	1-26
1.7.2	Enabling SMIs .....	1-27
1.7.3	Processing NMI or SMI Source .....	1-28
1.7.4	Accesses to Powered-Down Device SMI .....	1-29
1.7.5	Treatment of Pending SMIs .....	1-31
1.7.6	External SMI Pin .....	1-31

1.7.7	External SMI with a Single Device . . . . .	1-31
1.7.8	External SMI with Multiple Devices. . . . .	1-31
1.8	BATTERY-MANAGEMENT LOGIC . . . . .	1-32
1.8.1	Battery Level 1 . . . . .	1-32
1.8.2	Battery Level 2 . . . . .	1-32
1.8.3	Battery Level 3 . . . . .	1-33
1.8.4	Battery Level 4 . . . . .	1-33
1.8.5	AC Input Status Pin . . . . .	1-34
1.9	SUSPEND/RESUME PIN LOGIC. . . . .	1-34
1.9.1	Required Initialization . . . . .	1-37
1.9.2	Start of SMI Handler . . . . .	1-37
1.9.3	Suspend Input Caused the SMI . . . . .	1-37
1.9.4	Suspend Pseudocode . . . . .	1-38
1.9.5	Resume Input Caused the SMI . . . . .	1-39
1.9.6	Resume Pseudocode . . . . .	1-39
1.9.7	Things to Avoid. . . . .	1-39
1.10	AUTO LOW-SPEED LOGIC . . . . .	1-40
1.11	MICRO POWER OFF MODE . . . . .	1-40
1.12	OTHER POWER-SAVING FEATURES . . . . .	1-41
1.12.1	DMA Clock Stop . . . . .	1-41
1.12.2	Data-Path Disabling Logic . . . . .	1-41
1.12.3	Slow Refresh . . . . .	1-41
1.12.4	Quiet Bus . . . . .	1-41

**Chapter 2 MEMORY AND PCMCIA MANAGEMENT**

2.1	SYSTEM MEMORY: DRAM, SRAM, AND BUS. . . . .	2-2
2.1.1	DRAM Configurations. . . . .	2-3
2.1.2	SRAM Configurations. . . . .	2-3
2.1.3	Refresh and Wait States. . . . .	2-4
2.1.4	ISA, Local Bus, and Internal LCD Configurations. . . . .	2-6
2.2	ROM BIOS MEMORY. . . . .	2-7
2.3	ROM DOS MEMORY . . . . .	2-9
2.4	MEMORY MAPPING . . . . .	2-10
2.5	PCMCIA CONTROLLER . . . . .	2-16
2.5.1	Memory-Window Map. . . . .	2-19
2.5.2	I/O Window Map . . . . .	2-20
2.5.3	Control Logic . . . . .	2-20
2.5.4	Status Logic . . . . .	2-21
2.5.5	Programmable Logic . . . . .	2-21
2.5.6	Interrupt Handler. . . . .	2-22
2.6	OTHER MEMORY CONTROLLER INFORMATION . . . . .	2-23
2.6.1	ROM Chip-Select Command Gating. . . . .	2-23
2.6.2	Wait States and Command Delays. . . . .	2-23
2.6.3	High-Speed Clock ROM Cycles . . . . .	2-25
2.6.4	ROM Chip-Select Signal. . . . .	2-25
2.6.5	DOS Chip-Select Signal . . . . .	2-25
2.6.6	Self-Refresh DRAMs . . . . .	2-26
2.6.7	80-ns DRAM Support . . . . .	2-26

**Chapter 3 VIDEO CONTROLLER**

3.1	SIGNAL INTERFACES . . . . .	3-1
3.1.1	Memory Interface . . . . .	3-1
3.1.2	Display Interface . . . . .	3-2
3.1.3	Miscellaneous Signals Interface . . . . .	3-3
3.2	DISPLAY MEMORY . . . . .	3-3
3.3	GRAPHICS MODE . . . . .	3-4
3.4	TEXT MODE . . . . .	3-7
3.4.1	Character Byte . . . . .	3-9
3.4.2	Attribute Byte . . . . .	3-9
3.4.3	CGA Attribute Byte . . . . .	3-9
3.4.4	HGA Attribute Byte . . . . .	3-11
3.4.5	SRAM Display Data Area Partitioning . . . . .	3-11
3.5	FONTS . . . . .	3-13
3.5.1	Font Areas . . . . .	3-13
3.5.2	Video Controller Font Fetches to SRAM . . . . .	3-14
3.5.3	Storing Fonts in Video SRAM . . . . .	3-15
3.5.4	Font Example 1: The Letter A . . . . .	3-16
3.5.5	Font Example 2: The Letter M . . . . .	3-18
3.6	VIDEO REGISTER OVERVIEW . . . . .	3-20
3.6.1	Port Registers . . . . .	3-20
3.6.2	Index Registers . . . . .	3-20
3.7	VIDEO PORT REGISTERS . . . . .	3-21
3.7.1	HGA Index Address Register (Port 3B4h) . . . . .	3-21
3.7.2	HGA Index Data Register (Port 3B5h) . . . . .	3-22
3.7.3	HGA Mode Control Register (Port 3B8h) . . . . .	3-22
3.7.4	HGA Status Register (Port 3BAh) . . . . .	3-22
3.7.5	HGA Configuration Register (Port 3BFh) . . . . .	3-23
3.7.6	CGA Index Address Register (Port 3D4h) . . . . .	3-23
3.7.7	CGA Index Data Register (Port 3D5h) . . . . .	3-23
3.7.8	CGA Mode Control Register (Port 3D8h) . . . . .	3-24
3.7.9	CGA Color Select Register (Port 3D9h) . . . . .	3-25
3.7.10	CGA Status Register (Port 3DAh) . . . . .	3-26
3.8	STANDARD VIDEO INDEX REGISTERS . . . . .	3-26
3.8.1	Horizontal Registers (Video Indexes 00–03h) . . . . .	3-26
3.8.2	The Vertical Registers (Video Indexes 04–07h) . . . . .	3-27
3.8.3	Interlace Mode Register (Video Index 08h) . . . . .	3-27
3.8.4	Max Scan Line Register (Video Index 09h) . . . . .	3-27
3.8.5	Cursor Start and End Registers (Video Indexes 0A–0Bh) . . . . .	3-27
3.8.6	Start Address Registers (Video Indexes 0C–0Dh) . . . . .	3-27
3.8.7	Cursor Address Registers (Video Indexes 0E–0Fh) . . . . .	3-27
3.8.8	Reserved Registers (Video Indexes 10–11h) . . . . .	3-28
3.9	EXTENDED VIDEO INDEX REGISTERS . . . . .	3-28
3.9.1	Enable Software Switch Register (Video Index 12h) . . . . .	3-28
3.9.2	Disable Software Switch Register (Video Index 13h) . . . . .	3-28
3.9.3	Color Mapping Registers (Video Indexes 14–17h & 1C–1Fh) . . . . .	3-29
3.9.4	Screen Control Restore Register (Video Index 18h) . . . . .	3-30
3.9.5	Screen Control 2 Register (Video Index 19h) . . . . .	3-31

3.9.5.1	Auto Screen Blanking . . . . .	3-32
3.9.5.2	Display Controller Enable . . . . .	3-32
3.9.6	Screen Adjust Lower Byte Register (Video Index 1Ah) . . . . .	3-32
3.9.7	Screen Adjust Upper Byte Register (Video Index 1Bh) . . . . .	3-32
3.9.8	Control 1 Register (Video Index 20h) . . . . .	3-34
3.9.9	Text Truncation Start Register (Video Index 21h) . . . . .	3-35
3.9.10	Text Truncation Stop Register (Video Index 22h) . . . . .	3-35
3.9.11	Graphics Truncation Start Register (Video Index 23h) . . . . .	3-36
3.9.12	Graphics Truncation Stop Register (Video Index 24h) . . . . .	3-36
3.9.13	LCD Special Register (Video Index 25h) . . . . .	3-37
3.9.14	Programming Examples . . . . .	3-37

## Chapter 4 PC/AT PERIPHERAL REGISTERS

4.1	PC/AT-COMPATIBLE PORT REGISTERS . . . . .	4-2
4.1.1	Interrupt Controller Registers . . . . .	4-2
4.1.2	Programmable Interval Timer Registers . . . . .	4-3
4.1.3	DMA Controller Registers . . . . .	4-4
4.1.4	DMA Page Registers . . . . .	4-5
4.1.5	Parallel Port Interface Registers . . . . .	4-6
4.1.5.1	AT-Compatible Mode . . . . .	4-7
4.1.5.2	EPP-Compliant Mode . . . . .	4-8
4.1.6	UART Registers . . . . .	4-9
4.1.6.1	Transmitter Holding Register (Ports 2F8h & 3F8h) . . . . .	4-9
4.1.6.2	Receiver Buffer Register (Ports 2F8h & 3F8h) . . . . .	4-9
4.1.6.3	Divisor Latch Lower Byte (Ports 2F8h & 3F8h) . . . . .	4-10
4.1.6.4	Divisor Latch Upper Byte (Ports 2F8h & 3F9h) . . . . .	4-10
4.1.6.5	Interrupt Enable Register (Ports 2F9h & 3F9h) . . . . .	4-10
4.1.6.6	Interrupt Identification Register (Ports 2FAh & 3FAh) . . . . .	4-10
4.1.6.7	Line Control Register (Ports 2FBh & 3FBh) . . . . .	4-11
4.1.6.8	Modem Control Register (Ports 2FCh & 3FCh) . . . . .	4-12
4.1.6.9	Line Status Register (Ports 2FDh & 3FDh) . . . . .	4-12
4.1.6.10	Modem Status Register (Ports 2FEh & 3FEh) . . . . .	4-13
4.1.6.11	Scratch Pad Register (Ports 2FFh & 3FFh) . . . . .	4-13
4.1.7	Real-Time Clock . . . . .	4-14
4.1.7.1	Addressing . . . . .	4-14
4.1.7.2	RTC Registers . . . . .	4-15
4.2	MISCELLANEOUS PC/AT-COMPATIBLE PORT REGISTERS . . . . .	4-18
4.2.1	XT Keyboard Data Register (Port 060h) . . . . .	4-18
4.2.2	Port B Register (Port 061h) . . . . .	4-18
4.2.3	NMI/RTC Index Address Register (Port 070h) . . . . .	4-19
4.2.4	RTC Index Data Register (Port 071h) . . . . .	4-19
4.2.5	Port 92 (Port 092h) . . . . .	4-19

## Chapter 5 CONFIGURATION REGISTERS

5.1	CONFIGURATION REGISTER OVERVIEW . . . . .	5-2
5.1.1	Port Registers . . . . .	5-2
5.1.2	Index Registers . . . . .	5-2
5.1.2.1	Mandatory Settings . . . . .	5-2
5.1.2.2	Recommended Settings . . . . .	5-4



---

5.1.3	Configuring the ÉlanSC300 Microcontroller . . . . .	5-4
5.1.3.1	To Control the PC/AT Bus and Its Timing . . . . .	5-4
5.1.3.2	To Determine the Bus Configuration . . . . .	5-4
5.1.3.3	To Control CPU and PC/AT Compatibility . . . . .	5-5
5.1.3.4	To Control the Speed of the CPU. . . . .	5-5
5.1.3.5	To Control Direct Memory Accesses . . . . .	5-5
5.1.3.6	To Enable Interrupts and Specify How They Are Mapped . . . . .	5-5
5.1.3.7	To Set Up Memory Mapping (MMS Windows) . . . . .	5-6
5.1.3.8	To Set Up the System DRAM. . . . .	5-6
5.1.3.9	To Set Up the System SRAM. . . . .	5-7
5.1.3.10	To Set Up the Parallel Port. . . . .	5-7
5.1.3.11	To Set Up the UART. . . . .	5-7
5.1.3.12	To Set Up the PCMCIA Card Interface. . . . .	5-7
5.1.3.13	To Set Up the General-Purpose and PMC Pins. . . . .	5-9
5.1.3.14	To Control the Clocks (Phase-Locked Loops) . . . . .	5-9
5.1.3.15	To Control Power Management Activities and Events . . . . .	5-10
5.1.3.16	To Determine Power Management Status . . . . .	5-10
5.1.3.17	To Control the Power Management State Timers . . . . .	5-11
5.1.3.18	To Map ROM Accesses and Control ROM Cycles. . . . .	5-11
5.1.3.19	To Control SMIs and Determine Status . . . . .	5-11
5.1.3.20	To Control the Video Display . . . . .	5-12
5.2	CONFIGURATION PORT REGISTERS. . . . .	5-12
5.2.1	Configuration Address Register (Port 022h). . . . .	5-12
5.2.2	Configuration Data Register (Port 023h) . . . . .	5-12
5.3	CONFIGURATION INDEX REGISTERS . . . . .	5-13
5.3.1	PCMCIA I/O Window A1 Lower Byte Start Register (Index 00h) . . . . .	5-13
5.3.2	PCMCIA I/O Window A1 Lower Byte End Register (Index 01h). . . . .	5-13
5.3.3	PCMCIA I/O Window A1 Upper Byte Register (Index 02h) . . . . .	5-13
5.3.4	PCMCIA I/O Window A2 Lower Byte Start Register (Index 03h) . . . . .	5-13
5.3.5	PCMCIA I/O Window A2 Lower Byte End Register (Index 04h). . . . .	5-14
5.3.6	PCMCIA I/O Window A2 Upper Byte Register (Index 05h) . . . . .	5-14
5.3.7	PCMCIA I/O Card IRQ Redirection Control A Register (Index 06h) . . . . .	5-14
5.3.8	PCMCIA VPPA Address Register (Index 07h) . . . . .	5-15
5.3.9	Resume Mask Register (Index 08h). . . . .	5-16
5.3.10	Resume Status Register (Index 09h) . . . . .	5-17
5.3.11	PCMCIA Data Width Register (Index 0Ah). . . . .	5-17
5.3.12	Reserved Register (Index 0Bh) . . . . .	5-18
5.3.13	PCMCIA Socket B Status Register (Index 0Ch). . . . .	5-19
5.3.14	PCMCIA Status Change IRQ Enable Register (Index 0Dh) . . . . .	5-19
5.3.15	PCMCIA Status Change IRQ Redirection Register (Index 0Eh) . . . . .	5-20
5.3.16	Reserved Register (Index 0Fh) . . . . .	5-21
5.3.17	PCMCIA I/O Window B1 Lower Byte Start Register (Index 10h) . . . . .	5-21
5.3.18	PCMCIA I/O Window B1 Lower Byte End Register (Index 11h). . . . .	5-22
5.3.19	PCMCIA I/O Window B1 Upper Byte Register (Index 12h) . . . . .	5-22
5.3.20	PCMCIA I/O Window B2 Lower Byte Start Register (Index 13h) . . . . .	5-22
5.3.21	PCMCIA I/O Window B2 Lower Byte End Register (Index 14h). . . . .	5-22
5.3.22	PCMCIA I/O Window B2 Upper Byte Register (Index 15h) . . . . .	5-23
5.3.23	PCMCIA I/O Card IRQ Redirection Control B Register (Index 16h) . . . . .	5-23
5.3.24	PCMCIA VPPB Address Register (Index 17h) . . . . .	5-24

5.3.25	Reserved Registers (Indexes 18–39h) . . . . .	5-24
5.3.26	I/O Timeout Register (Index 40h) . . . . .	5-24
5.3.27	SMI Enable Register (Index 41h) . . . . .	5-25
5.3.28	SMI I/O Status Register (Index 42h) . . . . .	5-26
5.3.29	SMI Status Register (Index 43h) . . . . .	5-26
5.3.30	Miscellaneous 4 Register (Index 44h) . . . . .	5-27
5.3.31	PIO Address Register (Index 45h) . . . . .	5-27
5.3.32	PIO Timer Register (Index 46h) . . . . .	5-27
5.3.33	Drive Timer Register (Index 47h) . . . . .	5-29
5.3.34	Control A Register (Index 48h) . . . . .	5-30
5.3.35	Reserved Registers (Indexes 49–4Fh) . . . . .	5-30
5.3.36	MMS Memory Wait State 2 Register (Index 50h) . . . . .	5-30
5.3.37	ROM Configuration 2 Register (Index 51h) . . . . .	5-32
5.3.38	Reserved Registers (Indexes 52–5Fh) . . . . .	5-32
5.3.39	Command Delay Register (Index 60h) . . . . .	5-32
5.3.40	I/O Wait State Register (Index 61h) . . . . .	5-34
5.3.41	MMS Memory Wait State 1 Register (Index 62h) . . . . .	5-35
5.3.42	Wait State Control Register (Index 63h) . . . . .	5-37
5.3.43	Version Register (Index 64h) . . . . .	5-38
	5.3.43.1 Read Functions . . . . .	5-38
	5.3.43.2 Write Functions . . . . .	5-39
5.3.44	ROM Configuration 1 Register (Index 65h) . . . . .	5-40
5.3.45	Memory Configuration 1 Register (Index 66h) . . . . .	5-41
5.3.46	MMSA Address Extension 1 Register (Index 67h) . . . . .	5-42
5.3.47	Shadow RAM Enable 1 Register (Index 68h) . . . . .	5-42
5.3.48	Shadow RAM Enable 2 Register (Index 69h) . . . . .	5-43
5.3.49	Reserved Register (Index 6Ah) . . . . .	5-43
5.3.50	Miscellaneous 2 Register (Index 6Bh) . . . . .	5-43
5.3.51	MMS Address Extension 1 Register (Index 6Ch) . . . . .	5-44
5.3.52	MMS Address Register (Index 6Dh) . . . . .	5-44
5.3.53	MMS Address Extension 2 Register (Index 6Eh) . . . . .	5-46
5.3.54	Miscellaneous 1 Register (Index 6Fh) . . . . .	5-47
5.3.55	Miscellaneous 6 Register (Index 70h) . . . . .	5-48
5.3.56	MMSA Device 1 Register (Index 71h) . . . . .	5-49
5.3.57	MMSA Device 2 Register (Index 72h) . . . . .	5-50
5.3.58	MMSB Device Register (Index 73h) . . . . .	5-51
5.3.59	MMSB Control Register (Index 74h) . . . . .	5-52
5.3.60	Activity Mask 1 Register (Index 75h) . . . . .	5-53
5.3.61	Activity Mask 2 Register (Index 76h) . . . . .	5-54
5.3.62	Control B Register (Index 77h) . . . . .	5-55
5.3.63	Reserved Registers (Indexes 78–7Fh) . . . . .	5-55
5.3.64	Power Control 1 Register (Index 80h) . . . . .	5-55
5.3.65	Power Control 2 Register (Index 81h) . . . . .	5-56
5.3.66	NMI/SMI Enable Register (Index 82h) . . . . .	5-56
5.3.67	High-Speed to Low-Speed Timer Register (Index 83h) . . . . .	5-57
5.3.68	Low-Speed to Doze Timer Register (Index 84h) . . . . .	5-57
5.3.69	Doze to Sleep Timer Register (Index 85h) . . . . .	5-57
5.3.70	Sleep to Suspend Timer Register (Index 86h) . . . . .	5-57
5.3.71	Suspend to Off Timer Register (Index 87h) . . . . .	5-58

5.3.72	Software Mode Control Register (Index 88h)	5-58
5.3.73	General-Purpose I/O 0 Register (Index 89h)	5-59
5.3.74	PCMCIA REGA Address Register (Index 8Ah)	5-59
5.3.75	Reserved Register (Index 8Bh)	5-59
5.3.76	I/O Activity Address 0 Register (Index 8Ch)	5-59
5.3.77	I/O Activity Address 1 Register (Index 8Dh)	5-60
5.3.78	Reserved Register (Index 8Eh)	5-60
5.3.79	Clock Control Register (Index 8Fh)	5-60
5.3.80	Reserved Register (Index 90h)	5-61
5.3.81	General-Purpose I/O Control Register (Index 91h)	5-61
5.3.82	UART Clock Enable Register (Index 92h)	5-63
5.3.83	Reserved Register (Index 93h)	5-63
5.3.84	General-Purpose I/O 2 Register (Index 94h)	5-63
5.3.85	General-Purpose I/O 3 Register (Index 95h)	5-63
5.3.86	Reserved Registers (Indexes 96–99h)	5-64
5.3.87	Memory Write Activity Lower Boundary Register (Index 9Ah)	5-64
5.3.88	Memory Write Activity Upper Boundary Register (Index 9Bh)	5-64
5.3.89	General-Purpose I/O 1 Register (Index 9Ch)	5-65
5.3.90	Reserved Register (Index 9Dh)	5-66
5.3.91	PCMCIA REGB Address Register (Index 9Eh)	5-66
5.3.92	Auto Low-Speed Control Register (Index 9Fh)	5-66
5.3.93	Activity Status 1 Register (Index A0h)	5-67
5.3.94	Activity Status 2 Register (Index A1h)	5-67
5.3.95	PCMCIA Socket A Status Register (Index A2h)	5-68
5.3.96	CPU Status 0 Register (Index A3h)	5-68
5.3.97	CPU Status 1 Register (Index A4h)	5-69
5.3.98	NMI/SMI Control Register (Index A5h)	5-70
5.3.99	PCMCIA Status Change Register (Index A6h)	5-70
5.3.100	PMU Control 1 Register (Index A7h)	5-71
5.3.101	MMSA Socket Register (Index A8h)	5-72
5.3.102	MMSB Socket Register (Index A9h)	5-73
5.3.103	SMI MMS Page Register (Index AAh)	5-73
5.3.104	Power Control 3 Register (Index ABh)	5-74
5.3.105	Power Control 4 Register (Index ACh)	5-75
5.3.106	PMU Control 3 Register (Index ADh)	5-76
5.3.107	Reserved Register (Index AEh)	5-76
5.3.108	PMU Control 2 Register (Index AFh)	5-77
5.3.109	Function Enable 1 Register (Index B0h)	5-77
5.3.110	Function Enable 2 Register (Index B1h)	5-78
5.3.111	PIRQ Configuration Register (Index B2h)	5-79
5.3.112	Miscellaneous 5 Register (Index B3h)	5-81
5.3.113	PCMCIA Card Reset Register (Index B4h)	5-82
5.3.114	CA24–CA25 Control 1 Register (Index B5h)	5-82
5.3.115	CA24–CA25 Control 2 Register (Index B6h)	5-83
5.3.116	CA24–CA25 Control 3 Register (Index B7h)	5-83
5.3.117	ROM Configuration 3 Register (Index B8h)	5-84
5.3.118	Memory Configuration 2 Register (Index B9h)	5-85
5.3.119	Miscellaneous 3 Register (Index BAh)	5-86

---

**Appendix A CONFIGURATION INDEX REGISTER REFERENCE****Appendix B XT-KEYBOARD INTERFACE**

B.1	XT KEYBOARD ENABLE .....	B-2
B.2	KEYBOARD INTERFACE CONTROL .....	B-2
B.3	KEYBOARD DATA PORT .....	B-2
B.4	I/O MAP SUMMARY .....	B-2
B.5	PINS USED .....	B-2
B.6	TIMING .....	B-3

**INDEX**

---

**LIST OF FIGURES**

1-1	PMU Operating-Mode Transitions . . . . .	1-4
1-2	PLL Control Flowchart . . . . .	1-18
1-3	State Transition Flowchart . . . . .	1-20
1-4	SMI Processing Flowchart . . . . .	1-26
1-5	SMI Device-Powerdown Flowchart. . . . .	1-30
2-1	Typical AT Address Space . . . . .	2-2
2-2	ÉlanSC300 Microcontroller Bus Configurations . . . . .	2-7
2-3	High Memory . . . . .	2-8
2-4	Copying ROM Contents . . . . .	2-9
2-5	Memory Mapping System . . . . .	2-11
2-6	MMSA and MMSB . . . . .	2-11
2-7	MMS Mapping Example . . . . .	2-14
2-8	PCMCIA Controller Block Diagram. . . . .	2-18
B-1	XT Keyboard Block Diagram . . . . .	B-1
B-2	XT Keyboard Timing Diagram . . . . .	B-3

## LIST OF TABLES

1-1	PMU Clock Speeds . . . . .	1-6
1-2	Inactivity States and Transition Intervals . . . . .	1-10
1-3	Power-Management Setup . . . . .	1-11
1-4	Power-Management Control Pin Settings. . . . .	1-12
1-5	PIO Timeout Settings . . . . .	1-12
1-6	SMI-Generation Settings for PIO Accesses . . . . .	1-13
1-7	PMC Pin Functionality . . . . .	1-14
1-8	Wake-Up Signal Descriptions . . . . .	1-23
1-9	Registers that Enable SMI Sources . . . . .	1-27
1-10	Battery-Level Management Functionality . . . . .	1-32
2-1	Memory Initialization Example . . . . .	2-3
2-2	SRAM Option Configuration . . . . .	2-4
2-3	Refresh Initialization Example . . . . .	2-4
2-4	Memory-Speed Initialization Example . . . . .	2-5
2-5	33-MHz Wait States . . . . .	2-6
2-6	Bus Configuration . . . . .	2-6
2-7	ROM BIOS Address Initialization . . . . .	2-8
2-8	MMS Mapping Example Settings . . . . .	2-14
2-9	ÉlanSC300 Microcontroller PCMCIA Signal Compatibility . . . . .	2-16
2-10	Memory-Window Mapping Registers . . . . .	2-19
2-11	Command Delay Duration for Various Cycles . . . . .	2-24
2-12	Wait States for Various Cycles . . . . .	2-24
2-13	ROMCS Wait-State Control-Bit Logic . . . . .	2-25
2-14	DOSCS Wait-State Control-Bit Logic . . . . .	2-26
3-1	Memory Interface Signals . . . . .	3-2
3-2	Display Interface Signals . . . . .	3-2
3-3	Miscellaneous Signals . . . . .	3-3
3-4	SRAM Address Mapping during CPU Access to Video Memory . . . . .	3-4
3-5	640×200 and 320×200 Graphics-Mode Video-Memory Data . . . . .	3-5
3-6	640×200 Graphics-Mode Video-Memory Data for a 480×320 LCD Panel. . . . .	3-6
3-7	CGA Text Mode Using Character Fonts 1 Area . . . . .	3-7
3-8	CGA Text Mode Using Character Fonts 2 Area . . . . .	3-8
3-9	CGA Text Mode Using Special Character Fonts Area . . . . .	3-8
3-10	HGA Text Mode . . . . .	3-8
3-11	CGA Foreground-Color Bit Values . . . . .	3-10
3-12	CGA Background-Color Bit Values. . . . .	3-10
3-13	HGA Attribute-Byte Bit Values . . . . .	3-11
3-14	Text-Mode Video-Memory Data: 640×200 Display, 80×25 Characters . . . . .	3-12
3-15	Text-Mode Video-Memory Data: 640×200 Display, 40×25 Characters . . . . .	3-12
3-16	Text-Mode Video-Memory Data: 480×320 LCD Panel, 60×40 Characters . . . . .	3-13
3-17	Font Areas . . . . .	3-14
3-18	SRAM Address Mapping during Video Controller Font Fetches. . . . .	3-15

---

3-19	Storing the Font for the Letter A . . . . .	3-17
3-20	Selecting the Font Area for the Letter A . . . . .	3-17
3-21	Storing the Font for the Letter M. . . . .	3-19
3-22	Selecting the Font Area for the Letter M. . . . .	3-19
3-23	Video Port Definitions . . . . .	3-20
3-24	6845 and Extended Registers . . . . .	3-21
3-25	Screen-Background Color-Bit Logic . . . . .	3-25
3-26	Enabling the Extended Video-Controller Registers . . . . .	3-28
3-27	Disabling the Extended Video-Controller Registers . . . . .	3-28
3-28	Gray-Scale Mapping Using the Color-Mapping Registers . . . . .	3-29
3-29	Mapping CGA Colors to Gray Scales . . . . .	3-30
3-30	LCD-Panel Select Logic . . . . .	3-31
3-31	Auto Screen-Blanking Timer Setting. . . . .	3-32
3-32	Page Boundaries in 80×25 Text Mode. . . . .	3-33
3-33	Printer I/O Port Select Logic . . . . .	3-34
3-34	Truncation-Register Programming Example. . . . .	3-36
3-35	640×200 LCD Panel Configuration . . . . .	3-38
3-36	480×320 LCD Panel Configuration . . . . .	3-39
3-37	320×240 LCD Panel Configuration . . . . .	3-40
3-38	320×240 LCD Panel Configuration . . . . .	3-41
3-39	720×348 LCD Panel Configuration . . . . .	3-42
4-1	Interrupt Controller 1 . . . . .	4-2
4-2	Interrupt Controller 2. . . . .	4-3
4-3	System Timer Registers . . . . .	4-3
4-4	DMA Controller 1 . . . . .	4-4
4-5	DMA Controller 2 . . . . .	4-5
4-6	DMA Page Registers . . . . .	4-6
4-7	Interrupt ID Bit Logic . . . . .	4-11
4-8	Word Length Bit Logic . . . . .	4-11
4-9	Real-Time Clock . . . . .	4-14
4-10	RTC Register Summary . . . . .	4-15
4-11	Register A Periodic-Interrupt Rate-Selection Bits (32.768 kHz) . . . . .	4-16
4-12	Register A Time-Base Divider-Chain Bits. . . . .	4-16
5-1	Configuration Port Registers. . . . .	5-2
5-2	Mandatory Configuration Bit Settings . . . . .	5-3
5-3	Recommended Configuration Bit Settings . . . . .	5-4
5-4	Interrupt Redirect Bit Logic . . . . .	5-15
5-5	ICIOA16S0–ICIOA16S1 Bit Logic . . . . .	5-18
5-6	ICIOB16S0–ICIOB16S1 Bit Logic. . . . .	5-18
5-7	IRQ Select Logic. . . . .	5-21
5-8	Interrupt Redirection Bit Logic . . . . .	5-24
5-9	PIO Timer Setting Logic . . . . .	5-28
5-10	PIO Address Range Decode Logic. . . . .	5-28

5-11	Hard Drive and Floppy Disk Drive Timer Setting Bit Logic . . . . .	5-29
5-12	ROM DOS Wait State Select Logic . . . . .	5-31
5-13	ROM DOS Command Delay Select Logic . . . . .	5-31
5-14	PCMCIA Wait State Select Logic . . . . .	5-31
5-15	PCMCIA Command Delay Select Logic . . . . .	5-32
5-16	8-Bit ISA I/O Access Command Delay . . . . .	5-33
5-17	8-bit ISA Memory Access Command Delay . . . . .	5-33
5-18	ROM BIOS Wait States . . . . .	5-33
5-19	Refresh Cycle Wait States . . . . .	5-33
5-20	Floppy Disk Drive Wait States . . . . .	5-34
5-21	Hard Drive Wait States . . . . .	5-35
5-22	General Bus I/O Wait States. . . . .	5-35
5-23	8-Bit ISA Memory-Cycle Wait States . . . . .	5-36
5-24	16-Bit ISA Memory-Cycle Wait States . . . . .	5-36
5-25	DRAM First Cycle Wait State Select Logic. . . . .	5-37
5-26	DRAM Bank Miss Wait State Select Logic . . . . .	5-37
5-27	SRAM Wait State Select Logic . . . . .	5-38
5-28	Read Version Stepping Level Decode . . . . .	5-38
5-29	Refresh Interval Select Logic . . . . .	5-39
5-30	Bus Option Status Table. . . . .	5-41
5-31	Memory Configuration (DRAM and SRAM) . . . . .	5-41
5-32	MMSA/B Page Register I/O Addresses . . . . .	5-45
5-33	Page Register Contents Description. . . . .	5-45
5-34	MMSA Base Addresses . . . . .	5-46
5-35	MMS Memory Range Select Logic . . . . .	5-47
5-36	PMU Mode Select Logic . . . . .	5-58
5-37	PLL Restart Time Select Logic . . . . .	5-61
5-38	Hit-Count Limit Bit Logic . . . . .	5-65
5-39	Trigger Period Select Logic . . . . .	5-66
5-40	Low-Speed Duration Period Select Logic. . . . .	5-67
5-41	Last PMU Mode Indicator Bits . . . . .	5-69
5-42	Present PMU Mode Indicator Bits. . . . .	5-69
5-43	Low-Speed PLL Mode CPU Clock Speed Select. . . . .	5-76
5-44	Latch and Buffer Logic . . . . .	5-78
5-45	High-Speed PLL Frequency Select . . . . .	5-79
5-46	RAM Mode Decode Logic. . . . .	5-79
5-47	Interrupt Redirect Logic . . . . .	5-80
5-48	ROM BIOS Enable and Wait-State Select Logic . . . . .	5-81
5-49	ROM DOS Enable and Wait-State Select Logic. . . . .	5-84
5-50	ROM DOS Linear Address Decode Size Select Logic . . . . .	5-85
5-51	Output Drive Strength Select Logic . . . . .	5-86
5-52	I/O Drive Type Description . . . . .	5-86
5-53	Parallel Port Pin Redefinition . . . . .	5-87





---

## INTRODUCTION

---

The Élan™ SC300 microcontroller is a highly integrated, low-power, single-chip system, composed of the Am386SXLV microprocessor plus the additional logic needed for an AT-compatible personal computer. It is designed for palmtop and battery-powered handheld computers. The ÉlanSC300 microcontroller features precise power management and high integration while providing the user with industry-standard interfaces.

The ÉlanSC300 microcontroller's architecture consists of the following functional modules:

- The Am386SXLV CPU itself, which is optimized for 3.3-V operation and includes System Management mode (SMM) power-management hardware
- A power-management controller that interfaces to the CPU's SMM and is tightly integrated with the internal clock-generator hardware
- A PC/AT-compatible memory controller and associated mapping hardware
- Two Revision-2.1 PCMCIA slots that are efficiently integrated with the memory controller
- An integrated LCD controller, optional CPU local bus, or full ISA-bus controller
- Standard PC/AT system logic and peripheral controllers (DMA, interrupt controller, timer, ISA-bus controller, and EPP parallel port)
- Additional peripheral controllers (UART and real-time clock)

### PURPOSE OF THIS MANUAL

This manual provides supplemental information unique to the ÉlanSC300 microcontroller—power- and memory-management programming considerations, video-design characteristics, and a detailed description of the additional registers.

For logic and software descriptions applicable to the Am386SXLV CPU, see the *3-Volt System Logic for Personal Computers* manual, PID 17028.

For information on the PC/AT-compatible peripherals in the ÉlanSC300 microcontroller (e.g., 8259A, 8254), refer to the documentation for the individual part.

## IN THIS MANUAL

This manual is organized in the following manner:

Chapter 1, “Power Management,” explains how to use the power-management features of the ÉlanSC300 microcontroller to conserve battery power.

Chapter 2, “Memory and PCMCIA Management,” describes the address spaces, including those that support the PCMCIA ports.

Chapter 3, “Video Controller,” describes the integrated video controller and the associated video I/O ports and video index registers.

Chapter 4, “PC/AT Peripheral Registers,” provides information on the standard PC/AT peripheral registers.

Chapter 5, “Configuration Registers,” provides information on the ÉlanSC300 microcontroller’s configuration registers.

Appendix A, “Configuration Index Register Reference,” lists the configuration index registers in alphabetical order.

Appendix B, “XT-Keyboard Interface,” explains how to implement the XT keyboard function in the ÉlanSC300 microcontroller.

## RELATED AMD PUBLICATIONS

### **PID No. Publication and Description**

- |       |  |
|-------|--|
| 18514 | <b><i>ÉlanSC300 Microcontroller Data Sheet</i></b><br>Describes the ÉlanSC300 microcontroller’s technical features, signal pins, internal controllers, and electrical specifications.  |
| 17028 | <b><i>3-Volt System Logic for Personal Computers Data Book</i></b><br>Includes the <i>Am386DXLV and Am386SXLV Microprocessors Technical Reference Manual</i> , which describes system management mode (SMM) and explains how to use the system management interrupt (SMI). |
| 19255 | <b><i>FusionE86<sup>SM</sup> Catalog</i></b><br>Provides information on tools that speed an E86 <sup>TM</sup> family embedded product to market. Includes products from expert suppliers of embedded development software.   |
| 12990 | <b><i>Fusion Newsletter</i></b><br>Contains quarterly updates on developments in the E86 <sup>TM</sup> family and features new Fusion Partner solutions.   |

To order publications, see “Documentation and Literature” on page iii.



# 1 POWER MANAGEMENT

---

In general, the purpose of today's power management is to reduce the wattage consumption of battery-powered computers in order to extend the useful amount of computing that can be done with a single battery charge. Power management is based on the following principal techniques:

- Employing process technologies (e.g., CMOS) that have intrinsically low power requirements
- Employing lower voltages whenever possible
- Continuously monitoring the activity of the computer and either turning off components that are not in use or reducing their clock speed

All these techniques are extensively employed in the ÉlanSC300 microcontroller, but only the third technique is under programmer control. Power management monitors all system activities (e.g., keyboard, screen, and disk events) and—based on the state of the system—determines in which operating mode the system should be running for best power conservation.

In addition, the ÉlanSC300 microcontroller can manage the power consumption of peripheral devices. This control can be integrated into the operating-mode mechanism, or it can be handled separately via the System Management Interrupt (SMI). The ÉlanSC300 microcontroller can be programmed so that various conditions, such as peripheral accesses, can cause an SMI to occur. An SMI causes the CPU to save the operating state of the processor and switch to a special interrupt service routine. This routine can then be used to turn on peripherals.

The ÉlanSC300 microcontroller's Power Management Unit (PMU) controls five Power-Management Control (PMC) pins and four Programmable General-Purpose I/O (PGP) pins that can be used by system designers to control different external peripherals.

External pins, such as the four low-battery interrupts ( $\overline{BL4}$ – $\overline{BL1}$ ), the AC power detect (ACIN), the external SMI (EXTSMI), and the  $\overline{SUS}/RES$  pin, can also cause SMIs or mode changes to occur. Certain power-management functions are disabled when the ACIN input is detected because it is assumed that the system is no longer using a battery.

This chapter contains sample microprocessor programs for setting up power-management functions for the ÉlanSC300 microcontroller. However, no attempt is made to cover all possible power-management situations, and the examples given are merely suggestions.

## 1.1 POWER MANAGEMENT UNIT

The primary design goal of the Power Management Unit (PMU) is to control the power of the entire system so as to eliminate or minimize the excess use of current, particularly when it is not needed at a specific time. The PMU uses the following techniques to conserve power:

- Slows down clocks when the system is not in active use
- Shuts off clocks to parts of the system which are idle
- Switches off power to parts of the system which are idle
- Reduces power use when batteries are low

An additional goal of a good PMU design is to make these functions as transparent to the user as possible, and to avoid any possibility of disastrous side effects, such as accidental loss of data. The ÉlanSC300 microcontroller's PMU includes the following principal components:

- **PMU State Machine** Defines certain levels of system activity and the allowable state transitions. Depending on the current level of activity registered by the PMU state, the system may, for example, run clocks at a high speed, a low speed, or turn them off. The current and previous PMU states may be read; in addition, the PMU may be forced into a specific state through a software command.
- **External Device-Control Interface** Allows the PMU to control external power switches to different devices. The on or off status of most of these devices may be specifically programmed for each of the available power-management states. In addition, dedicated logic enables the power to three specific devices to be automatically turned off after a specified time-out period during which no activity to the device has occurred. If one of these devices is accessed after power has been turned off, an SMI is automatically generated so that the I/O instruction can be retried after powering up the device.
- **Clock-Switching Logic** Synchronously switches different clock sources to the CPU clock, or switches off the CPU clock and phase-locked loops (PLLs).
- **Activity Monitors** Check for certain external or CPU events that indicate system activity. On receipt of an event, this logic causes the PMU state machine to switch to High-Speed PLL mode. By definition, activities function only while the CPUCLK signal is running.
- **State-Transition Timer** Defines the allowable periods of inactivity before PMU state changes occur.
- **Wake-Up Logic** Allows certain events to start the clocks and restart the on-board PLLs. Wake-ups are independent of whether the CPUCLK signal is on or off. Wake-ups force the PMU into High-Speed PLL mode.
- **NMI and SMI Control** Allows certain external, internally generated, or CPU events to generate a nonmaskable interrupt (NMI) or SMI to the CPU. If the CPU clock is not running when a triggering event occurs, this logic can cause the PMU to start the CPU clock to process the interrupt.
- **Battery-Management Logic** Includes four levels of battery-power handling. Certain levels can be programmed to generate SMIs or NMIs, slow the CPU clock, or force the system into Sleep or Suspend mode.

- **Suspend and Resume Pin Logic** Provides a user-operable method of forcing the PMU to enter Sleep mode or to wake up from Sleep, Suspend, or Off mode.
- **Auto Low-Speed Logic** Provides an option of slowing the CPU clock according to a programmable duty cycle while the system is in High-Speed PLL mode, providing additional power savings.
- **Other Power-Saving Features** Include additional power-saving functions which may be utilized at the system designer's discretion.

Each of these PMU components is discussed in greater detail later in this chapter.

### 1.1.1 Power-Management Modes

The ÉlanSC300 microcontroller's Power-Management Unit (PMU) provides the following power-management states or *modes*:

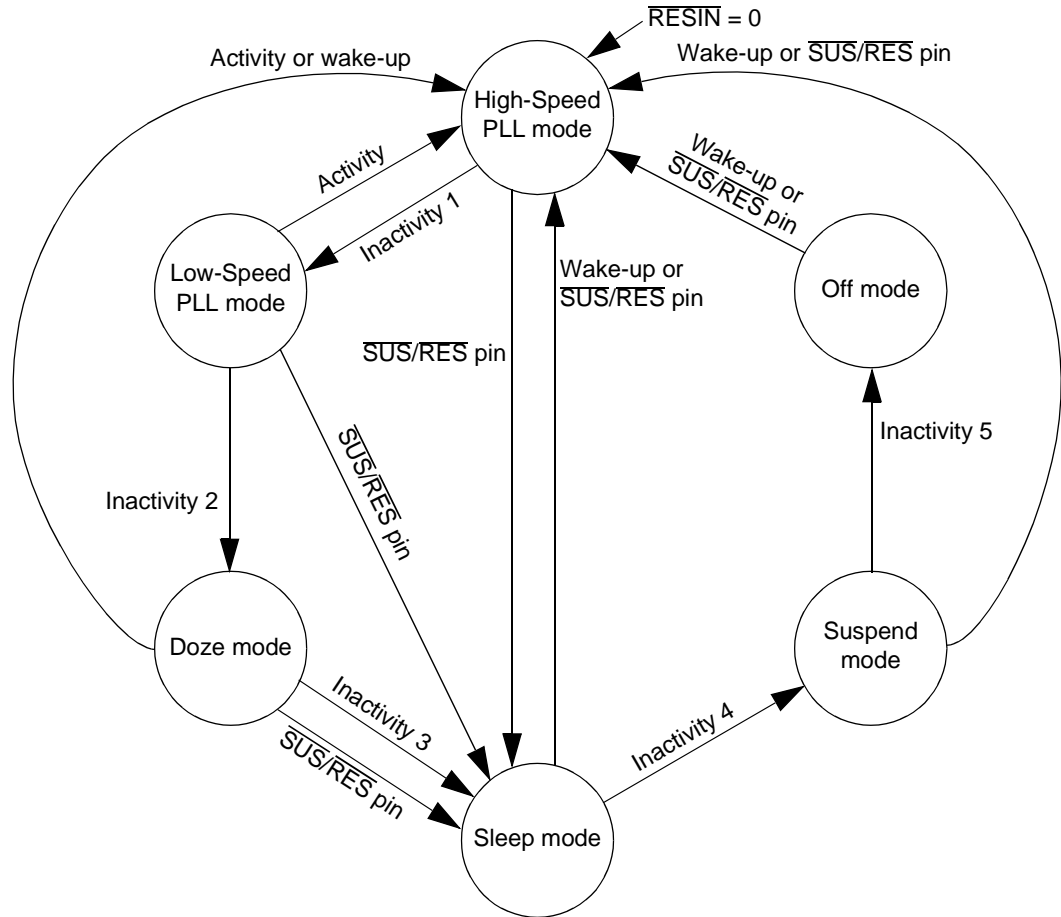
- High-Speed PLL
- Low-Speed PLL
- Doze
- Sleep
- Suspend
- Off

Each mode is defined by a different combination of CPU and peripheral operation. A given system may utilize all six states for the maximum granularity of PMU modes, or it may merge some of the states into as few as three effective PMU modes.

For a given PMU state, the controller can: specify different clock rates to various devices, turn off clocks, control external devices by way of programmable pins, and allow or disallow certain events to cause the unit to enter a subsequent PMU mode.

A state diagram for the PMU state machine is shown in Figure 1-1 on page 1-4. In addition, through the Software Mode Control register at Index 88h, the PMU can be forced into any mode except the Off mode by a software command.

Figure 1-1 PMU Operating-Mode Transitions



**Notes:**

1. This picture simplifies the function of the  $\overline{SUS/RES}$  pin. For more details, see "Suspend/Resume Pin Logic" on page 1-34.
2. ACIN Low and either  $\overline{BL2}$  or  $\overline{BL4}$  Low have the same effect as the  $\overline{SUS/RES}$  pin.

1.1.1.1

High-Speed PLL Mode

In High-Speed PLL mode, all system clocks run at their highest speeds. For the CPU, the high-speed PLL rate (supplied by the CLK2 signal) is software configurable to 40, 50, or 66 MHz, yielding internal CPU operation speeds (CPUCLK) of 20, 25, or 33 MHz, respectively. The low-speed PLL rate (also supplied by the CLK2 signal) is fixed at 18.432 MHz, yielding an internal CPU operation speed (CPUCLK) of 9.2 MHz.

The high-speed PLL rate only applies to certain types of CPU cycles. Normally, the CPU is clocked at the low-speed PLL rate, even in High-Speed PLL mode. For CPU DRAM, local-bus, fast-ROM, and idle cycles, the clock is dynamically switched to run at the high-speed PLL rate.

Any activity (or wake-up) defined by the software will cause the PMU to select High-Speed PLL mode. In this mode, power conservation takes a back seat to CPU processing power. In High-Speed PLL mode, liquid crystal display (LCD) logic and contrast voltages ( $\overline{LVDD/LVEE}$ ) are enabled.

Power-Management Control (PMC) pins can be used to control power to peripheral devices on a per-mode basis. Software can restore power to any peripherals whose power (controlled by its associated PMC pin state) was removed by a previous transition to a lower-power mode.

#### 1.1.1.2 Low-Speed PLL Mode

Low-Speed PLL mode is the first level of power conservation. Low-Speed PLL mode is entered after a specified elapsed time with no activity, programmed using the High-Speed to Low-Speed Mode Timer register at Index 83h. In this PMU mode, the CPU, DMA, and other internal system clocks run at reduced rates. The low-speed PLL clock, whose rate is always fixed at 18.432 MHz, is sent through a programmable divider. The minimum divisor is 2. This yields a maximum programmable CLK2 rate of 9.216 MHz, which results in a maximum internal operation speed (CPUCLK) of 4.608 MHz. PLL divisors of 2, 4, 8, and 16 can be selected for dividing the low-speed PLL clock in Low-Speed PLL mode. No dynamic switching of CLK2 to the high-speed PLL rate is done in this mode. All other clocks and peripherals run at full speed.

Power-management software may optionally shut off the high-speed PLL. Depending on the frequency of this PLL, up to 750  $\mu$ A may be saved by this action. If this is done, a PLL startup delay of 256 ms (programmed in the Clock Control register at Index 8Fh) must elapse before High-Speed PLL mode can be reentered.

PMC pins may be used to control power to peripheral devices on a per-mode basis. Software may restore power to any peripherals whose power (controlled by their associated PMC pin states) was removed by a previous transition to a lower-power mode. In Low-Speed PLL mode, LCD logic and contrast voltages (LVDD/LVEE) are enabled.

#### 1.1.1.3 Doze Mode

Doze mode is the second level of power conservation. The CPU, system, and DMA clocks, and the high-speed PLL are stopped. This mode is entered after a programmed time without activity has elapsed. For details, see “Low-Speed to Doze Timer Register (Index 84h)” on page 5-57.

By default, the CPU clock is stopped in Doze mode, along with the DMA clock and internal system clock (see Table 1-1 on page 1-6). The video, keyboard, UART, and 8254 timer clocks are driven by the low-speed PLL, which is enabled by default. By allowing these clocks to run, it is possible for timer and keyboard interrupts to be generated.

Table 1-1 PMU Clock Speeds

Mode	High-Speed CPUCLK	Low-Speed CPUCLK	Video Clock	DMA Clock	SYCLK	8254 Clock (Timer)	16450 Clock (UART)
High-Speed PLL	33/25/20 MHz	9.2 MHz	14.336 MHz	4.6 MHz	9.2 MHz	1.19 MHz	1.8432 MHz
Low-Speed PLL	9.2 MHz	4.608/2.304/1.152/0.56 MHz	14.336 MHz	2.3/1.2/0.58/0.29 MHz	9.2 MHz	1.19 MHz	1.8432 MHz
Doze	DC <sup>1</sup>	DC <sup>1</sup>	14.3 MHz/DC <sup>2</sup>	DC <sup>1</sup>	9.2 MHz/DC <sup>2</sup>	1.19 MHz/DC <sup>2</sup>	1.8 MHz/DC <sup>2</sup>
Sleep	DC	9.2 MHz/DC <sup>4</sup>	14.3 MHz/DC <sup>2</sup>	4.6 MHz/DC <sup>4</sup>	DC	1.19 MHz/DC <sup>2</sup>	1.8 MHz/DC <sup>2</sup>
Suspend	DC	9.2 MHz/DC <sup>4</sup>	14.3 MHz/DC <sup>2</sup>	4.6 MHz/DC <sup>4</sup>	DC	1.19 MHz/DC <sup>2</sup>	1.8 MHz/DC <sup>2</sup>
Off	DC	9.2 MHz/DC <sup>4</sup>	14.3 MHz/DC <sup>3</sup>	4.6 MHz/DC <sup>4</sup>	DC	1.19 MHz/DC <sup>3</sup>	1.8 MHz/DC <sup>3</sup>

**Notes:**

The DMA clock can be stopped except during DMA transfers. The Function Enable 1 register at Index B0h controls this function. The CPU clock speed in Low-Speed PLL mode is selectable. For information, see “PMU Control 3 Register (Index ADh)” on page 5-76.

1. Can be programmed to run intermittently (on the IRQ0 pin) at 9.2 MHz.
2. Is a programmable option, but not on a per-clock basis—all clocks with this note are controlled by a single on/off select for that PMU mode.
3. Is a programmable option—reflects the setting in Suspend mode.
4. Can be programmed to run at 9.2 MHz during temporary-on NMI or SMI handlers.

Through an option enabled by setting bit 3 of the MMSB Control register at Index 74h, the PMU can be programmed to periodically start the low-speed CPU clock when the IRQ0 pin (generated by the 8254 timer) is asserted. By default, the clock runs only while IRQ0 is active or the ISR0 bit is High, stopping on the next refresh after this condition is no longer valid. In this case, the CPU clock runs at 9.2 MHz. The 8259 interrupt controller must be programmed to unmask IRQ0. By setting bit 0 of the PMU Control 2 register at Index AFh, the run time may be extended for 64 refresh cycles after ISR0 goes Low.

By setting bit 7 of the Power Control 1 register at Index 80h, the low-speed and video PLLs can be shut down in Doze mode. Shutting down the low-speed PLL also shuts down the 8254 timer clock, the UART clock, and the keyboard clock; therefore, the IRQ0 wake-up cannot be used in this instance. The high-speed PLL is always shut down in this mode.

The PMC pins may be programmed to a specific state for this mode. The LCD-Panel  $V_{DD}$  Voltage-Control ( $\overline{LVDD}$ ) switch is active. The LCD-Panel  $V_{EE}$  Voltage-Control ( $\overline{LVEE}$ ) switch is active if the video PLL is not shut down. An NMI or SMI may be generated upon entering Doze mode. The CPUCLK signal runs at 9.2 MHz during the NMI or SMI handler.



## 1.1.1.4

## Sleep Mode

Sleep mode is the third level of power conservation. In addition to the clocks disabled in Doze mode, the keyboard clock (external SYSCLK signal) is disabled, regardless of whether the low-speed PLL is enabled. Sleep mode is entered after a programmed time without activity has elapsed. For details, see “Doze to Sleep Timer Register (Index 85h)” on page 5-57.

In this mode, the CPU, system, and DMA clocks are stopped and cannot be restarted unless one of the following events occurs:

- An SMI or NMI on  $\overline{BL1}$  or  $\overline{BL3}$  (when enabled)
- An SMI or NMI generated on a change to Suspend mode
- A wake-up event causes an exit from Sleep mode to High-Speed PLL mode

In the first two cases, the clock runs only during the SMI or NMI routine and then stops again. The keyboard clock is also shut down and can only be restarted by waking up to High-Speed PLL mode. The high-speed PLL is always shut down in Sleep mode.

By setting bit 3 of the Power Control 2 register at Index 81h, the low-speed and video PLLs also may be shut down in this mode. In this case, the video and low-speed PLLs are restarted before responding to an SMI or NMI or changing to High-Speed PLL mode. When changing to High-Speed PLL mode, the high-speed PLL is also restarted. Note that the low-speed PLL is divided to generate the 8254 timer clock.

The PMC pins may be programmed to a specific state for Sleep mode. The  $\overline{LVEE}$  switch is inactive, but  $\overline{LVDD}$  remains active. An NMI or SMI may be generated upon entering Sleep mode, in which case the handler runs at 9.2 MHz.

## 1.1.1.5

## Suspend Mode

With regard to the clocks and PLLs, Suspend mode has the same functionality as Sleep mode. But bit 7 of the Power Control 2 register at Index 81h enables shutdown of the video and low-speed PLLs in Suspend mode. The distinction between Suspend and Sleep mode is in the way the external Power-Management Control (PMC) pins behave and may be programmed to behave. The PMC pins may be programmed to a specific state for this mode. Both  $\overline{LVEE}$  and  $\overline{LVDD}$  are inactive in this mode. An NMI or SMI may be generated upon entering Suspend mode, in which case the handler runs at 9.2 MHz.

## 1.1.1.6

## Off Mode

Off is a powered-down mode in which the Programmable General-Purpose 2 and 3 (PGP2 and PGP3) pins are set to a predefined state, and memory refresh may be disabled. The state of the PGP pins is determined by the General-Purpose I/O 2 and 3 registers at indexes 94h and 95h, and the General-Purpose I/O Control register at Index 91h.

The system cannot be programmed to enter Off mode directly. The only method of Off mode entry is by expiration of the Suspend to Off Mode Timer register at Index 87h. When this happens, the PMU state machine is left in Suspend mode, and an internal, nonreadable flip-flop is set, indicating Off mode. An NMI or SMI may be generated upon entering Off mode.

Refresh may be programmed to be disabled when the PMU is in Off mode. Setting bit 7 of the Memory Configuration 2 register at Index B9h causes the  $\overline{RAS}$  and  $\overline{CAS}$  outputs to be driven Low when the PMU is in Off mode. The system logic should power off the DRAM in

this mode, or the Low  $\overline{\text{RAS}}$  and  $\overline{\text{CAS}}$  outputs may keep the row buffers enabled, thus drawing additional power from the DRAM devices. DRAM content is invalid when exiting from Off mode when the disable-refresh feature is being used.

### 1.1.2 PMU Operating-Mode Transitions

Figure 1-1 on page 1-4 shows the ÉlanSC300 microcontroller's six operating modes and the transitions that can occur between them. In this diagram, the term *inactivity*, followed by a number from 1 to 5, refers to different timer intervals of inactivity, which are set by programming the Mode Timer registers at indexes 83–87h. The term *activity* refers to the programmer-specified computing activities set in the following registers:

- Activity Mask 1 and 2 registers at indexes 75–76h
- I/O Activity Address 0 and 1 registers at indexes 8C–8Dh
- Memory Write Activity Lower and Upper Boundary registers at indexes 9A–9Bh
- PMU Control 1–3 registers at indexes A7h, AFh, and ADh

For more information, see the programming examples later in this chapter.) The  $\overline{\text{SUS/RES}}$  pin is an external pin that can be triggered to cause a transition between modes.

Each power-management mode is characterized by a different clock-frequency pattern—the CPU, system bus, and many peripheral devices have their own clocks. Several clock-switching permutations are possible through the use of the PMU modes. The PLLs act as sources for other clocks as follows:

- **High-Speed PLL** Generates the high-speed CPU/memory clock only. It can be programmed to run at either 40, 50, or 66 MHz, yielding an internal CPU operation frequency of 20, 25, or 33 MHz, respectively. The PMU state machine has controls that can disable this clock to prevent it from being used or, in addition, turn off the PLL.
- **Low-Speed PLL** Is divided to generate the following clocks:
  - Low-speed CPU/internal system clock
  - 8254 timer clock
  - 8250 UART clock
  - 9.2-MHz keyboard-controller clock/external SYSCLK

The PMU state machine has controls that can disable some of these clocks to prevent them from being used or, in addition, turn off the PLL entirely.

- **Video PLL** Generates the clock for the ÉlanSC300 microcontroller's internal video controller. The disable control for this clock is shared with the low-speed PLL. If the low-speed PLL is turned off, this PLL is also turned off.

As shown in Table 1-1 on page 1-6, in High-Speed PLL mode, the CPUCLK signal can be programmed to run at 33 MHz, 25 MHz, 20 MHz, or 9.2 MHz during DRAM, local-bus, fast-ROM, and idle cycles. In all other cases, the CPUCLK signal runs at 9.2 MHz.

In Low-Speed PLL mode, the CPUCLK signal can be programmed to run at 4.6 MHz, 2.27 MHz, 1.13 MHz, or 0.568 MHz. The CPUCLK signal cannot run at 9.2 MHz in Low-Speed PLL mode, and it always runs at the same speed regardless of the type of cycle.

### 1.1.3 PMU Clock Sources

The ÉlanSC300 microcontroller's PMU uses the 32-kHz clock to derive its internal timing. This clock runs off the ÉlanSC300 microcontroller's internal oscillator, which cannot be disabled. Many events are synchronized with the internal refresh signal, which by default is derived from the 32-kHz clock. If the ÉlanSC300 microcontroller's PMU is being used in a system design, the refresh clock must not be set to Timer Channel 1 because the timer is disabled in some PMU modes.

The remainder of this section describes the functionality of the system during each of the PMU states. Later sections discuss the different ways the PMU can be caused to enter these states.

### 1.1.4 Reading the PMU Mode

The current PMU mode can be read from the CPU Status 1 register at Index A4h. If the PMU is in Off mode, this register indicates Suspend mode. PMU mode changes always take effect on the next refresh after the mode change was registered. The mode that is read from the CPU Status 1 register at Index A4h is one refresh delay in advance of the internal signals that actually execute the functions of the PMU mode. Therefore, if it is necessary to know the exact mode of the PMU at a specific time (e.g., for the purpose of determining the state of the PMC signals), the software must read the CPU Status 1 register on two successive refreshes and verify that the mode has not changed.

### 1.1.5 Merging of PMU Modes

Although six PMU modes are defined, the system designer may reduce the effective number of PMU modes by defining identical functions for some of the modes. For example, assuming that Full-ISA or Local-Bus modes are being used, a three-mode system that effectively merges the Doze, Sleep, Suspend, and Off modes can be achieved. To merge the Doze, Sleep, Suspend, and Off modes, use the following procedure:

1. Set bit 7 of the Power Control 1 register at Index 80h and bits 3 and 7 of the Power Control 2 register at Index 81h to disable the low-speed and video PLLs in Doze, Sleep, and Suspend modes.
2. Set all the PMC bits to the same value for Doze, Sleep, and Suspend modes.
3. Do not enable the PGP2 and PGP3 pins to change in Off mode.

The net effect of this procedure is to create a three-mode system, effectively consisting of High-Speed PLL, Low-Speed PLL, and Suspend modes, where Doze, Sleep, Suspend, and Off modes have been merged into a single new *pseudo-Suspend* mode. The one exception to this scenario is that the  $\overline{\text{LVDD}}$  pin is hardwired to be deasserted only in Suspend and Off modes.  $\overline{\text{VVEE}}$  will deassert in the Sleep, Suspend, and Off modes or when the video PLL is turned off.

### 1.1.6 Programming Example: Power-Management Setup

Neglecting peripheral control for the moment, the first step in setting up power-management for a system is to define what constitutes *activity*. The following events constitute activity:

- DMA requests and interrupt requests (DRQ, IRQ)
- Keyboard, LPT, COM, and programmable I/O port accesses
- MMS, video memory, and programmable memory range accesses
- Hard drive and floppy disk drive accesses
- AC adapter active (rising edge of ACIN)

The second step is to define the absence of activity as *inactivity*. The next step is to define the time intervals of inactivity to be allowed before the system automatically shifts itself to lower levels of power consumption. For this example, the inactivity time intervals used are shown in Table 1-2 on page 1-10.

Table 1-2 Inactivity States and Transition Intervals

State	From this Mode	To this Mode	Interval
Inactivity 1	High-Speed PLL	Low-Speed PLL	1/16 s
Inactivity 2	Low-Speed PLL	Doze	10 s
Inactivity 3	Doze	Sleep	10 min
Inactivity 4	Sleep	Suspend	10 s
Inactivity 5	Suspend	Off	1 hr

The settings in Table 1-3 on page 1-11 implement these steps. In this case, activity is considered to include most conventional interrupts and peripheral accesses. However, accesses to special I/O port ranges or memory ranges (as might be required to support a PCMCIA socket, for example) are not considered activity. The inactivity timings shown are fairly conventional for notebook and palmtop computers.

Table 1-3 Power-Management Setup

Instruction	Ports	Index and Data	Comment
IOW IOW	022h 023h	75h 0100 0000	Activity: DMA requests, IRQ15–IRQ2 (except IRQ8 and IRQ4–IRQ3), keyboard, and MMS Inactivity: AC adapter
IOW IOW	022h 023h	76h 1011 0000	Activity: Hard drive, floppy disk drive, LPT, COM, and video I/O accesses Inactivity: Programmable I/O ports and memory ranges
IOW IOW	022h 023h	08h 1110 0011	Activity: IRQ8 and IRQ4–IRQ3 (these wake up the system from Sleep, Suspend, and Off modes)
IOW IOW	022h 023h	AFh 1100 0000	Set the granularity of the Low-Speed to Doze Mode Timer register to 1/4 s and the High-Speed to Low-Speed Mode Timer register to 1/16 s, respectively.
IOW IOW	022h 023h	83h 0000 0001	Set the High-Speed to Low-Speed Mode Timer register to 1/16 s (High-Speed to Low-Speed mode transition).
IOW IOW	022h 023h	84h 0010 1000	Set the Low-Speed to Doze Mode Timer register to 10 s (Low-Speed to Doze mode transition).
IOW IOW	022h 023h	85h 1001 0110	Set the Doze to Sleep Mode Timer register to 10 min (Doze to Sleep mode transition).
IOW IOW	022h 023h	86h 1010 0000	Set the Sleep to Suspend Mode Timer register to 10 s (Sleep to Suspend mode transition).
IOW IOW	022h 023h	87h 0011 1000	Set the Suspend to Off Mode Timer register to 59 min, 44 s [1 hr] (Suspend to Off mode transition).

**Note:**

For this and subsequent examples, the index registers cannot be programmed directly, but must be accessed by writing the location of the Index register to address 22h and the Data register to address 23h.

For example, the first I/O access in the above example can either be implemented using 8-bit accesses:

```
mov al,75h
out 22h,al
mov al,40h
out 23h,al
```

or 16-bit accesses:

```
mov ax,4075
out 22h,ax
```

### 1.1.7 Programming Example: Peripheral-Device Power

The simplest way to control peripheral devices is to integrate their control into the above mode-based system. The ÉlanSC300 microcontroller allows five external Power-Management Control (PMC) pins to be activated in such a way that they can indicate to external devices the mode of the system. Simple logic can be used to disable the peripheral when the system is in specific modes.

As a concrete example, consider the previous example, which has several peripheral devices (floppy disk drive, LCD panel, and so forth). Suppose the following peripheral-power-management scheme is to be implemented:

- 10 s of inactivity—Turn off the LCD backlighting and the floppy disk drive
- 10 min of inactivity—Turn off the LCD panel and the serial port transceiver

This scheme can be implemented by using Doze mode to signal the backlighting and the floppy disk drive, and Sleep or Suspend mode to signal the LCD panel and the transceiver. The PMC pins can be programmed to provide this control through the Power Control 1–4 registers at indexes 80–81h and AB–ACh.

The settings shown in Table 1-4 on page 1-12 provide the following scheme:

- PMC1 is activated when the system is in High-Speed PLL or Low-Speed PLL mode
- PMC2 is activated when the system is in Doze mode
- PMC3 is activated when the system is in Sleep mode
- PMC4 is activated when the system is in Suspend mode

The designer can use the PMC4–PMC2 pins to disable the LCD backlighting, the floppy disk drive, the LCD panel, and the transceiver.

Table 1-4 Power-Management Control Pin Settings

Instruction	Ports	Index and Data	Comment
IOW IOW	022h 023h	ACh 0001 0000	Activate PMC1 when in High-Speed PLL or Low-Speed PLL mode.
IOW IOW	022h 023h	80h 0100 0000	Activate PMC2 when in Doze mode.
IOW IOW	022h 023h	ABh 1000 1011	Activate PMC3 when in Sleep mode, and activate PMC4 when in Suspend mode.

A more elaborate system permits the control of each peripheral on an individual basis. The settings shown in Table 1-5 on page 1-12 extend the PMC initialization example to show how to power down a PCMCIA device (such as a wireless communicator) after 8 s. The following example shows how the ÉlanSC300 microcontroller can be programmed to make the PMC1 pin change state after 8 s of system inactivity. The designer can use this signal to turn off PCMCIA power.

Table 1-5 PIO Timeout Settings

Instruction	Ports	Index and Data	Comment
IOW IOW	022h 023h	45h 1100 000	Set PIO address to the window at port 300h.
IOW IOW	022h 023h	46h 0100 0110	Set the I/O window size to 8 bytes, and the timeout to 8 s.

If the time-out period expires, the PMC1 pin is pulled Low, which causes PCMCIA power to be removed. If the program then attempts to access the PCMCIA card via I/O addresses in the range 300–307h, then an SMI is generated. This procedure allows the program to turn on PCMCIA power before reissuing the access. The settings shown in Table 1-6 on page 1-13 enable SMI generation for PIO accesses to that address range.

Table 1-6 SMI-Generation Settings for PIO Accesses

Instruction	Ports	Index and Data	Comment
IOW IOW	022h 023h	A9h 1100 xxxx	Set the SMI Memory-Mapping System (MMS) (bits 23–22) to 00; enable SMI MMS.
IOW IOW	022h 023h	AAh 1000 0000	Set the SMI MMS to 200000h.
IOW IOW	022h 023h	43h 0000 0000	Reset the SMI Status register.
IOW IOW	022h 023h	41h 0000 0100	Enable SMI for PIO accesses.

**Note:**

*In this and subsequent examples, the notation xxxx refers to a field of bits whose value must be preserved. That is, the programmer must execute an I/O-read–modify–I/O-write cycle to ensure that the current contents of this field are not changed.*

An SMI causes the CPU to store its internal state at location 060000h, which in this example is mapped by the Memory Mapping System to 200000h. The CPU begins executing at the reset vector, and the code executes the SMI handler after it checks the SMI flag.

By checking the SMI Status register at Index 43h, the SMI handler determines that a PIO access caused the SMI. Then the handler can turn on PCMCIA power by setting bit 4 of the Power Control 4 register at Index ACh, which causes the PMC1 pin to be reset, thus restoring power to the device.

## 1.2 EXTERNAL-DEVICE CONTROL INTERFACE

The external device-control interface includes the following pins:

- Power-Management Control (PMC4–PMC0)
- Programmable General-Purpose (PGP3–PGP2)
- LCD-Panel Voltage-Control ( $\overline{\text{LVDD}}$  and  $\overline{\text{LVEE}}$ )
- Latched Power ( $\overline{\text{LPH}}$ )

### 1.2.1 Power-Management Control Pins

The Power-Management Control (PMC4–PMC0) pins may be used for either or both of the following purposes:

- General-purpose control of external devices in conjunction with the PMU state machine
- Timer-controlled shutdown of a floppy disk drive (addresses 3F0–3F7h), hard drive (addresses 1F0–1F7h), or user-specified I/O device in conjunction with the ÉlanSC300 microcontroller’s SMI interface.

#### 1.2.1.1 General-Purpose Control Using the PMU State Machine

When using the PMC pins for general-purpose control, PMC4 and PMC2–PMC0 are non-inverting and drive a 0 at reset. PMC3 is inverting and will drive a 1 at reset. Each pin may be programmed to drive a unique state in High-Speed PLL, Low-Speed PLL, Doze, Sleep, and Suspend modes. Pin state-switching occurs on the next refresh cycle after a PMU state change. The signals are not synchronous and may glitch when changing to High-Speed PLL mode. Internal gray encoding prevents glitching when the PMU states are cycling down sequentially from High-Speed PLL mode. Table 1-7 on page 1-14 summarizes the functionality of these pins.

Table 1-7 PMC Pin Functionality

PMC Pin No.	Control Register	Output Sense	SMI Timer Function
0	Index ACh, bits 3–0	Noninverting	Floppy disk drive (3F0–3F7h)
1	Index ACh, bits 7–4	Noninverting	Programmable I/O address
2	Index 80h, bits 6 and 2 Index 81h, bits 6 and 2	Noninverting	(None)
3	Index ABh, bits 3–0	Inverting	(None)
4	Index ABh, bits 7–4	Noninverting	Hard drive (1F0–1F7h)

#### 1.2.1.2 Timer-Controlled Shutdown Using the SMI Interface

When using PMC4 or PMC1–PMC0 for timer-controlled shutdown of devices, the SMI timer logic must be enabled through the SMI Enable register at Index 41h. In addition, the bits in the General-Purpose I/O 0–1 and 4 registers that correspond to the modes in which the devices normally run should be programmed to a 1. A PMC output pin goes High to enable a device. The pin goes Low to disable a device under any of the following conditions:



- The SMI device timer expires.
- The PMU enters Sleep, Suspend, or Off mode.
- The PMU enters another mode for which the device is programmed to turn off.

It is also important to understand that an expiring SMI device timer clears all four of the PMC control bits for that device by a short pulse generated when the timer expires. In addition, the control-register bits for that device are held in reset when the PMU is in Sleep, Suspend, or Off mode. These conditions only apply to PMC pins that have an SMI device timer enabled by the SMI Enable register at Index 41h. For more information, see “Accesses to Powered-Down Device SMI” on page 1-29.

### 1.2.2 Programmable General-Purpose Pins 2 and 3

Through an option enabled via the General-Purpose I/O 2 and 3 registers at indexes 94–95h and the General-Purpose I/O Control register at Index 91h, the PGP3–PGP2 pins can be enabled to switch from High to Low when the PMU enters Off mode. This option can be enabled as follows:

- **PGP2** Clear bits 4 and 5 of the General-Purpose I/O Control register at Index 91h, and set bit 7 of the General-Purpose I/O 2 register at Index 94h.
- **PGP3** Clear bits 6 and 7 of the General-Purpose I/O Control register, and set bit 7 of the General-Purpose I/O 3 register at Index 95h.

This setup causes the PGP pins to default to 1. When the PMU enters Off mode, the PGP2 or PGP3 pin is driven by the inverse of bit 7 of the General-Purpose I/O 2 register or the General-Purpose I/O 3 register, respectively. Because the PMU timer delay to Off mode can be set to as long as 256 min, this feature can be used to turn off a device after a prolonged period of inactivity.

### 1.2.3 LCD-Panel Voltage-Control Pins

The LCD-Panel Voltage-Control ( $\overline{\text{LVDD}}$  and  $\overline{\text{LVEE}}$ ) pins are designed to sequence the power to an LCD display as follows:

- $\overline{\text{LVDD}}$  controls the panel logic voltage
- $\overline{\text{LVEE}}$  controls the contrast voltage

Unlike the PMC pins, these control pins are hardwired for a specific function and are not programmable. A 0 on one of these pins indicates that power should be switched on. When the external reset pin,  $\overline{\text{RESIN}}$ , is held active,  $\overline{\text{LVDD}}$  and  $\overline{\text{LVEE}}$  are forced to a 1 (power off). One refresh cycle after  $\overline{\text{RESIN}}$  is driven inactive,  $\overline{\text{LVDD}}$  changes to 0 (power on).  $\overline{\text{LVEE}}$  changes to 0 on the next refresh after  $\overline{\text{LVDD}}$  changes.

When the PMU enters Sleep mode, or when the video PLL is turned off in Doze mode,  $\overline{\text{LVEE}}$  changes to 1, disconnecting the negative voltage.  $\overline{\text{LVDD}}$  will remain active until the PMU reaches Suspend mode. When waking up from Suspend mode (i.e., entering High-Speed PLL mode),  $\overline{\text{LVDD}}$  will precede  $\overline{\text{LVEE}}$  by one refresh cycle.

The LCD-panel data and control signals are all forced to 0 in the PMU modes which are programmed to disable the low-speed and video PLLs through bits 3 and 7 of the Power Control 1 and 2 registers at indexes 80h and 81h.

**Note:** When using  $\overline{LVDD}$  and  $\overline{LVEE}$  to control the voltage on an LCD panel, do not allow the BIOS to force the PMU directly into Suspend mode by writing to the Software Mode Control register at Index 88h. Doing so causes  $\overline{LVDD}$  and  $\overline{LVEE}$  to switch off simultaneously, which violates the correct power sequencing for most LCD panels. If a forced shutdown is desired, force the PMU into Sleep mode instead, and allow the PMU timer to sequence the PMU into Suspend mode.

Some LCD panels may require a maximum time between switching off  $\overline{LVEE}$  and  $\overline{LVDD}$ . In this case, the Sleep to Suspend Mode Timer register at Index 86h should be programmed to fall within this time.

#### 1.2.4 Latched Power Pin

The Latched Power ( $\overline{LPH}$ ) pin can be used to indicate a low battery. The default state of  $\overline{LPH}$  is 0. When enabled by setting bit 7 of the MMSB Control register at Index 74h, a 0 on the Battery Level 4 ( $\overline{BL4}$ ) input pin causes  $\overline{LPH}$  to drive a 1, provided that the ACIN (AC Input Active) pin is also 0.

### 1.3 CLOCK-SWITCHING LOGIC

The ÉlanSC300 microcontroller's clock-switching logic handles the task of switching clock speeds as directed by the PMU or other input, and sequencing the shutdown and startup of the clocks and PLLs.

#### 1.3.1 CPU/Memory Clock Switching

The CPU clock-switching circuit delivers a signal that switches cleanly between the high-speed PLL clock and a low-speed clock source in High-Speed PLL mode. The low-speed clock source switches cleanly between the low-speed PLL clock (9.2 MHz) and a selectable slow clock on a PMU state transition (see Table 1-1 on page 1-6).

The high-speed PLL clock is used only under narrowly defined conditions. The high-speed clock must be enabled by setting bit 6 of the I/O Wait State register at Index 61h. Doing this enables the use of the high-speed clock after the next refresh cycle if the high-speed PLL is already started.

In addition, the PMU must be in High-Speed PLL mode, and the current bus cycle must be one of the following types of cycles:

- CPU idle
- Local DRAM
- Fast ROM
- Local bus

Use of the high-speed clock is disallowed when all of the following conditions are true:

- $\overline{BL1}$  is Low
- Bit 5 of the PMU Control 2 register at Index AFh has been set to enable this feature
- ACIN is Low

If the auto low-speed logic is enabled, use of the high-speed clock is disallowed periodically to conserve power. In cases where the use of the high-speed PLL clock is disallowed, the low-speed PLL's 9.2-MHz CPU clock is used.

When the PMU is in Low-Speed PLL mode, the CPU clock is generated from the low-speed clock source. This is a programmable divider chain controlled by bits 1–0 of the PMU Control 3 register at Index ADh that provides a clock frequency of 9.216 MHz that is divided by 2, 4, 6, or 8. These bits should be changed only when the PMU is not in Low-Speed PLL mode.

When a temporary-on condition occurs while the PMU is in Doze, Sleep, Suspend, or Off mode—and the CPU/memory clock is enabled to run—the 9.2-MHz CPU clock is used.

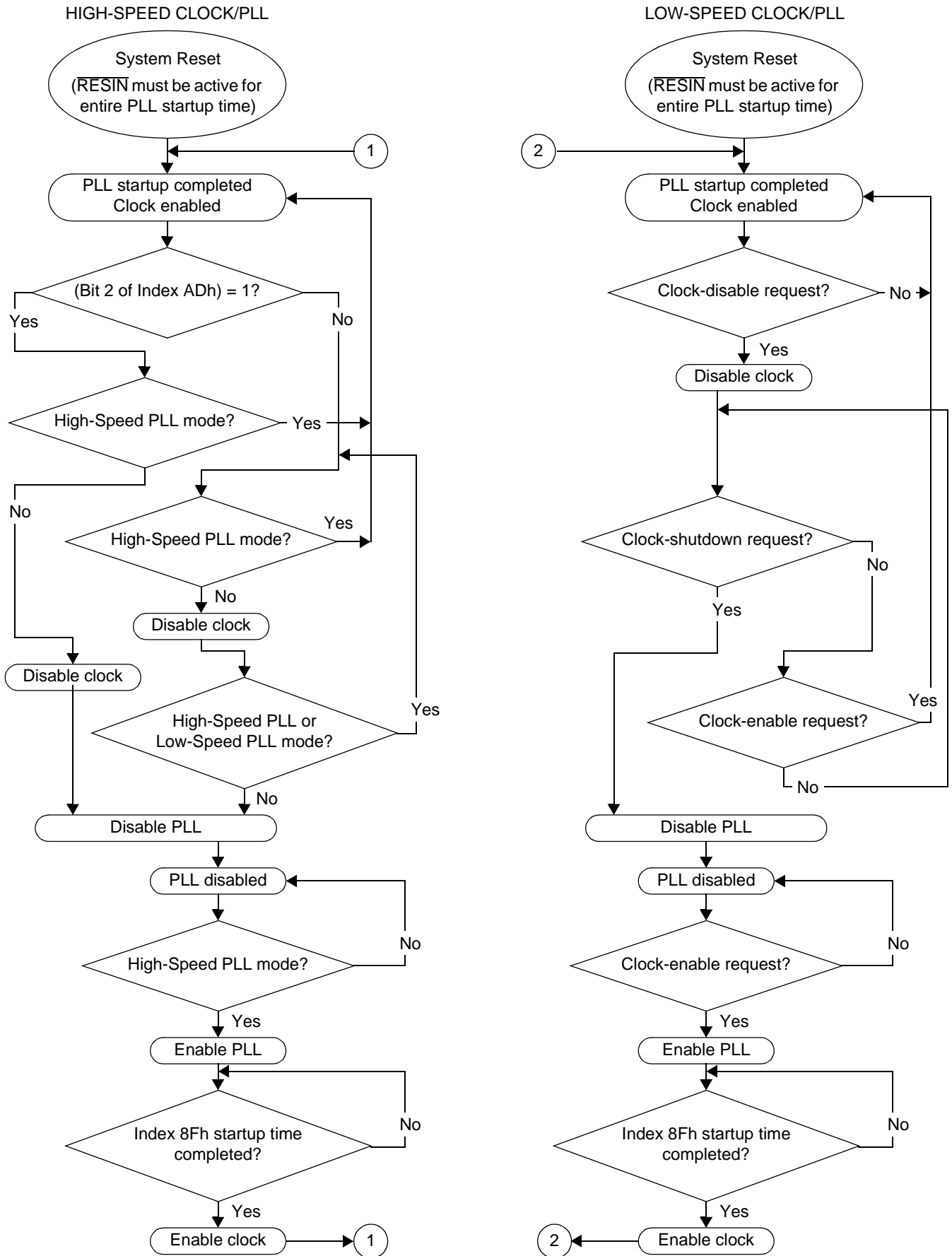
The high-speed PLL clock frequency may be selected by writing to bits 4–3 of the Function Enable 2 register at Index B1h. These bits should not be changed when the high-speed clock is enabled.

### 1.3.2 Clock Startup and Shutdown Logic

The clock startup and shutdown logic provides a mechanism for properly coordinating the activation of the ÉlanSC300 microcontroller's PLLs with the CPU's clock-enabling logic. The flowchart in Figure 1-2 on page 1-18 illustrates the logic flow. For the high-speed clock, there is a choice of turning off the PLL in either Low-Speed PLL mode or Doze mode, selectable through bit 2 of the PMU Control 3 register at Index ADh. For the low-speed clock, there are a number of options with regard to when the PLL is turned off. These options are controlled using bits 7 and 3 of the Power Control 1 and 2 registers at indexes 80–81h. The PLL startup time must be programmed in the Clock Control register at Index 8Fh to allow sufficient startup time for the PLL before the clocks are engaged. It is recommended that the startup time be programmed to at least 256 ms when starting from a PMU mode where the low-speed PLL is disabled. A startup time of 128 ms can be used if the low-speed PLL will never be disabled.

If the PMU is returning to High-Speed PLL mode from a mode where the high-speed PLL was disabled but the low-speed PLL was enabled, the CPU clock will begin operating at 9.2 MHz immediately. After the PLL startup time expires, the CPU clock switches to the high-speed PLL frequency if enabled to do so via bit 6 of the I/O Wait State register at Index 61h.

Figure 1-2 PLL Control Flowchart



## 1.4 ACTIVITY MONITORS

The activity-monitor logic keeps track of events that indicate that the CPU or peripherals are in demand. Examples of such events are I/O or memory decodes to certain addresses, DMA requests, interrupts, and changes in status signals. When an activity event is recognized, the PMU state machine immediately switches to High-Speed PLL mode on the next refresh cycle. The exception to this rule is that activities are not recognized during execution of SMLs or NMLs. All PMU activities are edge detected.

Activities that are referred to in Table 1-3 on page 1-11 can be classified as CPU related and non-CPU related. The CPU-related activities that the ÉlanSC300 microcontroller's PMU is able to count as events are selected via the Activity Mask 1 and 2 registers at indexes 75–76h, with status read from the Activity Status 1 and 2 registers at indexes A0–A1h.

The non-CPU-related activities that the PMU is able to count as events are selected using the Resume Mask register at Index 08h, with status read from the Resume Status register at Index 09h. Non-CPU-related activities are also classified as wake-up events. Note that it also is possible for some of the CPU-related activities to wake up the PMU. Both types of wake-ups are discussed in “Wake-Up Logic” on page 1-23.

Detected activities in the Activity Status 1 and 2 registers at indexes A0–A1h and the Resume Status register at Index 09h are indicated by a 1 in the appropriate bit. After reading one of these registers, software should write any value to clear the register.

Figure 1-3 on page 1-20 shows how activities play a part in the power-management flow.

Figure 1-3 State Transition Flowchart

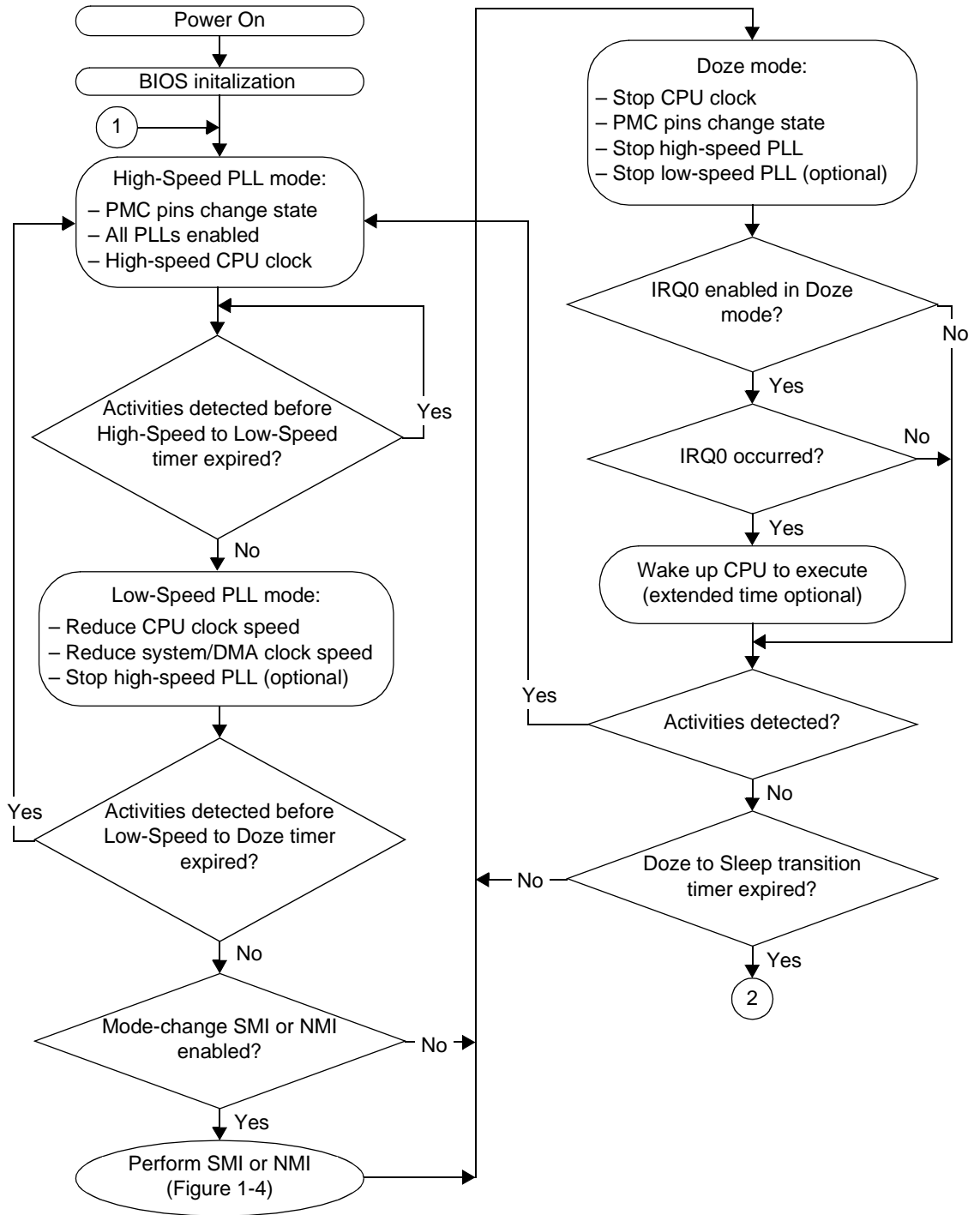
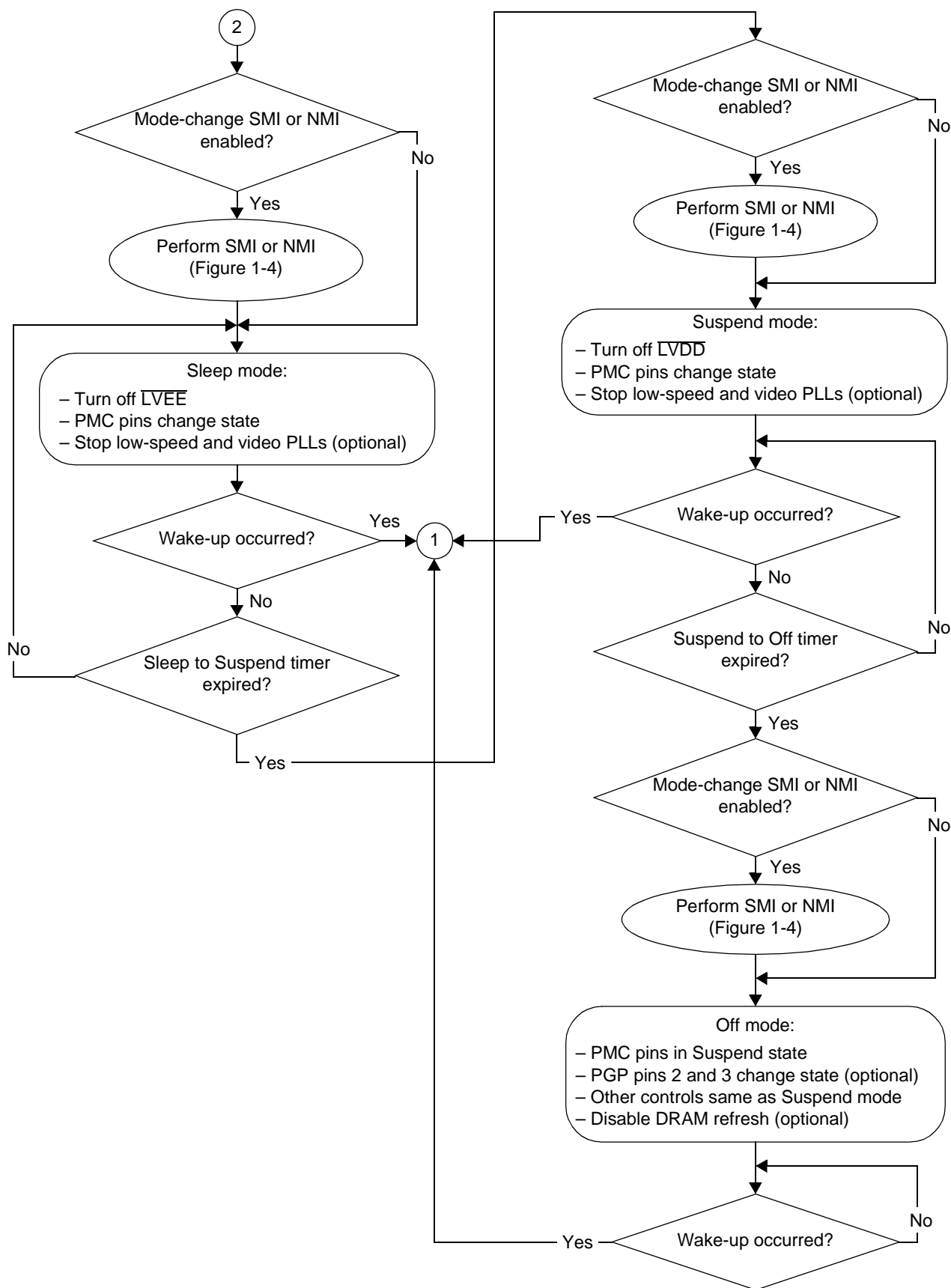


Figure 1-3 State Transition Flowchart (Continued)



## 1.5 STATE-TRANSITION TIMER

The state-transition timer logic allows the system designer to specify the amount of time that the PMU waits between state transitions when no activity is occurring. As shown in Table 1-3 on page 1-11, the time values are programmed through the Mode Timer registers at indexes 83–87h.

The High-Speed to Low-Speed Mode Timer register at Index 83h is used in High-Speed PLL mode. By default, it is programmed in increments of  $1/512$  s, with a minimum time of  $1/512$  s when set to a value of 1 and a maximum time of 498 ms ( $1/512 \cdot 255$ ) when programmed to FFh. If bit 6 of the PMU Control 2 register at Index AFh is 1, the time increments for this register change to  $1/16$  s, for a minimum time of  $1/16$  s and a maximum time of 15.94 s. When this time expires, the PMU changes to Low-Speed PLL mode.

The Low-Speed to Doze Mode Timer register at Index 84h is used in Low-Speed PLL mode. By default, it is programmed in increments of  $1/16$  s, with a minimum time of  $1/16$  s when set to a value of 1 and a maximum time of 15.94 s when programmed to FFh. If bit 7 of the PMU Control 2 register at Index AFh is 1, the time increments for this register change to  $1/4$  s, for a minimum time of  $1/4$  s and a maximum time of 63.75 s. When this time expires, the PMU changes to Doze mode.

The Doze to Sleep Mode Timer register at Index 85h is used in Doze mode. It is always programmed in 4-s increments, with a minimum time of 4 s and a maximum time of 1020 s ( $4 \cdot 255$ ). When this time expires, the PMU changes to Sleep mode.

The Sleep to Suspend Mode Timer register at Index 86h is used in Sleep mode. It is always programmed in  $1/16$ -s increments, with a minimum time of  $1/16$  s and a maximum time of 15.94 s ( $1/16 \cdot 255$ ). When this time expires, the PMU changes to Suspend mode.

The Suspend to Off Mode Timer register at Index 87h is used in Suspend mode. It is always programmed in 64-s increments, with a minimum time of 64 s and a maximum time of 16320 s ( $64 \cdot 255$ ), equivalent to 4 hr and 32 min. When this time expires, the PMU changes to Off mode. Since Off mode is the last state, there is no timer for it.

The State Transition Timer, also known as the PMU Timer, is an internal timer whose value cannot be read by the system. This timer counts up from 0 and is compared to the Transition Timer register associated with the current PMU mode. When the PMU Timer exceeds the time programmed into the appropriate Mode Timer register, the mode change is initiated and the PMU Timer is reset to 0.

When one of the above registers is cleared, upon reaching the PMU mode corresponding to that register, the PMU timer will reset and stop counting. The PMU will thus remain in that mode unless an activity is detected or the PMU is forced into another state via the  $\overline{\text{SUS/RES}}$  pin,  $\overline{\text{BL2}}$  or  $\overline{\text{BL4}}$  pin, or a write to the Software Mode Control register at Index 88h. Thus, for example, if the High-Speed to Low-Speed Mode Timer register at Index 83h is cleared, the PMU never leaves High-Speed PLL mode unless one of the previously mentioned events occurs.



The PMU timer is also reset by any of the following events:

- An unmasked PMU activity event
- A write to the Software Mode Control register at Index 88h
- An SMI or NMI caused by the battery-level pins, a PMU mode change, or a  $\overline{\text{SUS}}/\overline{\text{RES}}$  pin pulse (the timer is held in reset during one of the above SMIs until a write to the NMI/SMI Control register at Index A5h occurs)
- The high-speed or low-speed PLL is started (the timer is held in reset until startup is completed)

## 1.6

### WAKE-UP LOGIC

The ÉlanSC300 microcontroller's wake-up logic provides a mechanism for allowing certain external events to bring the PMU out of Sleep, Suspend, or Off mode into High-Speed PLL mode. These events are defined in the Resume Mask register at Index 08h. See Table 1-8 on page 1-23 for a list of the wake-up signals and associated trigger mechanisms. If the PMU is in Low-Speed PLL or Doze mode, these wake-up events function as activities, returning the PMU to High-Speed PLL mode.

Table 1-8 Wake-Up Signal Descriptions

Signal	Trigger	Description
DRQ1	Rising edge	Active until DRQ1 goes Low. Not maskable at the DMA controller.
DRQ2	Rising edge	Active until DRQ2 goes Low. Not maskable at the DMA controller.
DRQ3	Rising edge	Active until DRQ5 goes Low. Not maskable at the DMA controller.
DRQ5	Rising edge	Active until DRQ5 goes Low. Not maskable at the DMA controller.
DRQ6	Rising edge	Active until DRQ6 goes Low. Not maskable at the DMA controller.
DRQ7	Rising edge	Active until DRQ7 goes Low. Not maskable at the DMA controller.
IRQ1	Rising edge	Active until the keyboard controller deasserts IRQ1. Not maskable at the PIC.
IRQ3	Rising edge	Active until IRQ3 goes Low or channel 3 receives INT/ACK and EOI. Maskable at the PIC.
IRQ4	Rising edge	Active until IRQ4 goes Low or channel 4 receives INT/ACK and EOI. Maskable at the PIC.
IRQ8	Rising edge	Active until IRQ8 goes Low or channel 8 receives INT/ACK and EOI. Maskable at the PIC.
RI	Falling edge	Active only until the next refresh. Clearing bit 5 of the Resume Status register at Index 09h prior to the next refresh may not clear the $\overline{\text{RI}}$ activity status.
BVD1_A	Falling edge	Active only until the next refresh. Clearing bit 0 of the Resume Status register at Index 09h prior to the next refresh may not clear the BVD1_A activity status.
BVD1_B	Falling edge	Active only until the next refresh. Clearing bit 1 of the Resume Status register at Index 09h prior to the next refresh may not clear the BVD1_B activity status.

The events that may be allowed are Ring-Indicate signals from either of the PCMCIA slots or the internal UART, as well as IRQ3 or IRQ4 (from the internal UART or an external pin, depending on the system configuration) and IRQ8 (from the internal RTC only). In

addition to the events defined in the Resume Mask register, a rising edge on DRQ2 could be programmed as a wake-up event through bit 2 of the Activity Mask 1 register at Index 75h. A rising edge on ACIN may be enabled as a wake-up activity (bit 6 of the Activity Mask 1 register and bit 4 of the PMU Control 3 register at Index ADh). And a rising edge on IRQ1 (AT keyboard interrupt) could also be enabled as a wake-up activity (bit 4 of the Activity Mask 1 register). If the system is configured in Maximum ISA mode, it is also possible to use DRQ1 or DRQ7–DRQ3 as rising-edge wake-up events (bits 0, 2, and 3 of the Activity Mask 1 register). If the system is configured in Local-Bus mode, the choice of DRQs is limited to DRQ1, DRQ2, or DRQ5.

When using IRQ or DRQ pins as wake-up signals, external hardware must hold the active state (High) until the PMU returns to High-Speed PLL mode. (This is not necessary for the Ring-Indicate wake-up signals because they are internally latched.) IRQ wake-up events (IRQ3, IRQ4, and IRQ8) are masked at the PIC and must be enabled in order to generate a wake-up. However, unlike the IRQs, the DRQ events are not masked at the DMA controller. Also, for IRQ wake-ups, care must be taken to insure signal integrity on the IRQ lines. Glitches on the IRQ lines as small as 4 ns will cause a wake-up to occur. The PMU has no ability to filter glitches; therefore, the system could be returned to High-Speed PLL mode by what amounts to a spurious interrupt.

If using multiple DRQs and/or IRQs as wake-up signals, the system designer must ensure that one wake-up signal does not block the others by being held continuously in the active state while its corresponding PMU-activity-mask bit is set to 0. These wake-up sources are logically ORed before edge detection is performed. Wake-ups received during Temporary-On SMI or NMI routines (see “Temporary-On Mode” on page 1-26) require the following special attention:

1. The wake-up is delayed (not serviced by the PMU logic) until the SMI or NMI routine is completed.
2. The wake-up is serviced and cleared before exiting the SMI or NMI handler when all of the following conditions are true:
  - CPU interrupts are enabled by the routine (to handle other non-wake-up events)
  - There is an associated interrupt service routine (ISR) for the wake-up event
  - The wake-up is not masked at the PIC

Note that the interrupt handler does not have to explicitly clear the wake-up source. The CPU just has to acknowledge the active IRQ channel. This results in a lost wake-up event from the first case. Therefore, if CPU interrupts need to be serviced from within an SMI/NMI handler, the IRQs that are set up as wake-ups must be masked off at the PIC (disabled) at the beginning of the SMI/NMI routine, and unmasked (reenabled) upon exiting the routine. Make sure CPU interrupts are disabled before reenabling wake-ups at the PIC.

A false wake-up is generated when enabling BVD1\_A or BVD1\_B as a ring-in wake-up while the signal is being held Low. A wake-up from one of these signals is supposed to be generated by a falling edge of the signal. However, if the PCMCIA device is holding the signal Low and software enables the wake-up and enables the socket as an I/O card (via the Resume Mask register at Index 08h), then the PMU will generate a wake-up. This probably will not be noticed by software running in Full-On mode. However, the wake-up status bits (1–0) of the Resume Status register at Index 09h get set and remain set until

explicitly cleared. It is these *unexpected* status bits in the Resume Status register that are potentially a problem. Software must clear the Resume Status register and the Activity Status 1 register at Index A0h after enabling the BVD pins as wake-ups and enabling the devices as I/O cards (via the Resume Mask register). This clears the status registers of false event flags.

The ring-in wake-ups ( $\overline{RI}$ , BVD1\_A, and BVD1\_B) persist only until the next refresh. If software tries to clear the status bits in the Resume Status register prior to that subsequent refresh, the bits may remain set. Software must clear the Resume Status register on two successive refreshes.

An additional wake-up event is the  $\overline{SUS/RES}$  pin. This event is not maskable by the system. For information, see “Suspend/Resume Pin Logic” on page 1-34.

Figure 1-3 on page 1-20 and shows how wake-ups play a part in the power-management flow.

Clearing bit 0 in this register allows a falling edge on the BVD1\_A pin to cause a PMU wake-up event. Bit 6 must be set to 1 to enable this. The falling edge is latched internally by the PMU.

## 1.7 SMI AND NMI CONTROL

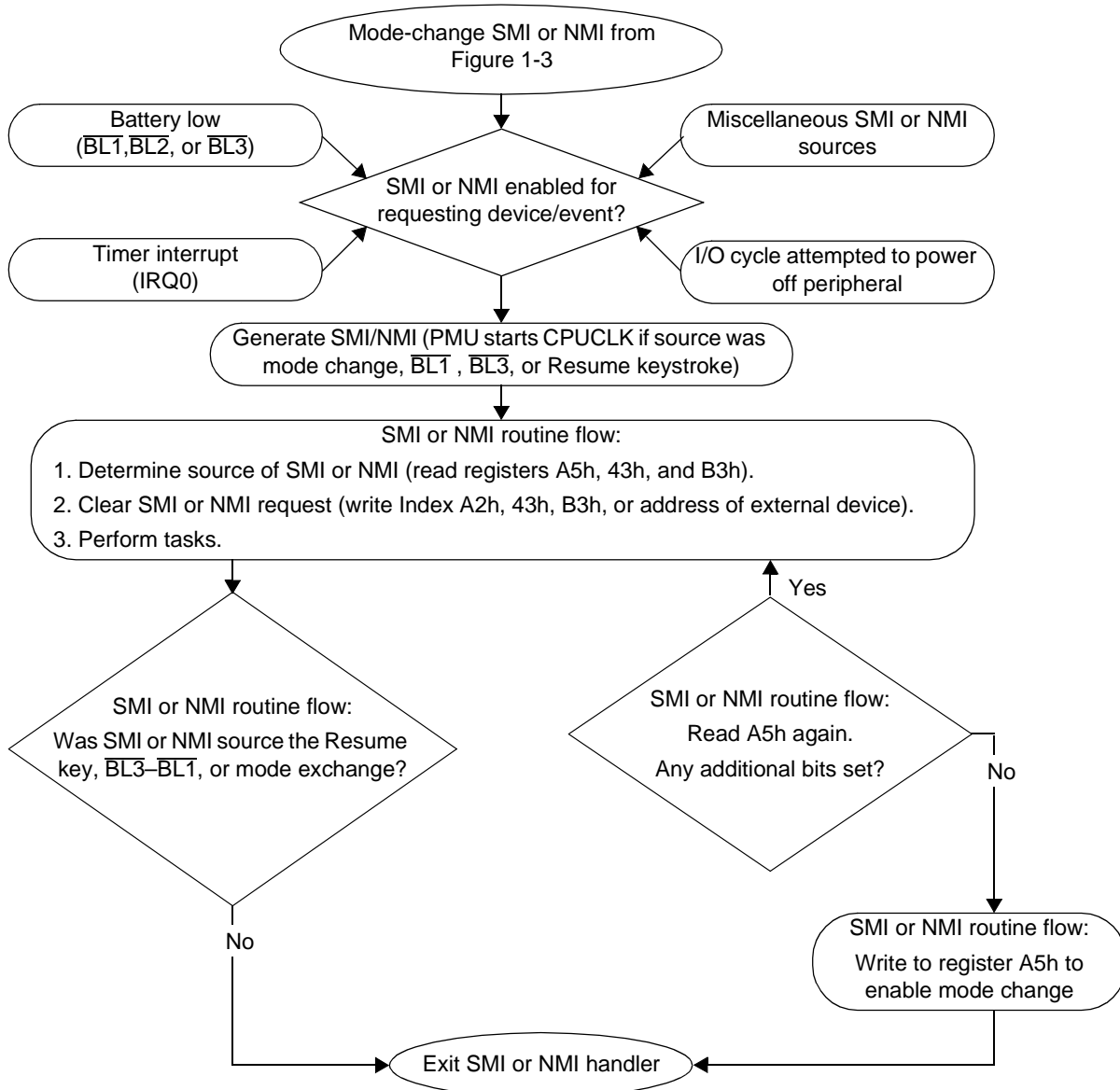
The use of SMIs gives the designer the ability to customize and extend the features of a system’s power management. This is due to the transparency of SMIs relative to operating systems and application software. The ÉlanSC300 microcontroller may be programmed to generate SMIs from the following sources:

- PMU mode changes (e.g., on a change from Sleep mode to Suspend mode)
- Battery-level changes ( $\overline{BL1}$ ,  $\overline{BL2}$ , and  $\overline{BL3}$ )
- $\overline{SUS/RES}$  pin activation (only on wake-up)
- Attempted I/O accesses to devices that the PMU has powered down after a specified time-out period
- Interrupts from IRQ0 (Timer Channel 0 must be programmed, and IRQ0 must be enabled at the 8259 PIC)
- PCMCIA status changes
- System-management interrupts on the EXTSMI pin
- I/O accesses to the RTC or the external 8042 keyboard controller

Of the above SMI sources, the first three may be programmed to generate an NMI instead of an SMI.

Refer to Figure 1-4 on page 1-26 for the general flow of an SMI or NMI.

Figure 1-4 SMI Processing Flowchart



### 1.7.1 Temporary-On Mode

Only the first three SMI or NMI sources listed above will cause the CPU, DMA, and internal system clocks to start if they were stopped. If not in the High-Speed PLL or Low-Speed PLL mode, the clocks will run based on the 9.2-MHz frequency until the next refresh after an I/O write to the NMI/SMI Control register at Index A5h is performed. This is called Temporary-On mode.

This mode is useful when it is desired to have the CPU perform certain tasks before changing PMU modes, or when the CPU receives a low-battery indication or a resume keystroke, regardless of whether it is asleep or awake. For the first two sources, when in Temporary-On mode, the PMU remains in the same state it was in before the CPU clock started. It does not change to High-Speed PLL mode. For the third source, the PMU makes a transition to High-Speed PLL mode before the SMI is generated. Writing to the

NMI/SMI Control register causes the PMU to leave Temporary-On mode and return to normal operation for the mode it was previously in, except that for a mode-change SMI, the PMU will advance to the next PMU mode after leaving Temporary-On mode. Note that if the PMU was in High-Speed PLL mode when receiving a battery-level-change SMI or NMI, it does not slow down the clock, but continues running at High-Speed PLL mode clock rates. While the PMU is in Temporary-On mode, all activities, wake-up events, and PMU mode transition timers are disabled. However, activities and wake-up events are internally latched and cause the appropriate action to be taken when exiting Temporary-On mode.

When the system SMI handler determines that an SMI has been generated from the first three sources, the write to the NMI/SMI Control register at Index A5h should always be performed at the end of the SMI handler. The NMI/SMI Control register should be read just prior to writing to it. If any additional SMI flags are set, they should be processed prior to writing the NMI/SMI Control register.

In some cases, it may be important to ensure that the CPU clocks are stopped prior to exiting the SMI/NMI routine. This prevents the main code from accessing a device that was just powered down in the routine. This can be accomplished by polling the Port B register for a refresh after the write to the NMI/SMI Control register. The clocks are stopped when the refresh occurs. When the clocks start again, the SMI/NMI routine finishes executing.

### 1.7.2 Enabling SMIs

To enable SMI generation, bit 7 of the MMSB Socket register at Index A9h must be 1. If bit 7 is 0, the first three SMI and NMI sources generate NMIs instead of SMIs. When enabling SMIs, the SMI MMS Page register at Index AAh and bits 4 and 5 of the MMSB Socket register must be programmed to set the MMS page where SMI CPU core data will be preserved. Bit 0 of the Miscellaneous 2 register at Index 6Bh gives the programmer the option of forcing address bit 20 Low during SMI routines.

Table 1-9 on page 1-27 shows the registers that enable the various SMI sources.

Table 1-9 Registers that Enable SMI Sources

Register Name	Index
SMI Enable	41h
PCMCIA Status Change IRQ Enable	0Dh
PCMCIA Status Change IRQ Redirection	0Eh
NMI/SMI Control	A5h
Function Enable 1	B0h
PMU Control 3	ADh
NMI/SMI Enable	82h

### 1.7.3 Processing NMI or SMI Source

On reset, firmware must check bit 7 of the Version register at Index 64h to determine if the reset was caused by an SMI. Upon receiving an SMI and entering the SMI handler, the CPU must then poll the various SMI status registers to determine the source of the SMI. The status registers are as follows:

- Miscellaneous 5 register at Index B3h for the EXTSMI pin
- SMI Status register at Index 43h for the following sources:
  - Hard drive
  - Floppy disk drive
  - Programmable I/O
  - 8042 accesses
  - RTC accesses
  - IRQ0 generation
  - PMU status changes
  - PCMCIA status changes

If PMU status changes (bit 5) cause the SMI, the SMI handler must read the NMI/SMI Control register at Index A5h to determine the source of the PMU SMI. If PCMCIA status changes (bit 7) cause the SMI, the SMI handler must read the PCMCIA Status Change register at Index A6h to determine the PCMCIA SMI source.

If the SMI or NMI is caused by either PMU status changes, PCMCIA status changes, or IRQ0 generation, the PCMCIA Socket A Status register at Index A2h must be written in order to reset the SMI generation logic. This write must occur after reading the SMI Status register at Index 43h and the NMI/SMI Control register at Index A5h. Failure to follow this order may result in additional SMIs being generated with incorrect status.

If the SMI or NMI was caused by a PMU status change, a write to the NMI/SMI Control register must occur near the end of the routine. Prior to performing this write, the NMI/SMI Control register should be read again to determine if additional PMU status changes need to be processed. If additional PMU status changes need to be processed, the write to the NMI/SMI Control register should be delayed, the PCMCIA Socket A Status register should again be written, and any code specific to the new status change should be executed.

A condition exists where the SMI or NMI state-transition status (as reflected in the NMI/SMI Control register at Index A5h) can be different than the PMU's current state. If a state-transition SMI or NMI is received (and subsequently queued by the PMU logic) while the system is executing another state-transition SMI or NMI routine, and the system is forced to another state via the Software Mode Control register at Index 88h, *before* exiting the routine but *after* clearing the NMI/SMI Control register (to allow the state transition), then the incoherency causing the difference can occur.

The source of the incoherency lies in the fact that as soon as the current SMI or NMI completes, the CPU begins processing the queued-up state-transition SMI or NMI. Upon entry to the handler, if the code reads the NMI/SMI Control register to determine the source of the SMI/NMI, the PMU reports a state transition based on the mode change

---

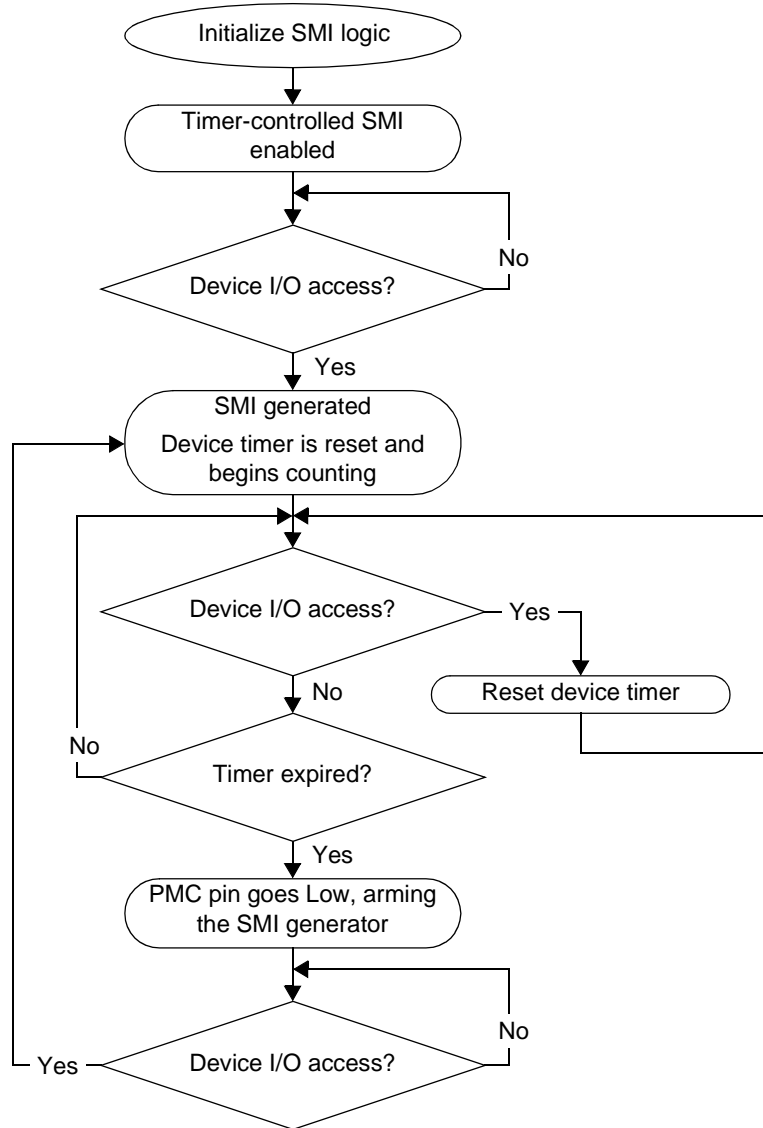
*before* the write to the Software Mode Control register, while a read of the current state as shown in the CPU Status 1 register at Index A4h reports the mode that occurred *after* the write to the Software Mode Control register.

To prevent this incoherency from occurring, forced state transitions caused by writes to the Software Mode Control register should not be performed after a write to the NMI/SMI Control register from within a Temporary-On SMI or NMI routine.

#### 1.7.4 Accesses to Powered-Down Device SMI

The ÉlanSC300 microcontroller's automatic device-power-down logic consists of a set of three timers, each dedicated to a specific device at a hardwired (floppy-disk and hard drive) or programmed (PIO) address, and associated with a specific PMC pin (see Table 1-7 on page 1-14). The flowchart in Figure 1-5 on page 1-30 illustrates the operation of this logic. Each timer may be programmed to one of 16 possible time-out values. See "PIO Timer Register (Index 46h)" on page 5-27 and "Drive Timer Register (Index 47h)" on page 5-29. Each timer is cleared when an activity is detected on the assigned address. The addresses for floppy-disk and hard drives are hardwired to the AT standard locations. The PIO Address register at Index 45h is used to set the PIO base address. Bits 6 and 7 of the PIO Timer register at Index 46h set the number of bytes decoded. Optionally, bit 0 (IRQEN) of the Miscellaneous 4 register at Index 44h may be set to enable a hard-drive interrupt (IRQ14) and a floppy-disk-drive interrupt (IRQ6) to also be considered activities to their respective timers. However, these IRQs will not generate SMIs.

Figure 1-5 SMI Device-Powerdown Flowchart



SIMs from the timer-controlled devices are enabled by bits 2–0 of the SMI Enable register at Index 41h. Once a device is enabled, the first I/O access to that device after enabling SIMs generates an SMI. This initial SMI can be differentiated from subsequent SIMs by virtue of the fact that the PMC power pins have not been cleared. After this first SMI is processed, the timer is enabled. Any I/O access (or optional interrupts) to the device reset the timer. If no accesses occur before the timer expires, the SMI generator is armed, and the next I/O access generates an SMI and resets the timer. Expiration of the timer also causes the appropriate General-Purpose I/O register to change to 0, which then causes the device to power down if the PMC pin is used to control device power. In this case, an SMI indicates that an I/O access was attempted to a powered-down device. In the ÉlanSC300 microcontroller, CPU I/O trapping is enabled, so it is possible to retry the attempted I/O instruction after powering up the device by reinitializing the appropriate PMC register and performing whatever device initialization commands are required. The states of address bit 0,  $\overline{\text{IOR}}$ ,  $\overline{\text{IOW}}$ , and  $\overline{\text{BHE}}$  are latched in the SMI I/O Status register at Index 42h at the end of the I/O command that caused the SMI. Note that this applies only



to the SMIs enabled at the SMI Enable register at Index 41h and the Function Enable 1 register at Index B0h. Optionally, firmware may enable the next I/O access to any of the timer-controlled devices to unconditionally cause an SMI by writing a 0 to the appropriate bit in the I/O Timeout register at Index 40h.

I/O trapping for powered-down devices is only applicable when the CPU is not servicing an SMI. Therefore, SMI handlers should poll the I/O Timeout register at Index 40h before accessing I/O devices that may be powered down. If any one of bits 2–0 are 0, and the corresponding bits in the SMI Enable register are 1, then I/O accesses to those devices will generate an SMI.

#### 1.7.5 Treatment of Pending SMIs

If an SMI is already in progress, the ÉlanSC300 microcontroller detects and holds as pending another incoming SMI from a different device. When the current SMI routine is finished executing, another SMI is immediately asserted if other SMIs have been generated and not cleared. The exceptions to this are SMIs that are generated from I/O decodes, which include the real-time clock (RTC), keyboard, floppy disk drive, hard drive, and PIO. In these cases, another SMI is not generated. However, the status bits in the SMI Status register at Index 43h are 1.

#### 1.7.6 External SMI Pin

The External SMI (EXTSMI) pin is enabled by bit 4 of the Function Enable 1 register at Index B0h. This pin may be used by an external device to generate an SMI. It may be used for a single device or multiple devices as explained below. Pin polarity may be selected using bit 5 of this register.

#### 1.7.7 External SMI with a Single Device

If the EXTSMI polarity bit (bit 5 of the Function Enable 1 register at Index B0h) is 1, a falling edge on EXTSMI causes an SMI request to be generated. EXTSMI should then be held Low by the external device until the SMI handler releases it via an I/O write to the external device. The state of the EXTSMI pin may be read at bit 1 of the Miscellaneous 5 register at Index B3h to determine that EXTSMI has been activated. If the EXTSMI polarity bit is 0, then all the above polarities are reversed.

#### 1.7.8 External SMI with Multiple Devices

The External SMI (EXTSMI) pin may be treated as an open-drain signal driven by multiple devices. As with a single device, each SMI-generating device should continue asserting EXTSMI until acknowledged by the SMI handler. After determining that the EXTSMI pin is asserted (bit 1 of the Miscellaneous 5 register at Index B3h), the SMI handler must poll each external device and clear its SMI-generating logic if needed. When all external devices are serviced, the EXTSMI pin should return to the inactive state. As long as the EXTSMI pin is active, SMIs to the CPU will be repeatedly generated.

## 1.8 BATTERY-MANAGEMENT LOGIC

The ÉlanSC300 microcontroller's battery-management logic allows the system designer to specify up to four battery-detection levels. The four battery-level-indication input pins are named  $\overline{BL1}$ ,  $\overline{BL2}$ ,  $\overline{BL3}$ , and  $\overline{BL4}$ . A 0 on one of these inputs indicates a low-battery condition for that level. Another pin named ACIN, when High, indicates that external non-battery power has been applied. The following paragraphs give a description of the functionality of each of these pins. Table 1-10 on page 1-32 provides a summary of the functionality of  $\overline{BL4}$ – $\overline{BL1}$ .

Table 1-10 Battery-Level Management Functionality

Pin	Slow Clock	Level SMI	Edge SMI	Force to Sleep	Force to Suspend
$\overline{BL1}$	■	■			
$\overline{BL2}$			■	■	
$\overline{BL3}$		■			
$\overline{BL4}$					■

### 1.8.1 Battery Level 1

The Battery Level 1 ( $\overline{BL1}$ ) pin is intended to be used as a first-line warning, indicating that battery power is low, but that enough power remains for continued use. The state of this pin may be read directly at the CPU Status 0 register at Index A3h.

Through an option enabled by bit 5 of the PMU Control 2 register at Index AFh, the CPU clock in High-Speed PLL mode can be forced to run at 9.2 MHz on all cycles when  $\overline{BL1}$  and ACIN are Low, thus prolonging battery life.

$\overline{BL1}$  may also be programmed to generate SMIs. This feature is enabled by bit 5 of the NMI/SMI Enable register at Index 82h. This is a level-triggered SMI (i.e., the input should be held in the trigger state, not pulsed). After system reset, a level change on  $\overline{BL1}$  from 1 to 0 generates an SMI. During the SMI service routine, the CPU must read the NMI/SMI Control register at Index A5h to reset the  $\overline{BL1}$  SMI-generation logic. This read also sets the logic to look for the opposite state on  $\overline{BL1}$  to trigger the next SMI. Therefore, any change of state on  $\overline{BL1}$  generates an SMI, provided that a read from the NMI/SMI Control register is always performed in the SMI service routine. Because the NMI/SMI Control register is also one of the SMI source-polling registers, this read is needed anyway to determine the SMI source. The SMI handler must also write to the PCMCIA Socket A Status register at Index A2h to clear the PMU SMI request.

An SMI generated by  $\overline{BL1}$  causes the CPU clock to be started, regardless of the PMU mode (see “Temporary-On Mode” on page 1-26). The clock remains running until the next refresh cycle following a write to the NMI/SMI Control register at Index A5h.

### 1.8.2 Battery Level 2

The Battery Level 2 ( $\overline{BL2}$ ) pin is intended to be used as a second-line warning in a four-level battery-management system, or as a final warning in a two-level system. The state of this pin may be read directly at the CPU Status 0 register at Index A3h.

The PMU may be programmed to automatically enter Sleep mode after a 0 is detected on  $\overline{BL2}$  while ACIN is also at 0. This feature is enabled by clearing bit 6 of the MMSB Control register at Index 74h, and is enabled by default. The PMU does not enter Sleep mode instantaneously, but steps down a mode level on each refresh until Sleep mode is reached. Once the PMU is in Sleep mode, the PMU state-transition timers control transitions to Suspend and Off modes. Enabled wake-up/activity events will cause the PMU to change to High-Speed PLL mode. However, the PMU will step down a mode level on each refresh if the  $\overline{BL2}$  pin is Low. PMU state transition to Sleep mode can be temporarily interrupted by enabling Low-to-Doze or Doze-to-Sleep mode-change NMI/SMI events in the NMI/SMI Enable register at Index 82h.

SMIs may be generated by  $\overline{BL2}$ . This feature is enabled by bit 6 of the NMI/SMI Enable register. There are important differences between the SMI behavior of  $\overline{BL2}$  relative to  $\overline{BL1}$  and  $\overline{BL3}$ . SMIs are generated only on falling edges of  $\overline{BL2}$ . During the SMI handler, the CPU must read the NMI/SMI Control register at Index A5h to reset the  $\overline{BL2}$  SMI generation logic. The SMI service routine must also write to the PCMCIA Socket A Status register at Index A2h to clear the PMU SMI request.

SMIs from  $\overline{BL2}$  are masked in Sleep, Suspend, and Off modes and when pin  $\overline{BL4}$  is Low. If an SMI from  $\overline{BL2}$  occurs in Doze mode, and the CPU clock is stopped, the clock is started to execute the SMI. The clock remains running until the next refresh cycle following a write to the NMI/SMI Control register.

### 1.8.3 Battery Level 3

The Battery Level 3 ( $\overline{BL3}$ ) pin is intended to be used as a third-line warning, indicating that battery power is low, but that enough power remains for limited use. The state of this pin may be read directly at the CPU Status 0 register at Index A3h.

$\overline{BL3}$  may also be programmed to generate SMIs. This feature is enabled by bit 7 of the NMI/SMI Enable register at Index 82h. The functionality of this SMI is analogous to that of the  $\overline{BL1}$  SMI.

An SMI generated by  $\overline{BL3}$  causes the CPU clock to be started, regardless of the PMU mode (see “Temporary-On Mode” on page 1-26). The clock remains running until the next refresh cycle following a write of 0 to the NMI/SMI Control register at Index A5h.

### 1.8.4 Battery Level 4

The Battery Level 4 ( $\overline{BL4}$ ) pin is intended to be used as an indication of the end of useful battery life. It cannot be programmed to generate an SMI, and the state of  $\overline{BL4}$  cannot be read.

The PMU may be programmed to automatically enter Suspend mode after a 0 is detected on  $\overline{BL4}$  while ACIN is also 0. This feature is enabled by clearing bit 7 of the MMSB Control register at Index 74h and is enabled by default. The PMU does not enter Suspend mode instantaneously, but steps down a mode level on each refresh until Suspend mode is reached. Once the PMU is in Suspend mode, the PMU state-transition timers control when (or if) there will be a transition to Off mode. After the PMU has been forced into this condition, enabled wake-up/activity events will cause the PMU to transition to High-Speed PLL mode. However, the PMU steps down a mode level on each refresh if the  $\overline{BL4}$  pin is Low. PMU state transition to Suspend mode can be temporarily interrupted by ena-

bling Low-to-Doze, Doze-to-Sleep, or Sleep-to-Suspend mode-change NMI/SMI events in the NMI/SMI Enable register at Index 82h.

Even if the automatic Suspend-mode feature is disabled, a 0 on  $\overline{BL4}$  always forces the PMU into Suspend mode on the next refresh after entering Sleep mode. However, system wake-up is allowed.

### 1.8.5 AC Input Status Pin

The AC Input Status (ACIN) pin is used to indicate that a permanent power source is driving the system. When ACIN is High (indicating that external power is applied), the PMU state-transition timers are disabled for High-Speed PLL to Low-Speed PLL, Low-Speed PLL to Doze, and Doze-to-Sleep mode transitions. This allows the system to remain in High-Speed PLL mode when AC power is available. Sleep-to-Suspend and Suspend-to-Off mode-transition timers are still enabled. This allows the PMU to change all the way to Off mode after the PMU changes to Sleep mode due to the  $\overline{SUS/RES}$  pin logic.

Setting bit 5 of the Miscellaneous 6 register at Index 70h has the same effect as asserting the ACIN pin, with the exception that the Force-to-Sleep and Force-to-Suspend functions of  $\overline{BL2}$  and  $\overline{BL4}$  are disabled only by asserting the external ACIN pin.

The ACIN pin can also be programmed to generate a PMU-activity event. Doing this ensures that if external power is connected to a system with dead batteries, the PMU will automatically wake-up. The activity is enabled by setting bit 4 of the PMU Control 3 register at Index ADh and clearing bit 6 of the Activity Mask 1 register at Index 75h. The PMU event is generated by detecting a rising edge on an internal signal that is the logical OR of the external ACIN pin and the internal software SACIN bit (in the Miscellaneous 6 register at Index 70h).

## 1.9 SUSPEND/RESUME PIN LOGIC

The Suspend/Resume ( $\overline{SUS/RES}$ ) pin allows the system designer to provide an end-user mechanism for either waking up the system (i.e., forcing it into High-Speed PLL mode) or putting it into Sleep mode. In a typical system design, this is realized by way of a special key on the system that is logically connected to this pin.

The suspend/resume logic is triggered by a rising edge on the external  $\overline{SUS/RES}$  pin. If the PMU is in High-Speed PLL, Low-Speed PLL, or Doze mode, such an event causes the PMU to immediately step down through the PMU states (one step per refresh cycle) until it reaches Sleep mode, at which point control passes back to the PMU state machine. If the PMU is in Sleep, Suspend, or Off mode, and it receives a rising edge on the same pin, it will jump directly to High-Speed PLL mode on the next refresh cycle.

The information in this section should be used as a guide to understanding many of the controls and caveats of implementing an SMI as it pertains to a suspend/resume operation. The pseudocode that is provided only addresses handling of  $\overline{SUS/RES}$  activity. Therefore, it represents only a portion of a potentially more complex SMI handler. Code designed to manage a platform that utilizes SMIs to restart instructions after reinitializing powered-down devices, handling battery-low warnings, waking up as a result of an activity other than the  $\overline{SUS/RES}$  activity, and so forth requires additional event handling that is beyond the scope of this section.

In addition, this section assumes that you understand the System Management mode (SMM) capabilities of the ÉlanSC300 microcontroller's CPU core. For more information on SMM and SMI functionality for the ÉlanSC300 microcontroller's CPU core, see the *3-Volt System Logic for Personal Computers* manual, PID 17028.

This section deals with handling a suspend/resume operation using the SMI capability of the ÉlanSC300 microcontroller. The PMU is designed to be able to perform a minimal suspend/resume operation without any software intervention. However, software may be required to handle platform-specific code that takes care of peripheral-device state save, device power-down, or general cleanup prior to suspend. If this is done, software may be required to perform device-state restoration or other tasks upon resume so that the entire process is transparent to the application code.

These tasks may be handled by either an NMI or an SMI. This is a system-design decision. The use of an SMI to handle suspend/resume operations is shown here for demonstration purposes.

The ÉlanSC300 microcontroller's PMU has the following capabilities that are critical to implementing a suspend/resume capability using SMIs:

1. The PMU, if it is in Sleep, Suspend, or Off mode and upon detecting a rising edge on the  $\overline{\text{SUS/RES}}$  pin, may generate a processor SMI request (bit 0 of the NMI/SMI Enable register at Index 82h).
2. The PMU, if it is in Sleep, Suspend, or Off mode and upon detecting a rising edge on the  $\overline{\text{SUS/RES}}$  pin, wakes up the processor into High-Speed PLL mode.
3. The PMU, if it is in High-Speed PLL, Low-Speed PLL, or Doze mode, and upon detecting a rising edge on the  $\overline{\text{SUS/RES}}$  pin, steps down one PMU state per refresh until it reaches Sleep mode.
4. The PMU may generate a processor SMI request when the Sleep-to-Suspend mode timer expires (Sleep to Suspend Mode Timer register at Index 86h and bit 3 of the NMI/SMI Enable register at Index 82h).

Features 1 and 4 must be enabled individually to implement an SMI-based suspend/resume capability (see the indicated configuration registers). Features 2 and 3 are automatic, and require no setup. Note that there is no capability for the PMU to directly generate an SMI request as a result of activity on the  $\overline{\text{SUS/RES}}$  pin that would result in a transition to Sleep mode. For the purpose of this documentation, assume that the suspend SMI request is generated by utilizing feature 4.

The ÉlanSC300 microcontroller's  $\overline{\text{SUS/RES}}$  pin serves both as a go-to-sleep (suspend) and a wake-up (resume) signal input. In either case, a rising edge on the  $\overline{\text{SUS/RES}}$  pin is considered activity. This manual refers to such activity on the  $\overline{\text{SUS/RES}}$  input when the PMU is in Sleep, Suspend, or Off modes as a *resume input*. The same activity, if it occurs when the PMU is in High-Speed PLL, Low-Speed PLL, or Doze mode is referred to as a *suspend input*.

Other issues that should be considered are as follows:

- While processing a suspend input, an SMI occurs as a result of the Sleep-to-Suspend transition (in this example). In order to process the SMI, the PMU enters Temporary-On mode. Near the end of the SMI handler, a write of any value should be made to the NMI/SMI Control register at Index A5h. This causes the PMU to transition into Suspend mode on the refresh following the write to the NMI/SMI Control register.
- When the PMU generates an SMI, a bit is set in the NMI/SMI Control register to indicate to software which PMU event generated the SMI. Writing any value to the NMI/SMI Control register clears this bit.
- Prior to writing the NMI/SMI Control register, the register should be read again to determine if any additional SMI events have occurred during the time it took to process the previous event. If additional SMI events have occurred, they should also be processed starting with a write to the PCMCIA Socket A Status register at Index A2h. Otherwise, these events may be lost.
- If in Temporary-On mode, activity on the  $\overline{\text{SUS/RES}}$  pin is delayed until a write to the NMI/SMI Control register occurs.
- Temporary-On mode is not a PMU mode in the normal sense. Its state cannot be read at the CPU Status 1 register at Index A4h like other PMU modes. It is really a special-case state that turns on system clocks to allow the CPU to temporarily process instructions. This allows servicing an event that may occur while the PMU is in a clock-stopped mode.
- A suspend input causes further activity on the  $\overline{\text{SUS/RES}}$  pin to be ignored until the PMU has finished transitioning to Sleep mode.
- The PMU can request that the processor perform an SMI. This request is in the form of holding the CPU's  $\overline{\text{SMI}}$  signal active. The CPU ignores this request if it is already processing an SMI, and it continues to do so until the SMI handler executes the RES3 instruction. The RES3 instruction signals the end of the SMI. If the PMU is asserting an SMI request when RES3 is executed, another  $\overline{\text{SMI}}$  signal is generated by the CPU. Otherwise, control returns to the software that was initially interrupted by the SMI.
- Once an SMI handler is entered, it must clear the PMU's SMI request to the CPU. This is done by writing any value to the PCMCIA Socket A Status register at Index A2h. The  $\overline{\text{SMI}}$  CPU input is level sensitive. As stated above, if a PMU SMI request is pending upon execution of RES3, another SMI is generated immediately by the CPU. This write to the PCMCIA Socket A Status register should happen after the SMI Status register at Index 43h and the NMI/SMI Control register at Index A5h are read to determine the cause of the SMI, but before the SMI Status register and the NMI/SMI Control register are written in order to clear their status.
- When a resume input is serviced by the PMU, it places the PMU into Full-On mode and then generates the resume SMI. In other words, the PMU is *not* in a Temporary-On mode during this SMI. A suspend input at this point in the code will immediately start transitioning the PMU towards Sleep mode.
- The SMM memory area can be redirected through an MMS page at 060000h to point to any location in system memory.

### 1.9.1 Required Initialization

For the following  $\overline{\text{SUS}}/\overline{\text{RES}}$  SMI-handler design to work, the following configuration-register initializations must be performed:

- Initialize the Sleep-to-Suspend time-out to some nonzero value, such as writing 01h to the Sleep to Suspend Mode Timer register at Index 86h.
- Allow the resume input to generate an SMI. This is done via bit 0 of the NMI/SMI Enable register at Index 82h.
- Set bit 7 of the MMSB Socket register at Index A9h to select an SMI instead of an NMI. This choice applies to all PMU events that are unmasked via the NMI/SMI Enable register at Index 82h.
- Set up the SMM memory area to the desired system memory location (bits 5–4 of the MMSB Socket register at Index A9h and the SMI MMS Page register at Index AAh).
- Mask all wake-up events (via the Resume Mask register at Index 08h) and activities that can cause the system to wake up (bits 6 and 4–0 of the Activity Mask 1 register at Index 75h). This means only the resume input can wake up the system from Sleep or Suspend modes, ensuring the resume SMI is always executed.

### 1.9.2 Start of SMI Handler

SMI handler code must determine why the handler has been entered. Since all SMIs cause the processor to enter Real mode and jump to the reset vector, it must be determined early in the boot code whether an SMI handler is being executed (bit 7 of the Version register at Index 64h). If so, the handler must determine whether a suspend or a resume is in progress. This may be accomplished by reading the NMI/SMI Control register at Index A5h and checking to see whether the SMI was caused by a Sleep-to-Suspend mode transition (bit 3) or by a resume input (bit 0).

### 1.9.3 Suspend Input Caused the SMI

If a suspend input caused the SMI, the following events occurred:

1. The system was running in High-Speed PLL, Low-Speed PLL, or Doze mode.
2. A rising edge was detected on the  $\overline{\text{SUS}}/\overline{\text{RES}}$  input (a suspend input).
3. The PMU started to ignore further activity on the  $\overline{\text{SUS}}/\overline{\text{RES}}$  input ( $\overline{\text{SUS}}/\overline{\text{RES}}$  activity).
4. The PMU stepped down through the PMU modes, one per refresh, until Sleep mode was entered. (Note that the PMU would wake up to High-Speed PLL mode if a resume input had occurred at this point.)
5. The Sleep-to-Suspend timer expired.
6. The PMU mode started to change from Sleep to Suspend, causing an SMI. At the same time, the PMU became capable of buffering (delaying) one subsequent  $\overline{\text{SUS}}/\overline{\text{RES}}$  activity. (If this activity occurred at this point, it would be delayed until Index A5h was next written. The PMU would then process the activity as if it had just occurred.) In addition, the PMU entered Temporary-On mode, causing the CPU to start running at 9.2 MHz.
7. When the SMI handler was entered, SMIs were automatically masked by the CPU until the next execution of the RES3 instruction.

#### 1.9.4 Suspend Pseudocode

1. The PCMCIA Socket A Status register at Index A2h is written in order to clear the PMU SMI request to the CPU.
2. Platform-dependent code prepares the system for the suspend operation.
3. The bit location in the SMM RAM state-save area that contains the value of bit 12 of DR7 (the processor's state) must be cleared. This handles an ÉlanSC300 microcontroller CPU errata where bit 12 of DR7 is automatically set prior to the SMM state save. If this bit is not cleared in the SMM state-save area prior to the CPU executing a RES3 instruction, the erroneous bit is reloaded into DR7 and the trace opcode (F1h) is redefined as a *soft* SMI. The next trace instruction causes a soft SMI to occur.
4. The NMI/SMI Control register at Index A5h is read to determine if any additional SMI events have occurred while processing the suspend SMI. These additional events should also be processed or they will be lost.
5. The NMI/SMI Control register is written, which causes the PMU to exit Temporary-On mode on the next refresh, and transition from Sleep to Suspend mode. At the same time, the ability of the next rising edge on the  $\overline{\text{SUS/RES}}$  input to generate a wake-up into High-Speed PLL mode is converted to immediate instead of buffered.
6. At this point, the SMI handler polls bit 4 of the AT Compatible B port (61h), looking for at least one refresh. Each time the bit toggles, a refresh has occurred. This must be done to ensure that the PMU transition into Suspend mode occurs before any more of the suspend SMI handler is executed. The CPU clock thus stops before the RES3 instruction is executed. This is done to prevent control from being returned to the application that was interrupted by the SMI. This should be avoided because the system has been prepared for Suspend mode. Peripheral devices may have been turned off that cause errors if accessed in this state.

The CPU clock is now stopped and the system is suspended. From this point on, an edge on the  $\overline{\text{SUS/RES}}$  pin is considered a resume input. Note that although RES3 has not been executed by the CPU (thus ending the suspend SMI), all remaining suspend-SMI code after this is processed as the result of a resume input. The effect is that the last few instructions of the suspend handler (everything before RES3) are really the first few instructions of the resume code. This is not to say that the resume SMI entry point is anywhere in the suspend SMI handler. The remainder of the handler execution follows:

1. A resume input is detected. This transitions the PMU into High-Speed PLL mode and causes it to assert an SMI request to the CPU. Note that the SMI request is not seen by the CPU at this point because RES3 has not been generated for the suspend SMI.
2. The CPU RES3 instruction is executed to signal the end of the SMI routine. This allows the CPU to detect pending SMIs asserted by the PMU. As a result of the pending SMI request, an SMI is generated immediately, and the SMI routine is entered to process the resume.



### 1.9.5 Resume Input Caused the SMI

If a resume input caused the SMI, the following events occurred:

1. The CPU was in the Sleep, Suspend, or Off mode.
2. A rising edge was detected by the  $\overline{SUS}/\overline{RES}$  input.
3. The PMU jumped directly into High-Speed PLL mode on the first refresh following the resume input.

**Note:** At this point, the next  $\overline{SUS}/\overline{RES}$  edge is considered a suspend input.

4. The resume input caused an SMI.
5. The SMI handler was entered, and SMIs were automatically masked by the CPU until a RES3 is executed.

### 1.9.6 Resume Pseudocode

1. The PCMCIA Socket A Status register at Index A2h is written in order to clear the PMU SMI request to the CPU.
2. Platform-dependent code prepares the system for the resume operation.
3. The NMI/SMI Control register at Index A5h is read to determine if any additional SMI events have occurred while processing the suspend SMI. These additional events should also be processed or they will be lost.
4. When the PMU generates an SMI, a bit is set in the NMI/SMI Control register to indicate to software the PMU event that generated the SMI. The NMI/SMI Control register is written to clear this bit.
5. The bit location in the SMM RAM state-save area that contains bit 12 of DR7 (the processor's state) must be cleared. This handles an ÉlanSC300 microcontroller CPU errata where bit 12 of DR7 is automatically set prior to the SMM state save. If this bit is not cleared in the SMM state-save area prior to the CPU executing a RES3 instruction, the erroneous bit will be reloaded into DR7. The trace opcode (F1h) is redefined as a *soft* SMI. The next trace instruction then causes a soft SMI to occur.
6. The CPU RES3 instruction is executed to signal the end of the SMI routine and to arm further SMIs.
7. The application program regains control.

### 1.9.7 Things to Avoid

Do not allow the suspend handler to execute RES3 until the resume keystroke occurs. Failure to adhere to this results in control returning to the application program with the CPU clock running for a short period of time. If application code is executed during this time, there is a chance that powered-down peripherals will be accessed. Even DRAM may be powered down at this point as a result of the preparation for Suspend mode.

## 1.10 AUTO LOW-SPEED LOGIC

The ÉlanSC300 microcontroller's auto low-speed logic provides a way for the system designer to “fine-tune” the system's current consumption in High-Speed PLL mode. Auto Low-Speed mode is enabled by setting bit 3 of the Control B register at Index 77h. Auto Low-Speed mode has no effect unless the high-speed clock is enabled at bit 6 of the I/O Wait State register at Index 61h. When enabled, the auto low-speed logic causes the use of the high-speed clock to be disallowed at periodic intervals when in High-Speed PLL mode. During the active trigger period, all bus cycles are performed at 9.2 MHz in High-Speed PLL mode. There is no effect in any other PMU mode. The start and end of the low-speed interval is synchronized with refresh. To program the trigger and duration periods, see “Auto Low-Speed Control Register (Index 9Fh)” on page 5-66.

## 1.11 MICRO POWER OFF MODE

The Micro Power-Off feature of the ÉlanSC300 microcontroller should not be confused with a PMU mode or state such as High-Speed PLL mode, Low-Speed PLL mode, or Suspend mode.

There is no software processing required or available to enter the Micro Power-Off mode. For most applications, Micro Power-Off mode is like completely turning off the power to the system while maintaining real-time clock operation and CMOS contents. The system enters Micro Power-Off mode immediately when the  $\overline{\text{IORESET}}$  pin is sampled Low. There is no option to generate an SMI or NMI to save the system state upon detection of  $\overline{\text{IORESET}}$  being asserted.

The type of micro power-off DRAM refresh performed (CAS-before-RAS, or self-refresh) is the same as that for which the part was configured (via bit 3 of the Miscellaneous 5 register at Index B3h) prior to sampling the  $\overline{\text{IORESET}}$  pin Low.

Exiting Micro Power-Off mode is analagous to a power-up cold boot, with the exception that the bits shown below remain set to their previously programmed values. The configuration-register bits that are not reset when exiting Micro Power-Off mode are as follows:

- Bits 1–0 of the Version register at Index 64h
- Bits 7–0 of the Clock Control register at Index 8Fh
- Bits 7–4 and 2–0 of the reserved register at Index 9Dh
- Bit 1 of the PMU Control 1 register at Index A7h
- Bit 3 of the Miscellaneous 5 register at Index B3h
- Bits 4–0 of the Miscellaneous 3 register at Index BAh

The one software option that relates to Micro Power-Off mode is whether the DRAM contents will be maintained along with the CMOS/RTC functions mentioned in the preceding paragraph. This option is enabled by setting bit 2 of the Miscellaneous 3 register at Index BAh. This bit is cleared upon core reset, using the  $\overline{\text{RESIN}}$  pin.  $\overline{\text{IORESET}}$  has no effect on this bit. The bit may be used by system firmware to determine whether or not DRAM has been retained after an  $\overline{\text{IORESET}}$  has occurred.

For more information on the Micro Power-Off mode, see the *ÉlanSC300 Microcontroller Data Sheet*, PID 18514.

## 1.12 OTHER POWER-SAVING FEATURES

The ÉlanSC300 microcontroller has additional features that provide reduced power consumption. These features can be enabled at the system designer's discretion.

### 1.12.1 DMA Clock Stop

This feature is enabled by bit 3 of the Function Enable 1 register at Index B0h. It causes the clock to the 8237 DMA controller to stop except when actually needed to perform a DMA transfer. This feature operates independently of the PMU mode. If clocks are not disabled by the PMU, the DMA clock starts after one of the DRQs goes active High, and the DMA clock continues running until AEN is deasserted.

### 1.12.2 Data-Path Disabling Logic

The following bits help limit the amount of data-bus toggling in the ÉlanSC300 microcontroller to peripheral devices which are not in use:

- Bits 4 and 5 of the PMU Control 1 register at Index A7h can be set to disable data-bus toggling to the internal display controller and to the UART and PMU blocks during memory cycles.
- Bit 4 of the Miscellaneous 4 register at Index 44h, if 1, disables data-bus toggling to most of the ÉlanSC300 microcontroller (except between the CPU and the external data-bus interface) during internal DRAM cycles.

### 1.12.3 Slow Refresh

For systems using DRAM that supports slow refresh, the ÉlanSC300 microcontroller provides five user-selectable refresh rates. The default refresh rate is the slowest: 8192 refreshes per second. If bits 0 and 1 of PMU Control 1 register at Index A7h are 0, the refresh rate is controlled by bits 1 and 0 of the Version register at Index 64h. Supported rates are 32768, 16384, 10922, and 8192 refreshes per second. If bit 0 of the PMU Control 1 register at Index A7h is 0 and bit 1 is 1, the refresh rate becomes 65536 refreshes per second; if bit 0 is 1, the 8254 becomes the refresh source.

### 1.12.4 Quiet Bus

Setting bit 2 of the Control A register at Index 48h disables  $\overline{\text{MEMR}}$  and  $\overline{\text{MEMW}}$  on the ISA bus from toggling during internal memory cycles.





From the programmer's perspective, the ÉlanSC300 microcontroller manages five separate memory address spaces: three 16-Mbyte spaces and two 64-Mbyte spaces.

- 16-Mbyte address spaces
  - System memory
  - ROM BIOS
  - ROM DOS
- 64-Mbyte address spaces
  - PCMCIA Card A
  - PCMCIA Card B

In addition, the ÉlanSC300 microcontroller manages a single 64-Kbyte I/O address space.

Only the system-memory address space is directly accessible in its entirety; the ROM-BIOS and ROM-DOS address spaces are directly accessible in part. The PCMCIA-card address spaces and the bulk of the ROM-BIOS and ROM-DOS address spaces are accessible only via a memory-mapping system.

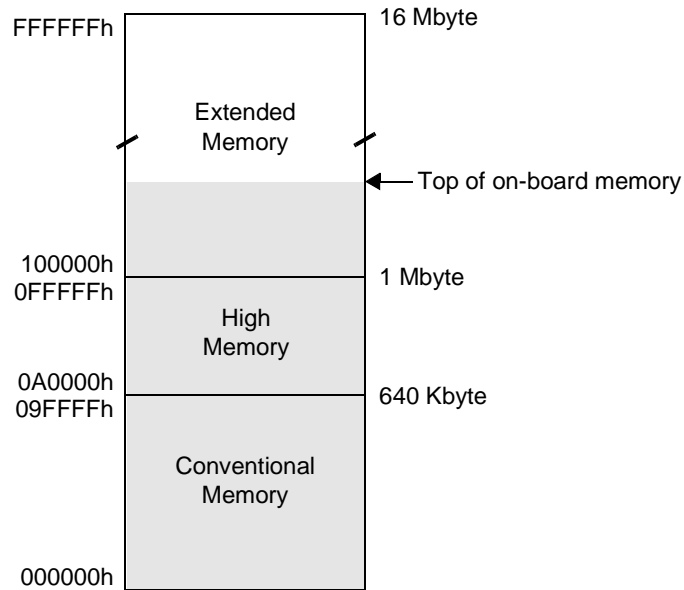
The actual memory of a system based on the ÉlanSC300 microcontroller consists of memory devices—DRAM, SRAM, EPROMs, and flash memory cards. Usually, only a subset of each address space corresponds with any device, and many addresses are not associated with any device. For example, a typical configuration might have only 8 Mbyte of DRAM system memory, 128 Kbyte of BIOS ROM, 128 Kbyte of DOS ROM, and 4 Mbyte of flash memory on each of the two PCMCIA cards.

The ÉlanSC300 microcontroller's Memory Controller Unit (MCU) is responsible for managing access to the address spaces and the associated physical memory.

2.1 SYSTEM MEMORY: DRAM, SRAM, AND BUS

Figure 2-1 on page 2-2 shows a typical system address space for the ÉlanSC300 microcontroller.

Figure 2-1 Typical AT Address Space



The three regions of system address space are conventional memory (the first 640 Kbyte), high memory (the next 384 Kbyte), and extended memory (the remaining 15 Mbyte). Only the first Mbyte (conventional and high memory)—along with the first 64 Kbyte of extended memory—can be accessed while the ÉlanSC300 microcontroller is in Real mode without using the MMS. All 16 Mbyte of system-address space may be addressed while the ÉlanSC300 microcontroller is in Protected mode. For more information on Real and Protected modes, see the *Am486<sup>®</sup> Microprocessor Software User's Manual*, PID 18497.

In Figure 2-1 on page 2-2, the shaded portion of the address space represents on-board memory. That is, it represents the portion of the system address space that is currently associated with physical memory chips attached to the ÉlanSC300 microcontroller's system board (e.g., in SIMMs). The unshaded portion can be associated with memory devices connected to the ISA or local bus. These addresses will be empty if no ISA or local-bus memory devices are attached.

The ÉlanSC300 microcontroller may have one of two on-board memory configurations:

- DRAM
- SRAM

This choice must be made when the system is designed. It cannot be changed by the programmer. The choice of DRAM or SRAM is indicated by bit 0 of the Miscellaneous 6 register at Index 70h; the default is DRAM. The choice of DRAM or SRAM changes the meaning and behavior of several pins.

The ÉlanSC300 microcontroller supports three different bus configurations:

- Partial ISA bus with internal LCD controller
- Local bus with a more complete ISA bus
- Maximum ISA bus

The selection of one of these three options also must be made when the system is designed and may not be changed dynamically. Like the choice of DRAM or SRAM, this choice also affects the meaning and behavior of several pins. For information on exactly which pins are affected, refer to the *ÉlanSC300 Microcontroller Data Sheet*, PID 18514.

### 2.1.1 DRAM Configurations

The ÉlanSC300 microcontroller supports up to 16 Mbyte of on-board DRAM, organized as one or two memory banks (16-bit data path to memory). The  $\overline{RAS0}$  and  $\overline{RAS1}$  signals enable the banks, while the  $\overline{CAS0L}$ ,  $\overline{CAS0H}$ ,  $\overline{CAS1L}$ , and  $\overline{CAS1H}$  signals are used to select the upper and lower sections of each bank. Each bank is 16 bits wide and contains 1, 2, or 4 chips depending on the bit architecture of the memory chips. The following bits are used to specify the memory size and architecture:

- Bits 4–2 of the Memory Configuration 1 register at Index 66h
- Bits 7–6 of the Function Enable 2 register at Index B1h
- Bit 7 of the PCMCIA Card Reset register at Index B4h

Insight into the mapping of both DRAM and SRAM configurations can be gained by examining Table 5-31 on page 5-41 and Table 5-46 on page 5-79. As an example of how to program the memory configuration, the settings in Table 2-1 on page 2-3 select a 4-Mbyte memory that is organized into two banks and configured for Enhanced-Page-mode operation.

Table 2-1 Memory Initialization Example

Instruction	Ports	Data	Comment
IOW IOW	022h 023h	B4h 0100 0000	Enable the Memory Configuration 1 register at Index 66h to select memory configuration.
IOW IOW	022h 023h	66h xxx1 0011	Select 4-Mbyte memory configuration (Enhanced Page mode). Note that bits 6–5 are read only.

### 2.1.2 SRAM Configurations

DRAM accesses use the MA11–MA0 pins to hold either a row address or a column address. SRAM accesses are handled via a different addressing scheme—one that SRAM shares with all the other addressable devices on the ISA or local bus.

For all cycles that are not system-DRAM accesses or accesses to the ÉlanSC300 microcontroller's internal I/O ports, the MA11–MA0 pins function as SA23–SA12 (the high 12 bits of the system address). This is not a configuration option but instead occurs automatically.

Also, SA12 has an alternate function as MA11 when the device is configured for 1-Mbit × 16-bit DRAMs. All devices on the ISA bus, all SRAM, and all ROM-DOS, ROM-BIOS, and PCMCIA spaces are thus addressed with the SA23–SA13 and SA12–SA0 signals.

In the case of SRAM, up to 16 Mbyte may be accessed, organized as two banks. Each bank is 16 bits wide. The same memory configuration is used for DRAM and is programmed using bits 4–2 of the Memory Configuration 1 register at Index 66h.

SRAM accesses are specified by setting bit 0 of the Miscellaneous 6 register at Index 70h. If this is done,  $\overline{\text{CAS0L}}$ ,  $\overline{\text{CAS0H}}$ ,  $\overline{\text{CAS1L}}$ , and  $\overline{\text{CAS1H}}$  function as SRAM chip-select pins (SRAMCS3–SRAMCS0). Table 2-2 on page 2-4 shows the key SRAM access pins.

Table 2-2 SRAM Option Configuration

Pin Name	I/O	Function
$\overline{\text{CAS0L}}$ [SRCS0]	O	SRAM Bank 0 Low Byte Select
$\overline{\text{CAS0H}}$ [SRCS1]	O	SRAM Bank 0 High Byte Select
$\overline{\text{CAS1L}}$ [SRCS2]	O	SRAM Bank 1 Low Byte Select
$\overline{\text{CAS1H}}$ [SRCS3]	O	SRAM Bank 1 High Byte Select
SA23–SA0	O	Address

### 2.1.3 Refresh and Wait States

The ÉlanSC300 microcontroller supports two memory refresh modes:

- 8254-based DRAM refresh
- Slow refresh DRAM

The 8254-based DRAM refresh cannot be selected if power management is performed because the 8254 will be turned off in Suspend mode. Memory refresh needs to remain enabled even if SRAMs are being used. The PMU uses the refresh pulse to synchronize events. The refresh frequency is under programmer control. See “PMU Control 1 Register (Index A7h)” on page 5-71 and “Version Register (Index 64h)” on page 5-38.

For example, the settings shown in Table 2-3 on page 2-4 perform the following operations:

- Select Slow Refresh mode for DRAM refreshes. The DRAMs are refreshed at the rate selected by bits 1 and 0 of the Version register at Index 64h.
- Select the 32-kHz clock input as the refresh source.

Table 2-3 Refresh Initialization Example

Instruction	Ports	Data	Comment
IOW IOW	022h 023h	A7h xxxx xx00	Enable slow refresh; select 32-kHz clock for refresh source; disable refresh during Sleep mode.



The ÉlanSC300 microcontroller also supports a variety of speeds of SRAM and ROM. Some speeds require wait states to be inserted in memory accesses between the point in a cycle when the read or write signal is asserted and the point where the data is transferred to or from the memory device. The number of wait states to be inserted under different conditions is under programmer control.

When the system is configured to use SRAMs, bit 4 of the Wait State Control register at Index 63h and bits 1–0 of the Memory Configuration 1 register at Index 66h control the number of SRAM wait states as shown in Table 5-27 on page 5-38.

Page-mode DRAMs are organized so that successive locations are along the same row, or *page*. This allows the memory controller to generate a single row address if successive accesses happen to lie in the same page (a *page hit*). When a page hit occurs, the memory cycle can be shortened by eliminating the row-address portion of the cycle. At the programmer's discretion, wait states may be inserted into the Page-mode cycle. The programmer has the option to set the number of wait states after a page miss and the number of first-cycle wait states. The number of wait states for a page-hit read cycle and a page-hit write cycle are fixed at 0 and 1, respectively.

The settings in Table 2-4 on page 2-5 set up Page-mode DRAM accesses, using the MOD select bit of the Memory Configuration 1 register at Index 66h. They set the first-cycle wait state to 3 cycles (the default) and the bank-miss wait states to 5 by setting the following bits:

- Bits 5 and 6 of the Wait State Control register at Index 63h
- Bit 5 of the ROM Configuration 1 register at Index 65h
- Bit 4 of the MMS Memory Wait State 1 register at Index 62h

Table 2-4 Memory-Speed Initialization Example

Instruction	Ports	Data	Comment
IOW IOW	022h 023h	66h xxxx xx10	Set memory mode to Page-mode DRAMs. Note that bits 7–5 of this register are read only because they affect the choice of bus.
IOW IOW	022h 023h	63h x11x xxxx	Set top bits of first-cycle wait state to 3 and bank-miss wait state to 5.
IOW IOW	022h 023h	65h xx1x xxxx	Set low bit of first-cycle wait to 3.
IOW IOW	022h 023h	62h xxx1 xxxx	Set low bit of bank-miss waits to 5.

The ÉlanSC300 microcontroller also supports Enhanced Page mode when both banks of DRAM are used. It may be selected using bit 0 of the Memory Configuration 1 register at Index 66h. This mode effectively doubles the page size by arranging the DRAM address lines such that one page is spread across both DRAM banks. This avoids the precharge penalty that would otherwise occur when incrementing across the bank-section boundary. For more information on the Enhanced-Page-mode address translation, see the *ÉlanSC300 Microcontroller Data Sheet*, PID 18514.

Table 2-5 on page 2-6 shows the number of wait states required in a 33-MHz ÉlanSC300 microcontroller configuration for 70-ns Page-mode DRAMs.

Table 2-5 33-MHz Wait States

Cycle Type	Wait States
Read cycle, page hit	0
Write cycle, page hit	1
First cycle	3
Page miss	5

#### 2.1.4 ISA, Local Bus, and Internal LCD Configurations

The ÉlanSC300 microcontroller is designed as a single-chip integration of all the principal components needed for a palmtop computer. One of the principal components is the liquid crystal display (LCD) video controller. The ÉlanSC300 microcontroller supports the following strategies for video control:

- An on-chip controller
- A local-bus interface
- The ÉlanSC300 microcontroller's Maximum ISA bus interface

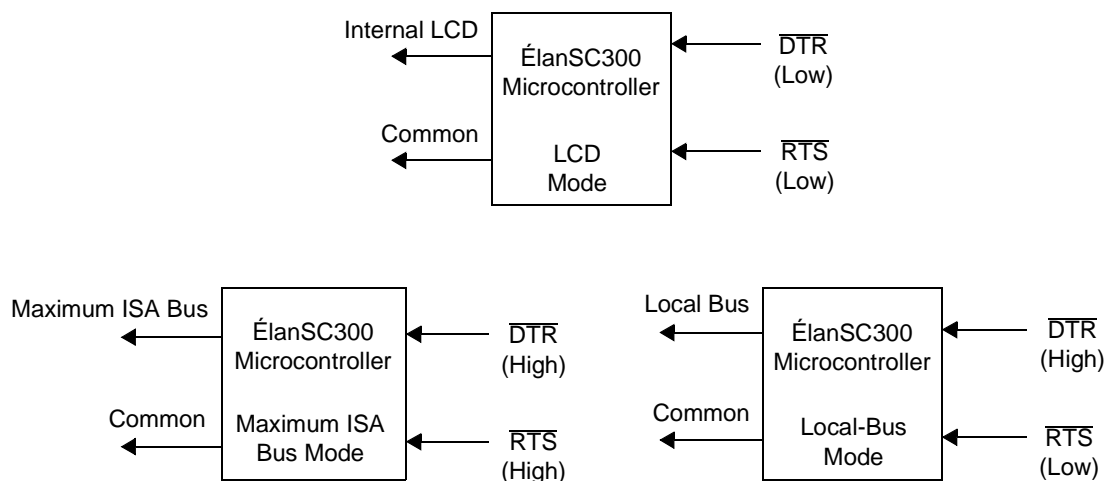
The local-bus interface is integrated with the memory controller and the ISA-bus controller, and it permits fast transfers to and from external local-bus peripherals such as video controllers. The on-chip controller also supports a subset of the ISA bus for external accesses.

These three strategies are alternatives because they share the same set of pins. The configuration employed (internal LCD controller, local bus, or Maximum ISA bus) depends on the initial values of bits 5 and 6 in the Memory Configuration 1 register at Index 66h at system reset. These initial values are determined by the values of the  $\overline{DTR}$  and  $\overline{RTS}$  pins, which are established by pull-up or pull-down resistors attached to the pins. They are not set at run time by programming the registers. Table 2-6 on page 2-6 shows the meaning of these pins and bits.

Table 2-6 Bus Configuration

Bus Selected	$\overline{DTR}$ (CFG1)	$\overline{RTS}$ (CFG0)
Internal CGA	0	0
Local Bus	1	0
Maximum ISA	x	1

Figure 2-2 ÉlanSC300 Microcontroller Bus Configurations



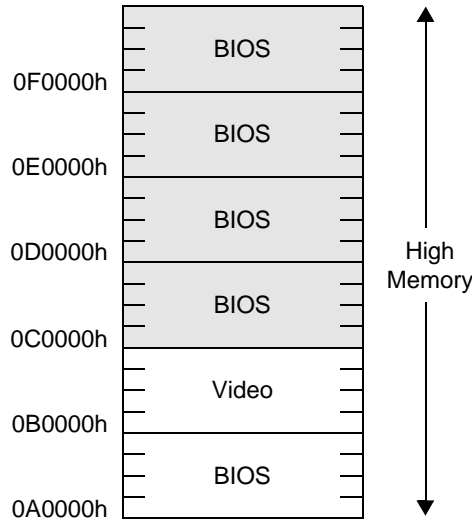
## 2.2 ROM BIOS MEMORY

ROM-BIOS memory is accessed when the ÉlanSC300 microcontroller activates the BIOS chip-select signal ( $\overline{ROMCS}$ ). The combination of this signal and the 24 address lines defines a separate 16-Mbyte ROM-BIOS address space. However, usually only a few hundred Kbyte of this space are occupied by physical ROM devices. ROM BIOS may be accessed directly or as a function of the MMS.

With the ÉlanSC300 microcontroller, up to 320 Kbyte of non-MMS ROM BIOS is supported, and addresses in the ranges 0A0000–0AFFFFh and 0C0000–0FFFFFFh can be supported. The address range for the ROM BIOS is set via the ROM Configuration 1 register at Index 65h and the ROM Configuration 2 register at Index 51h. For information, see Chapter 4, “PC/AT Peripheral Registers.” Figure 2-3 on page 2-8 shows a diagram of high memory with all the address ranges indicated that may be programmed in the ÉlanSC300 microcontroller to generate  $\overline{ROMCS}$  (indicated by the word *BIOS*). This figure also shows the address ranges that may be shadowed. Accesses to the boot-vector location, FF0000–FFFFFFh, always asserts the  $\overline{ROMCS}$  signal.

ROM BIOS is accessed as an 8-bit device unless the system hardware drives  $\overline{MCS16}$  active, which results in a 16-bit access. There is no programmable option to set whether the ROM BIOS is an 8-bit or 16-bit device.

Figure 2-3 High Memory



**Note:**

The shaded area (0C0000–0F0000h) is shadowable.

As an example, the settings shown in Table 2-7 on page 2-8 implement a 128-Kbyte ROM BIOS located at 0E0000–0FFFFFh.

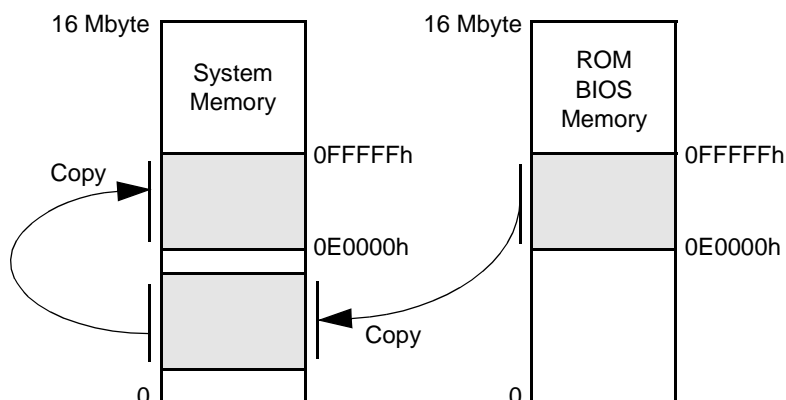
Table 2-7 ROM BIOS Address Initialization

Instruction	Ports	Data	Comment
IOW IOW	022h 023h	51h xxxx xxx0	Disable $\overline{ROMCS}$ for 0A0000–0AFFFFh.
IOW IOW	022h 023h	65h xxxx 0010	Enable $\overline{ROMCS}$ for 0E0000–0FFFFFh; disable it for 0C0000–0DFFFFh.

In order to speed up BIOS accesses, many PC systems copy the ROM-BIOS contents into on-board DRAM at startup. Then the system is set to send ROM-BIOS accesses to the DRAM instead of the ROM. This technique is called *ROM shadowing*.

The usual implementation is to copy the BIOS image stored in ROM to the same addresses in system RAM, and then transfer program control to the BIOS image in system RAM (see Figure 2-4 on page 2-9).

Figure 2-4 Copying ROM Contents



For example, to shadow ROM to RAM in a system where the BIOS ROM is a 128-Kbyte part located from 0E0000h to 0FFFFFFh in CPU address space, the following algorithm can be implemented:

1. Copy the ROM-based image into a 128-Kbyte buffer in conventional RAM.
2. Jump to the exact same place in the conventional RAM-based BIOS image that was being executed in the ROM-based BIOS image.
3. Enable the ÉlanSC300 microcontroller's shadow-RAM function and disable the shadow RAM's write-protect feature by setting bits 4 and 7 of the ROM Configuration 1 register at Index 65h.
4. Set the shadow-RAM address range to match the BIOS ROM's range of 0E0000h to 0FFFFFFh by writing FFh to the Shadow RAM Enable 2 register at Index 69h.
5. Disable  $\overline{ROMCS}$  for the range 0E0000–0FFFFFFh by setting bit 0 and clearing bit 1 of the ROM Configuration 1 register.
6. Copy the conventional RAM-based BIOS image to shadow RAM from 0E0000h to 0FFFFFFh. Once this is done, shadow RAM may optionally be write protected by clearing bit 7 of the ROM Configuration 1 register.
7. Jump to the same place in the shadow-RAM-based BIOS image that was being executed in the conventional RAM-based BIOS image.

**Note:** Use caution when performing read-modify-write sequences of the ROM Configuration 1 register. Bit 0 reads the inverse of what was last written; therefore, it must be flipped again prior to write-back to keep  $\overline{ROMCS}$  settings unchanged.

### 2.3 ROM DOS MEMORY

ROM-DOS memory is accessed whenever the ÉlanSC300 microcontroller activates the  $\overline{DOSCS}$  chip-select signal. The combination of this signal and the 24 address lines defines a separate 16-Mbyte ROM-DOS address space. This implementation is particularly well suited for accessing ROMs that contain a disk image loaded with DOS. Hence the name ROM DOS.

The ROM-DOS address space can be configured as either an 8-bit or 16-bit device. Bit 1 of the ROM Configuration 2 register at Index 51h controls this. The default is an 8-bit device.

The ROM-DOS address space is typically accessed using the ÉlanSC300 microcontroller's MMS, which is described in "Memory Mapping" on page 2-10. In addition to using MMS, up to 15 Mbyte minus 64 Kbyte of ROM-DOS address space can also be accessed as direct system memory using linear address decodes. Bits 3–0 of the ROM Configuration 3 register at Index B8h determine the address range used for the ROM-DOS decode. If these bits are 0, the ROM-DOS address space is accessible only through MMS. Table 5-50 on page 5-85 provides the bit settings for the allowable address ranges. Note that the top of the address range is fixed at FFFFFFFh. The address range FF0000–FFFFFFh is reserved for ROM BIOS.

As the size of the ROM-DOS linear-decode address space increases, the lower address boundary decreases. Care must be taken to ensure that the ROM-DOS linear-decode address range does not overlap the on-board DRAM or SRAM range. Bits 7–4 of the Miscellaneous 1 register at Index 6Fh can be used to prevent the upper portion of system DRAM or SRAM from being accessed as direct system memory, thereby eliminating the conflict.

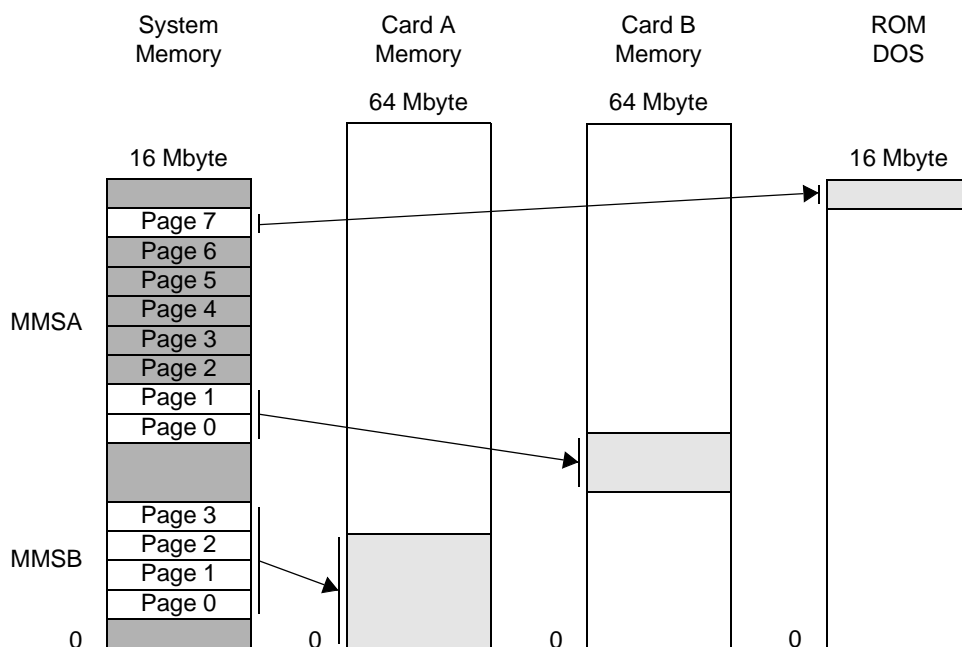
## 2.4 MEMORY MAPPING

The ÉlanSC300 microcontroller's Memory Controller supports a Memory Mapping System (MMS). The MMS defines two windows in the first Mbyte of system address space:

- MMSA, with 8 16-Kbyte pages
- MMSB, with 4 16-Kbyte pages

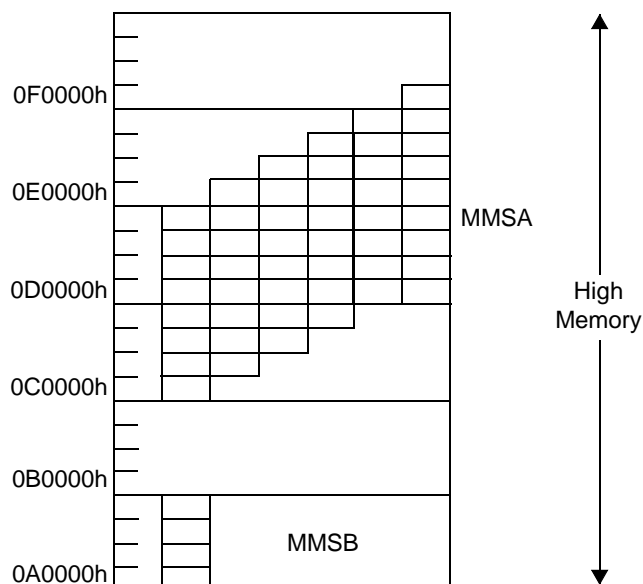
The programmer can set these pages to point to any similarly sized page of ROM DOS, PCMCIA (A or B), BIOS ROM, or on-board DRAM or SRAM. Once these pages are programmed, an address generated in a system-address page is automatically translated into an access to the mapped page of the other address space. Figure 2-5 on page 2-11 shows the basic idea. Using the ÉlanSC300 microcontroller's MMS windows, it is possible to access every addressable location in each of the five address spaces listed.

Figure 2-5 Memory Mapping System



The beginning of the MMSA window may be located at one of six starting addresses between 0C0000h and 0D4000h. The MMSB window always starts at 0A0000h. Figure 2-6 on page 2-11 shows how high memory contains the MMSA and MMSB windows. Notice that MMSA can have six different positions, depending on how the MMS Address register at Index 6Dh is programmed. The MMSA window size is fixed at 128 Kbyte (8 pages × 16 Kbyte). The MMSB window size is fixed at 64 Kbyte (4 pages × 16 Kbyte).

Figure 2-6 MMSA and MMSB



The MMS windows and their corresponding pages are set up by programming several configuration registers in the ÉlanSC300 microcontroller. In particular, control is exercised through the following registers:

- ROM Configuration 1 register at Index 65h
- MMSA Address Extension 1 register at Index 67h
- MMS Address Extension 1 and 2 registers at indexes 6Ch and 6Eh
- MMS Address register at Index 6Dh
- MMSA Device 1 and 2 registers at indexes 71h and 72h
- MMSB Device register at Index 73h
- MMSB Control register at Index 74h
- MMSA Socket register at Index A8h
- MMSB Socket register at Index A9h
- CA24–CA25 Control 1–3 registers at indexes B5–B7h

All the index registers used in this section can be referenced in Appendix A, “Configuration Index Register Reference,” as MMS registers or PCMCIA registers except for the ROM Configuration 1 register. Bits 3–0 of the MMS Address register at Index 6Dh define the location in I/O space of the eight MMSA and four MMSB page registers. Bit 1 of the MMSB Control register at Index 74h determines whether the page-register I/O locations are accessing MMSA page registers or MMSB page registers.

Each MMS window contains a global switch that enables or disables all the pages within that window. Bit 5 of the ROM Configuration 1 register at Index 65h is the switch bit for MMSA. Bit 0 of the MMSB Control register at Index 74h is the switch bit for MMSB.

As stated earlier, each MMS page is a fixed size of 16 Kbyte. Each page is mappable to an equal-size page located on a 16-Kbyte boundary in one of the five address spaces: ROM DOS, PCMCIA (A or B), ROM BIOS, or on-board DRAM or SRAM. Each individual page within an MMS window can be enabled or disabled by setting or clearing a bit in its corresponding page register.

System RAM is not the default device that is accessed if an MMS window or a page of an MMS window is disabled. This allows externally decoded memory-mapped devices on the ISA bus to exist in the address spaces that are normally decoded by the MMS windows. Thus, if more than 640 Kbyte of system RAM is desired for use by a DOS memory extender (for example) the MMS page should be programmed to point to system RAM explicitly. This feature also allows a block of system address space in high memory to be left open for use by the EMM386 memory manager to map in pages of extended memory using capabilities provided by 386-class and higher processors.

Another implication of system RAM not being the default device is that if the base address for MMSA is not set to 0C0000h, only externally decoded devices can utilize the address space abandoned by MMSA in this example. More specifically, if MMSA is made to start at 0C8000h, the 32 Kbyte of address space from 0C0000h to 0C7FFFh is usable only if decoded by an external ISA memory-mapped device (e.g., a VGA ROM).

The address spaces—ROM DOS, PCMCIA (A or B), and ROM BIOS—are implemented as chip-select signals in the ÉlanSC300 microcontroller to eliminate the need for external



address decoding. Cycles to these devices use the ISA command signals, MEMR and MEMW. Thus, any external device that does not have access to the chip signals generated by the ÉlanSC300 microcontroller (such as ISA devices) may have address conflicts with cycles that use the MMS mapping.

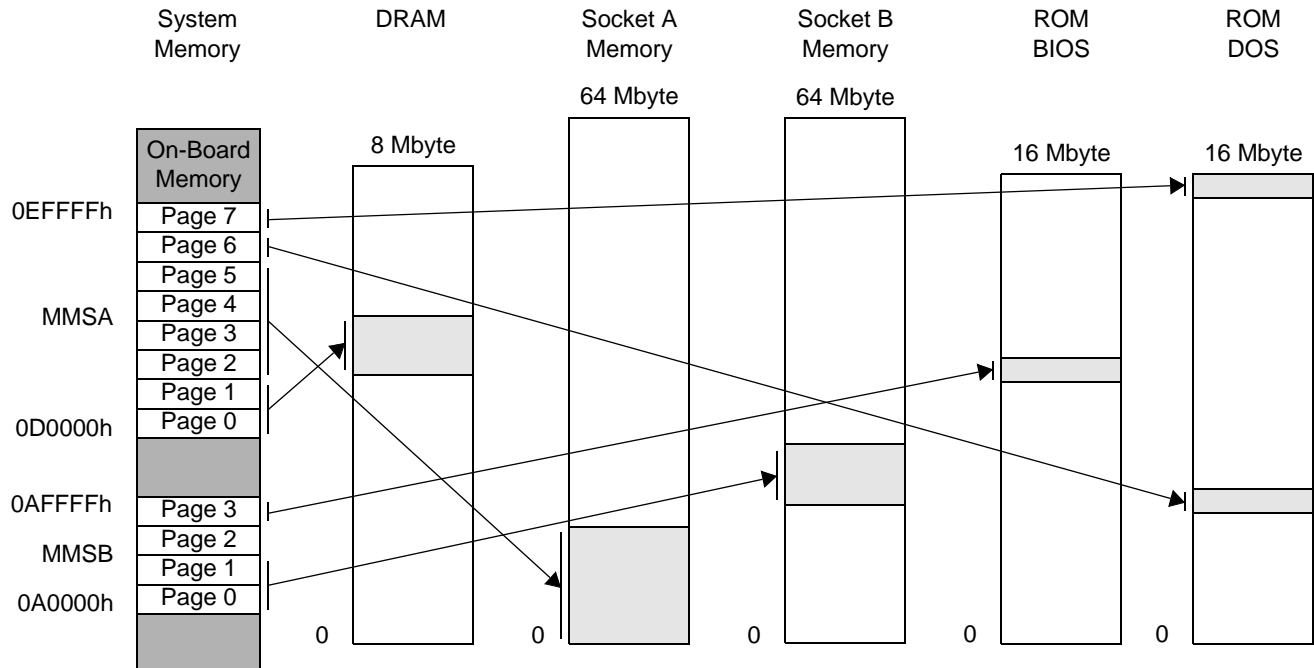
For PCMCIA implementations, the system hardware and firmware designer may take advantage of optional separate command signals for PCMCIA accesses. These command signals are pin multiplexed with signals used by the parallel port, thus reducing the parallel port's functional capabilities when the PCMCIA command-signal option is used. For details, see "Miscellaneous 3 Register (Index BAh)" on page 5-86.

There is no separate command-signal option for interfacing to the DOS ROM. It is suggested that the DOS or application ROM be made to physically reside above 1 Mbyte in the system-address space to avoid conflicts with devices whose decoding is fixed by the PC/AT architecture.

It is invalid to have ROM BIOS shadowing enabled for a region that contains an enabled MMS page. If this is done, conflicts will result when accessing the overlapping areas. For example, if ROM BIOS shadowing is enabled for the address range 0E0000–0EFFFFh, and the MMSA starting address is set to 0C4000h, then MMSA page 7 (0E0000–0E3FFFh) should always remain disabled.

Figure 2-7 on page 2-14 shows an example of MMS mapping between system-address space and all five address spaces. In this example, there are 8 Mbyte of on-board memory. Four pages of PCMCIA slot A are being mapped, along with two pages of slot B, one page of ROM-BIOS space, and two pages of ROM-DOS space. In addition, two pages of on-board memory are also mapped. The settings in Table 2-8 on page 2-14 define the MMS mapping shown in Figure 2-7 on page 2-14.

Figure 2-7 MMS Mapping Example



**Notes:**

The configuration is summarized as follows:

MMSA base address: 0D0000h

- Page 0: On-board (5FC000–5FFFFFFh)
- Page 1: On-board (600000–603FFFh)
- Page 2: Card A (0000–3FFFh)
- Page 3: Card A (4000–7FFFh)
- Page 4: Card A (8000–BFFFh)
- Page 5: Card A (C000–FFFFh)
- Page 6: ROM DOS (220000–223FFFh)
- Page 7: ROM DOS (FFC000–FFFFFFh)

MMSB base address: 0A0000h

- Page 0: Card B (1F8000–1FBFFFh)
- Page 1: Card B (1FC000–1FFFFFFh)
- Page 2: (Not mapped)
- Page 3: ROM-BIOS (400000–403FFFh)

Table 2-8 MMS Mapping Example Settings

Instruction	Ports	Data	Comment
IOW IOW	022h 023h	66h xxx1 011x	Set on-board memory size to 8 Mbyte. Note that bits 7–5 are read only because they affect the choice of bus.
IOW IOW	022h 023h	74h xxxx xx1x	Select MMSA for programming.
IOW IOW	022h 023h	6Dh 0100 0000	Set MMSA base address to D0000h and I/O base address to 0208h.
IOW IOW	022h 023h	6Ch 1000 0000	Set all translated MMSA A23s (page 7 = 1, others = 0).
IOW IOW	022h 023h	6Eh 0000 1110	Set A22 and A21 for MMSA pages 0–3.

Table 2-8 MMS Mapping Example Settings (Continued)

Instruction	Ports	Data	Comment
IOW IOW	022h 023h	67h 1101 0000	Set A22 and A21 for MMSA pages 4–7.
IOW IOW IOW IOW IOW IOW IOW IOW	0208h 2208h 4208h 6208h 8208h A208h C208h E208h	1111 1111 1000 0000 1000 0000 1000 0001 1000 0010 1000 0011 1000 1000 1111 1111	Translate A20–A14 of MMSA page 0; enable the page. Translate A20–A14 of MMSA page 1; enable the page. Translate A20–A14 of MMSA page 2; enable the page. Translate A20–A14 of MMSA page 3; enable the page. Translate A20–A14 of MMSA page 4; enable the page. Translate A20–A14 of MMSA page 5; enable the page. Translate A20–A14 of MMSA page 6; enable the page. Translate A20–A14 of MMSA page 7; enable the page.
IOW IOW	022h 023h	71h 1010 0101	Set memory type for MMSA pages: 0–1 on-board, 2–3 PCMCIA.
IOW IOW	022h 023h	72h 0000 1010	Set memory type for MMSA pages: 4–5 PCMCIA, 6–7 ROM DOS.
IOW IOW	022h 023h	A8h xx00 00xx	Select socket: MMSA pages 2–5 mapped to socket A.
IOW IOW	022h 023h	65h x1xx xxxx	Enable MMSA.
IOW IOW	022h 023h	74h xxxx xx01	Enable MMSB and select MMSB for programming.
IOW IOW	022h 023h	6Ch xxxx 0x00	Set all translated MMSB A23s (page 2 not translated).
IOW IOW	022h 023h	B5h xx00 xxxx	Initialize CA25 and CA24 for MMSA page 2.
IOW IOW	022h 023h	6Eh 10xx 0000	Set A22 and A21 for MMSB pages 0–3 (except page 2).
IOW IOW IOW IOW	0208h 2208h 4208h 6208h	1111 1110 1111 1111 0xxx xxxx 1000 0000	Translate A20–A14 of MMSB page 0; enable the page. Translate A20–A14 of MMSB page 1; enable the page. Disable translation for this page. Translate A20–A14 of MMSB page 3; enable the page.
IOW IOW	022h 023h	73h 11xx 1010	Set memory type for MMSB pages: 0–1 (PCMCIA), 3 (ROM BIOS), and 2 (not used).
IOW IOW	022h 023h	A9h xxxx xx11	Select socket: MMSB pages 0–1 mapped to socket B.
IOW IOW	022h 023h	B7h xxxx xx00	Initialize CA24 and CA25 for MMSB page 0.

## 2.5 PCMCIA CONTROLLER

The ÉlanSC300 microcontroller's PCMCIA controller supports two sockets (A and B), which are functionally compatible with the *PCMCIA 2.1 Card Interface Specifications*. However, some pins on the chip are shared with other PCMCIA-type cycles like ISA. Refer to Table 2-9 on page 2-16 for details.

Table 2-9 ÉlanSC300 Microcontroller PCMCIA Signal Compatibility

PCMCIA 2.1 Signals	Equivalent CPU Signals (Slot A)	Equivalent CPU Signals (Slot B)	Comments
A25–A0	C25–C24 A23–A12 (MA11–MA0) SA11–SA0	C25–C24 A23–A12 (MA11–MA0) SA12–SA0	Address Bus—Common to all devices on the system except C25–C24, which are exclusive to PCMCIA.
D15–D0	D15–D0	D15–D0	Data Bus—Common to all devices on the system.
CE1	MCEL_A	MCEL_B	Card Enable Even Byte—Exclusive PCMCIA signal.
CE2	MCEH_A	MCEH_B	Card Enable Odd Byte—Exclusive PCMCIA signal.
OE <sup>2</sup>	MEMR	MEMR	Output Enable—Shared ISA signal.
$\overline{WE}$ (PGM) <sup>3</sup>	MEMW	MEMW	Write Enable (Program)—Shared ISA signal.
RDY (BSY)/IRQ	RDY_A	RDY_B	Ready (Busy)/Interrupt Request—Exclusive PCMCIA signal.
$\overline{CD1}$ , $\overline{CD2}$	CD_A	CD_B	Card Detect Right Side (Pin 36)—Exclusive PCMCIA signal. Logical AND of $\overline{CD1}$ and $\overline{CD2}$ should be performed on the system and then fed back as $\overline{CD\_A}$ to the ÉlanSC300 microcontroller's CPU. This mechanism saves one pin.
$\overline{CD1}$ , $\overline{CD2}$	CD_A	CD_B	Card Detect Left Side (Pin 67)—Exclusive PCMCIA signal. Logical AND of $\overline{CD1}$ and $\overline{CD2}$ should be performed on the system and then fed back as $\overline{CD\_B}$ to the ÉlanSC300 microcontroller. This mechanism saves one pin.
WP/IOIS16	WP_A ( $\overline{IOS16\_A}$ )	WP_B ( $\overline{IOS16\_B}$ )	Write Protect/I/O IS 16-Bit Port—Exclusive PCMCIA signal.
REG	REG_A	REG_B	Attribute Memory Select—Exclusive PCMCIA signal.
BVD1/STSCHG	BVD1_A (STSCHG_A)	BVD1_B (STSCHG_B)	Battery Voltage Detect/Status Changed—Exclusive PCMCIA signal.

**Notes:**

1. The ÉlanSC300 microcontroller's PCMCIA Controller provides one VPP pin per socket. These pins do not supply the actual voltage to the card, but provide a mechanism to the system designer for voltage control. If two VPP pins are required per slot, use the PGP pins that are provided by the ÉlanSC300 microcontroller.
2. The PCMCIA  $\overline{OE}$  function may be configured to use the parallel port's  $\overline{SLCTIN}$  pin instead of being shared with the ISA  $\overline{MEMR}$  pin.
3. The PCMCIA  $\overline{WE}$  (PGM) function may be configured to use the parallel port's  $\overline{INIT}$  pin instead of being shared with the ISA  $\overline{MEMW}$  pin.

Table 2-9 ÉlanSC300 Microcontroller PCMCIA Signal Compatibility (Continued)

PCMCIA 2.1 Signals	Equivalent CPU Signals (Slot A)	Equivalent CPU Signals (Slot B)	Comments
BVD2/SPKR	BVD2_A (STSCHG_A)	BVD2_B (STSCHG_B)	Battery Voltage Detect/Audio Digital Waveform—Exclusive PCMCIA signal.
RESET	RST_A	RST_B	Card Reset—Exclusive PCMCIA signal.
WAIT	WAIT_AB	WAIT_AB	Extend Bus Cycle—Exclusive PCMCIA signal. The wait signal is intended to be used for both slots. The individual wait signals from both slots need to be logically ANDed to form the <u>WAIT_AB</u> signal before it goes in the ÉlanSC300 microcontroller.
IORD	IOR	IOW	I/O Read—Shared ISA signal.
IOWR	IOR	IOW	I/O Write—Shared ISA signal.
INPACK	(Not available)	(Not available)	Input Acknowledge—(Not Implemented in the ÉlanSC300 microcontroller's CPU.)

**Notes:**

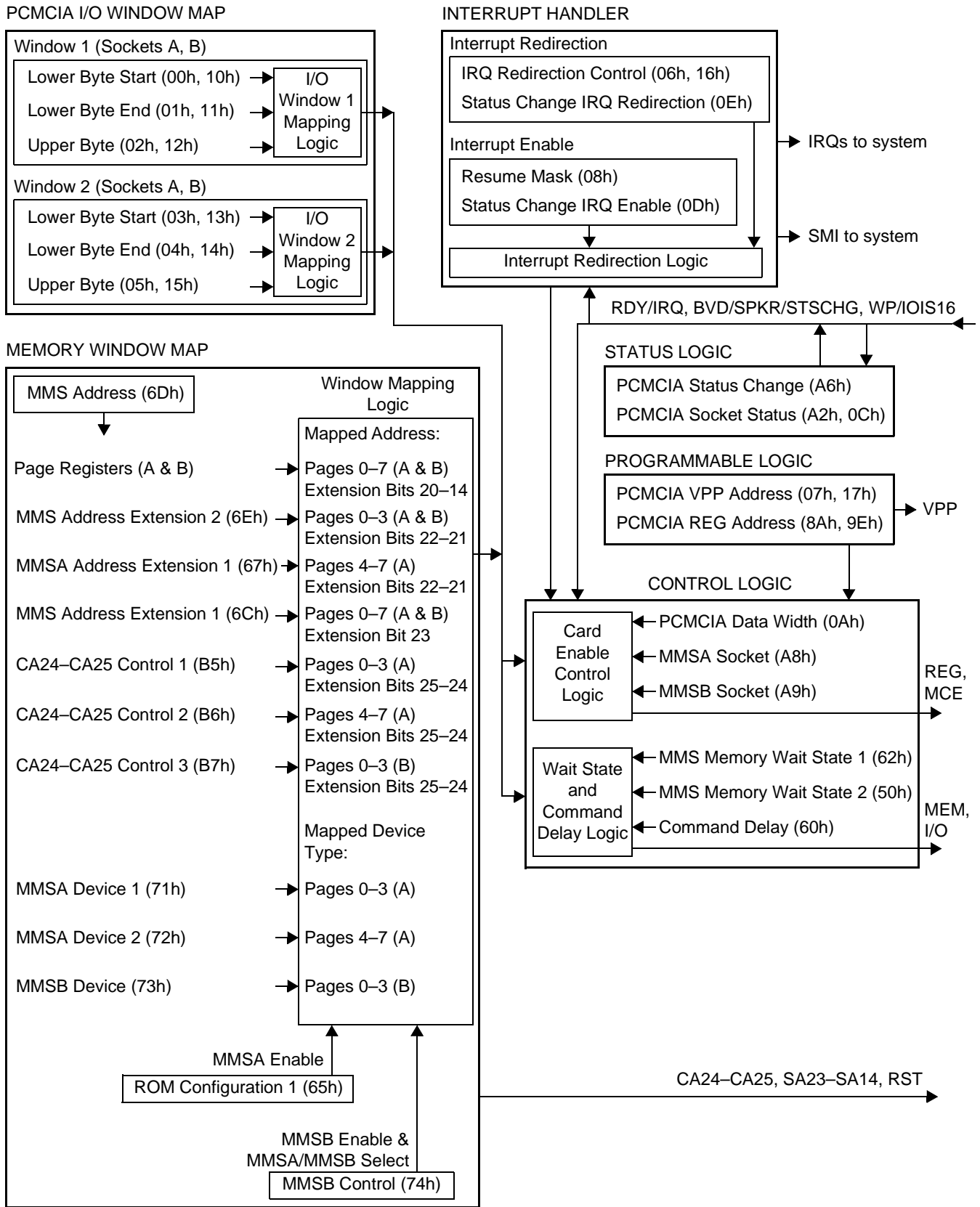
1. The ÉlanSC300 microcontroller's PCMCIA Controller provides one VPP pin per socket. These pins do not supply the actual voltage to the card, but provide a mechanism to the system designer for voltage control. If two VPP pins are required per slot, use the PGP pins that are provided by the ÉlanSC300 microcontroller.
2. The PCMCIA  $\overline{OE}$  function may be configured to use the parallel port's  $\overline{SLCTIN}$  pin instead of being shared with the ISA  $\overline{MEMR}$  pin.
3. The PCMCIA  $\overline{WE}$  (PGM) function may be configured to use the parallel port's  $\overline{TNIT}$  pin instead of being shared with the ISA  $\overline{MEMW}$  pin.

The ÉlanSC300 microcontroller's PCMCIA controller can logically be divided into six functional blocks. These functional blocks are as follows:

- Memory-window map
- I/O-window map
- Interrupt handler
- Control logic
- Status logic
- Programmable logic

Please refer to Figure 2-8 on page 2-18 for a pictorial view.

Figure 2-8 PCMCIA Controller Block Diagram



## 2.5.1 Memory-Window Map

The ÉlanSC300 microcontroller's Memory Management System (MMS) provides an access mechanism to map chunks of the 64 Mbyte of PCMCIA common/attribute memory into the system address space. Each of these chunks of memory is mapped through pages in the two MMS windows. See “Memory Mapping” on page 2-10 for detailed information about the two MMS windows, MMSA and MMSB.

MMS page-mapping logic translates pages in system-memory address space to an equivalent-size page in the PCMCIA card memory. The entire 64 Mbyte of the PCMCIA-memory address space can be accessed using this mechanism. Each of these pages can be mapped to either PCMCIA socket A or B. For information, see “MMSA Socket Register (Index A8h)” on page 5-72 and “MMSB Socket Register (Index A9h)” on page 5-73. Each socket can be independently configured for either 8-bit or 16-bit accesses (see “PCMCIA Data Width Register (Index 0Ah)” on page 5-17).

Selection of a PCMCIA card's attribute versus common memory is controlled by the PCMCIA  $\overline{\text{REG}}$  signal. The  $\overline{\text{REG}}$  signal for each of the ÉlanSC300 microcontroller's PCMCIA sockets is controllable on a per-socket basis. For more information on controlling the  $\overline{\text{REG}}$  signal, see “Programmable Logic” on page 2-21.

A PCMCIA memory card is accessed (or a window hit occurs) when the following conditions are met:

- The system-memory address is greater than or equal to the start address of one of the defined pages in either MMSA or MMSB.
- The system-memory address is less than or equal to the end address of one of the defined pages in either MMSA or MMSB.
- Corresponding to the page of the above address, either MMSA or MMSB is enabled and configured for PCMCIA socket A or B.
- The corresponding socket to the above window is configured as a memory card.

The memory-window mapping logic detects this condition and then translates the address into a 26-bit address for the PCMCIA card. The translated address is formed through the mapping registers listed in Table 2-10 on page 2-19.

Table 2-10 Memory-Window Mapping Registers

Register Name	Address Bits
CA24–CA25 Control 1–3 (indexes B5–B7h)	25–24
MMS Address Extension 1 (Index 6Ch)	23
MMS Address Extension 2 (Index 6Eh) MMSA Address Extension 1 (Index 67h)	22–21
MMS Address register (Index 6Dh)	20–14
(Least significant 14 bits are passed unchanged to the PCMCIA card)	13–0

In addition to the address translation, memory-mapping logic also sends a window-hit signal to the control-logic block so that proper control signals are issued to the PCMCIA interface.

## 2.5.2 I/O Window Map

The ÉlanSC300 microcontroller's PCMCIA Controller supports both 8-bit and 16-bit memory and I/O cards. Two I/O windows for each socket (A and B) are provided. These windows are programmable for any size between 1 and 256 bytes, but they cannot cross the 256-byte address boundary. Each individual window in sockets A and B can be enabled or disabled through software. If the slot is configured for memory only, then both I/O windows for that socket are automatically disabled regardless of the state of the individual window enables. If the sockets are enabled for I/O accesses, memory cycles will still propagate properly across the PCMCIA interface.

A start and end address for each window needs to be programmed. The upper 8 bits of the I/O window's start and stop addresses are identical and are programmed through a single index register per I/O window (refer to the PCMCIA I/O Window Upper Byte registers at indexes 02h, 05h, 12h, and 15h); likewise, the lower 8 bits of the I/O window's start and end addresses are programmed through two separate index registers per I/O window (refer to the PCMCIA I/O Window Lower Byte Start and End registers at indexes 00–01h, 03–04h, 10–11h, and 13–14h). This is why the maximum I/O window size is 256 bytes, and the I/O window cannot cross a 256-byte address boundary.

An I/O card is accessed (or an I/O window-hit occurs), when the following conditions are met:

- The lower byte of the system I/O address is greater than or equal to the low-byte start address of one of the defined windows in either socket A or socket B.
- The lower byte of the system I/O address is less than or equal to the low-byte end address of the same window as defined above.
- The upper byte of the system I/O address is equal to the high-byte address of the same window.
- Corresponding to the window above, the I/O window-enable bit is set.
- The corresponding socket to the above window is configured as an I/O card.

The I/O window-mapping logic detects this condition and then sends a window-hit signal to the control-logic block so that proper control signals are issued to the PCMCIA interface. No address translation takes place for the I/O window hit. The system address is passed directly to the PCMCIA sockets.

## 2.5.3 Control Logic

This block enables the control lines to both PCMCIA sockets. When a memory or I/O window-hit is detected, the control logic generates the  $\overline{MCEL}$  and  $\overline{MCEH}$  signals, the  $\overline{REG}$  signal, the memory command signals ( $\overline{MEMR}$  and  $\overline{MEMW}$ ), and the I/O command signals ( $\overline{IOR}$  and  $\overline{IOW}$ ) for the socket being accessed.

The PCMCIA cycle types, such as 8-bit versus 16-bit cycles, are controlled through the PCMCIA Data Width register at Index 0Ah. The ÉlanSC300 microcontroller's PCMCIA controller also supports programmable wait states and command delays. See "Wait States and Command Delays" on page 2-23 for more information. All the above items—along with the  $\overline{WAIT}$  signal from the PCMCIA card—have an effect on the length and type of the PCMCIA cycle performed.



## 2.5.4 Status Logic

This logic holds the current status of various pins on both sockets A and B. The following registers hold this information:

- PCMCIA Socket B Status register at Index 0Ch
- PCMCIA Socket A Status register at Index A2h
- PCMCIA Socket Status Change register at Index A6h

All of these registers are read only. However, the PCMCIA Socket Status Change register can be cleared through software.

The information in these registers is provided to the interrupt handler that services changes in the status of the PCMCIA card in order to determine which event caused the interrupt based on pin changes on the card interface. For example, a falling BVD signal can be programmed to generate an interrupt to indicate that a memory-card battery is getting low. Once an interrupt has been generated due to pin changes, software must clear the corresponding status bits in the PCMCIA Status Change register at Index A6h so that other interrupts can take place.

## 2.5.5 Programmable Logic

The programmable logic has two purposes:

- It generates the voltage selects for the cards through the VPP pins.
- It provides a means to program the  $\overline{REG}$  pin for attribute-memory accesses.

These pins are programmable through software.

Only one VPP control pin per socket is available. A programmable I/O address is provided for controlling the state of the VPP control pin. Any write to that programmable I/O address changes the state of the VPP control pin to the state of data-bus bit 0. If two programmable VPP voltage-control pins are required per slot, use the Programmable General-Purpose (PGP) pins that are provided in the ÉlanSC300 microcontroller. The PCMCIA VPPA and VPPB Address registers at indexes 07h and 17h are used to program the I/O address used for VPP control.

The  $\overline{REG}$  pin is also controlled in a similar manner to the VPP control pin. The  $\overline{REG}$  pin is asserted for I/O commands and is kept High during non-attribute-memory accesses. During attribute-memory accesses,  $\overline{REG}$  should be asserted for the memory commands as well. Therefore, programming the  $\overline{REG}$  pin Low and then generating a memory cycle allows access to the attribute memory. The PCMCIA REGA and REGB Address registers at indexes 8Ah and 9Eh are used to program the I/O address used to control the  $\overline{REG}$  pins.

Note that the  $\overline{REG}$  pin is not controllable on a per-page basis.  $\overline{REG}$  is supported on a per-socket level only. This may have implications for PCMCIA driver software.

## 2.5.6 Interrupt Handler

There are two different kinds of PCMCIA interrupts that can generate a system interrupt.

The first kind of interrupt is one that can be generated from the PC card through the IREQ line. These interrupts can happen only when the card is configured for I/O. A mechanism is provided by the ÉlanSC300 microcontroller's PCMCIA controller to redirect this interrupt to one of the system interrupts. Refer to the PCMCIA I/O Card IRQ Redirection Control A and B registers at indexes 06h and 16h. Also note that it is a programmable option to invert the state of the card's IREQ line as it is directed to one of the system interrupts.

The other kind of interrupt is comprised of those interrupts that can be generated internally in the ÉlanSC300 microcontroller based on the status changes on different pins from the card. These interrupts can occur for both memory and I/O cards. A mechanism for redirecting similar interrupts is also provided. Refer to the PCMCIA Status Change IRQ Redirection register at Index 0Eh. It is also possible to redirect these interrupts to generate an SMI instead of one of the standard system interrupts. These status-change interrupts are very useful for taking the prompt actions that are necessary under certain conditions.

Card status changes that can generate an interrupt are:

- Card inserted or removed ( $\overline{CD}$  pin)
- Card now available for access (RDY pin)
- Card status changed (STSCHG pin)
- Ring Indicate signal (RI pin)
- Battery going low (BVD pins)

Each of the above interrupts can individually be enabled or disabled through software. Once an interrupt is generated due to a status change on any of the above pins, software must clear that status-change pin in order for any more interrupts to be generated.

## 2.6 OTHER MEMORY CONTROLLER INFORMATION

### 2.6.1 ROM Chip-Select Command Gating

The  $\overline{\text{DOSCS}}$  and  $\overline{\text{ROMCS}}$  chip-select outputs of the ÉlanSC300 microcontroller are, by default, internally gated with the memory-read command ( $\overline{\text{MEMR}}$ ) or the memory-write command ( $\overline{\text{MEMW}}$ ). In the ÉlanSC300 microcontroller, the following configuration register bits may be used to disable the command gating and allow the  $\overline{\text{DOSCS}}$  and  $\overline{\text{ROMCS}}$  signals to be available as address decodes only:

- **Bit 2 of the Miscellaneous 5 register at Index B3h** Enables the  $\overline{\text{ROMCS}}$  signal as an address decode as follows:
  - 0 Address decode with command gating
  - 1 Address decode only
- **Bit 4 of the ROM Configuration 3 register at Index B8h** Enables the  $\overline{\text{DOSCS}}$  signal as an address decode as follows:
  - 0 Address decode with command gating
  - 1 Address decode only

When the CPU clock is stopped, the  $\overline{\text{ROMCS}}$  and  $\overline{\text{DOSCS}}$  chip-selects are forced High.

### 2.6.2 Wait States and Command Delays

The ÉlanSC300 microcontroller provides several programmable options for controlling the number of wait states and command delays inserted into a cycle. This section covers wait states and command delays for ROM-BIOS, ROM-DOS, PCMCIA, and ISA cycles.

A command delay is inserted between the point in a memory or I/O cycle where the address is placed on the bus, and the point where the memory or I/O, read or write, command signal is asserted. This delay gives slower devices extra time to decode the address. Command delays do not lengthen the overall cycle time. That is, the command signal is deasserted at the same time it normally is (determined by the number of wait states) without the command delay. The net effect is a shortened command time.

Table 2-11 on page 2-24 documents the duration of the command delay for the various cycles.

Table 2-11 Command Delay Duration for Various Cycles

Cycle Type	Command Delay Duration (in SYSCLK Cycles)
8-bit ISA memory	1 or 0.5 (determined by bit 2 of the Command Delay register at Index 60h)
8-bit PCMCIA memory	0, 0.5, or 1 (determined by bit 2 of the Command Delay register and bits 5 and 7 of the MMS Memory Wait State 2 register at Index 50h)
8-bit ROM DOS memory	0, 0.5, or 1 (determined by bit 2 of the Command Delay register and bits 2 and 6 of the MMS Memory Wait State 2 register)
8-bit ROM BIOS memory	0.5
16-bit memory (all)	0
8-bit external I/O (0100–03FFh)	0.5, 1, 2 (determined by bits 0 and 1 of the Command Delay register)
8-bit internal I/O (000–0FFh)	0.5
16-bit external I/O (0100–03FFh)	0.5

Wait states extend the amount of time the read- or write-command signal is asserted for memory or I/O accesses. This has the effect of increasing the total length of the cycle. Table 2-12 on page 2-24 documents the number of wait states for the various ROM-DOS, ROM-BIOS, PCMCIA, and ISA cycles.

Table 2-12 Wait States for Various Cycles

Cycle Type	Number of Wait States (in SYSCLK Cycles)
8-bit ISA memory 8-bit ROM BIOS (MMS-accessed)	2, 3, 4, or 5 (determined by bits 0 and 1 of the MMS Memory Wait State 1 register at Index 62h)
8-bit PCMCIA memory	1, 2, 3, 4, or ISA setting (determined by bits 5–3 of the MMS Memory Wait State 2 register at Index 50h)
8-bit ROM DOS (MMS accessed and linearly decoded)	1, 2, 3, 4, or ISA setting (determined by bits 2–0 of the MMS Memory Wait State 2 register)
8-bit ROM BIOS (not MMS accessed)	2 or 3 (determined by bits 7 and 4 of the Command Delay register at Index 60h)
16-bit ROM DOS, ROM BIOS, PCMCIA, and ISA memory	1, 2, 3, or 4 (determined by bits 3 and 2 of the MMS Memory Wait State 1 register)
8-bit internal I/O (000–0FFh)	2 or 4 (determined by bit 3 of the Wait State Control register at Index 63h)
8-bit floppy-disk-drive I/O (3F0–3F7h)	2, 3, 4, or 5 (determined by bits 1 and 0 of the I/O Wait State register at Index 61h)
8-bit hard-drive I/O (1F0–1F7h)	2, 3, 4, or 5 (determined by bits 3 and 2 of the I/O Wait State register)
Other 8-bit I/O (100–3FFh)	2, 3, 4, or 5 (determined by bits 5 and 4 of the I/O Wait State register)
16-bit I/O	3 or 4 (determined by bit 2 of the Wait State Control register)

### 2.6.3 High-Speed Clock ROM Cycles

To improve the ROM-access times, an option is provided so that accesses using the  $\overline{\text{ROMCS}}$  or  $\overline{\text{DOSCS}}$  chip-selects may run at the high-speed CPU clock rate rather than the low-speed CPU clock rate of 9.2 MHz.

### 2.6.4 ROM Chip-Select Signal

The high-speed CPU clock rate is enabled for the ROM Chip-Select ( $\overline{\text{ROMCS}}$ ) signal, and unique wait-state controls for each ROM chip-select may be programmed through the Miscellaneous 5 register at Index B3h.

- **Bit 6 of the Miscellaneous 5 register at Index B3h** Enables  $\overline{\text{ROMCS}}$  ROM accesses to run at the high-speed CPU clock rate as follows:
  - 0 Disabled (default)
  - 1 Enabled
- **Bits 5 and 4 of the Miscellaneous 5 register at Index B3h** Control the number of wait states for fast  $\overline{\text{ROMCS}}$  cycles (see Table 2-13 on page 2-25).

Table 2-13  $\overline{\text{ROMCS}}$  Wait-State Control-Bit Logic

Index B3h Bit 5	Index B3h Bit 4	Wait States
0	0	4
0	1	3
1	0	2
1	1	1

Note that if the  $\acute{\text{E}}\text{lanSC300}$  microcontroller is in its Maximum-ISA-Bus mode, the BALE output will not be generated for high-speed  $\overline{\text{ROMCS}}$  cycles. It is recommended that fast  $\overline{\text{ROMCS}}$  cycles not be enabled in systems where the target  $\overline{\text{ROMCS}}$  device is an 8-bit device, and other devices in the system assert  $\overline{\text{MCS16}}$ . This is because the  $\overline{\text{MCS16}}$  timing is violated when the ISA bus is running at the high-speed PLL frequency. For more details, refer to the  *$\acute{\text{E}}\text{lanSC300}$  Microcontroller ISA Bus Anomalies Application Note*, available from the AMD engineering support staff.

### 2.6.5 DOS Chip-Select Signal

The CPU clock rate is enabled for the DOS Chip-Select ( $\overline{\text{DOSCS}}$ ) signal, and unique wait-state controls for each ROM chip-select may be programmed through the ROM Configuration 3 register at Index B8h.

- **Bit 7 of the ROM Configuration 3 register at Index B8h** Enables  $\overline{\text{DOSCS}}$  ROM accesses to run at the high-speed CPU clock rate as follows:
  - 0: Disabled (default)
  - 1 Enabled
- **Bits 6 and 5 of the ROM Configuration 3 register at Index B8h** Control the number of wait states for fast  $\overline{\text{DOSCS}}$  cycles (see Table 2-14 on page 2-26).

Table 2-14  $\overline{\text{DOSC}}\overline{\text{S}}$  Wait-State Control-Bit Logic

Index B8h Bit 6	Index B8h Bit 5	Wait States
0	0	4
0	1	3
1	0	2
1	1	1

Note that if the ÉlanSC300 microcontroller is in its Maximum-ISA-Bus mode, the BALE output will not be generated for high-speed  $\overline{\text{DOSC}}\overline{\text{S}}$  cycles.

Bit 1 of the ROM Configuration 2 register at Index 51h should be 1 when enabling fast 16-bit  $\overline{\text{DOSC}}\overline{\text{S}}$  cycles. It is recommended that fast  $\overline{\text{DOSC}}\overline{\text{S}}$  cycles not be enabled in systems where the target  $\overline{\text{DOSC}}\overline{\text{S}}$  device is an 8-bit device, and other devices in the system assert  $\overline{\text{MCS}}\overline{16}$ . This is because the  $\overline{\text{MCS}}\overline{16}$  timing is violated when the ISA bus is running at the high-speed PLL frequency. For more details, see the *ÉlanSC300 Microcontroller ISA Bus Anomalies Application Note*, available from the AMD engineering support staff.

### 2.6.6 Self-Refresh DRAMs

Self-refreshing DRAMs are supported in the ÉlanSC300 microcontroller as follows:

- Bit 3 of the Miscellaneous 5 register at Index B3h enables Self-Refresh mode when the PMU changes to a mode that causes the CPU clock to stop as follows:
  - 0 Disabled (default)
  - 1 Enabled
- Upon exiting the stop clock, the system logic forces one CAS-before-RAS refresh cycle before the normal CAS-before-RAS refresh logic takes control.
- If a complete burst row refresh is required by the DRAM, the ÉlanSC300 microcontroller will not directly support this.

### 2.6.7 80-ns DRAM Support

The ÉlanSC300 microcontroller supports 80-ns DRAMs when running at 25 MHz.



The ÉlanSC300 microcontroller's integrated video controller implements a register super-set of the industry-standard CGA and Hercules Graphics Adapter (HGA) display-system architecture. Control registers are mapped into the CPU's I/O address space. Display data and fonts reside in an external 32-Kbyte SRAM, which is accessed directly by the CPU to update video information and by the video controller to refresh the display. The controller supports both graphics and text modes of operation.

CGA was originally designed to work on a 640×200 display, but this video controller is versatile enough to work with LCD panels that have other resolutions (480×320, 320×240, and 720×348). The examples in this document will refer to panel sizes of 640×200 and 480×320, but other sizes can be supported.

When using a nonstandard-size LCD (other than 640×200), care must be taken as to how memory is accessed, and, therefore, as to how data must be stored. In 80×25 text mode, CGA expects to provide 80 columns of characters across the display and 25 rows (640×200 pixels and an 8×8 font implies  $640 \div 8 = 80$  and  $200 \div 8 = 25$ ). Memory is accessed contiguously; the character byte that follows memory address 0B809Eh (character 79 on row 0) is displayed as character 0 on row 1. With a 480×320 LCD panel, the display shows 60×40 characters (480×320 pixels and an 8×8 font implies  $480 \div 8 = 60$  and  $320 \div 8 = 40$ ). Therefore, the end of row 0 is at 0B8076h, and the byte in 0B8078h is the first character on row 1. Each character in text mode is defined by 2 bytes (the character and attribute bytes), so 0003h is multiplied by 2 and added to 0B8000h). The same applies to graphics modes, but on pixel boundaries instead of character boundaries.

### 3.1 SIGNAL INTERFACES

The ÉlanSC300 microcontroller's internal video controller has three interfaces:

- A memory interface (for the external 32-Kbit×8-bit SRAM)
- A display interface to interact with the display (LCD panel)
- A third interface for miscellaneous signals

#### 3.1.1 Memory Interface

The ÉlanSC300 microcontroller's memory interface consists of the signals shown in Table 3-1 on page 3-2.

Table 3-1 Memory Interface Signals

Signal	Function
DSM14–DSMA0	Address bus for the SRAM
DSMD7–DSMD0	Data bus for the SRAM
DSCE	SRAM chip-enable
DSOE	SRAM output-enable
DSWE	SRAM write-enable

### 3.1.2 Display Interface

The display interface of the ÉlanSC300 microcontroller consists of the signals shown in Table 3-2 on page 3-2.

Table 3-2 Display Interface Signals

Signal	LCD Function Bit 4 of Video Index 18h = 1
M	AC modulation
FRM/VDRV	Frame pulse
CP1/HDRV	Line clock
CP2/VDO	Shift clock for the LCD
LCDD0/R	Data bit 0 (upper panel bit 0)
LCDD1/G	Data bit 1 (upper panel bit 1)
LCDD2/B	Data bit 2 (upper panel bit 2)
LCDD3/I	Data bit 3 (upper panel bit 3)
$\overline{TC}S16/LCDDL0$	Data bit 4 (lower panel bit 0)
$\overline{MC}S16/LCDDL1$	Data bit 5 (lower panel bit 1)
IRQ14/LCDDL2	Data bit 6 (lower panel bit 2)
$\overline{SB}HE/LCDDL3$	Data bit 7 (lower panel bit 3)

The display-data bits are used for the LCD in the following manner:

- Single-scan panel with 4 bits of data (bits 6 and 7 of Video Index 18h are 01b): LCDD3–LCDD0.
- Single-scan panel with 8 bits of data (bits 6 and 7 of Video Index 18h are 11b): LCDD3–LCDD0 and LCDDL3–LCDDL0.
- Dual-scan panel (bits 6 and 7 of Video Index 18h are 00b or 10b): LCDD3–LCDD0 (upper panel), LCDDL3–LCDDL0 (lower panel). For information, see the description for bit 1 in “Function Enable 2 Register (Index B1h)” on page 5-78.



### 3.1.3 Miscellaneous Signals Interface

The miscellaneous signals of the ÉlanSC300 microcontroller are listed in Table 3-3 on page 3-3.

The  $\overline{\text{LVDD}}$  and  $\overline{\text{LVEE}}$  pins are designed to sequence the power to an LCD display.  $\overline{\text{LVDD}}$  controls the panel logic voltage,  $\overline{\text{LVEE}}$  the contrast voltage. Unlike the PMC pins, these control pins are hardwired for a specific function and are not programmable. A logic 0 on one of these pins means power should be switched on. When the external reset pin  $\overline{\text{RESIN}}$  is held active,  $\overline{\text{LVDD}}$  and  $\overline{\text{LVEE}}$  are forced to logic 1 (power off). One refresh cycle after  $\overline{\text{RESIN}}$  is driven inactive,  $\overline{\text{LVDD}}$  changes to logic 0 (power on).  $\overline{\text{LVEE}}$  changes to logic 0 on the next refresh after  $\overline{\text{LVDD}}$  changes.

When the PMU enters Sleep mode, or if the video PLL is turned off in Doze mode,  $\overline{\text{LVEE}}$  will change to logic 1, disconnecting the negative voltage.  $\overline{\text{LVDD}}$  will remain active until the PMU reaches Suspend mode. When waking up from Suspend mode (i.e., entering High-Speed PLL mode),  $\overline{\text{LVDD}}$  will precede  $\overline{\text{LVEE}}$  by one refresh cycle.

The LCD panel data and control signals are all forced to logic 0 in the PMU modes that are programmed to disable the low-speed and video PLLs through bits 7 and 3 of the Power Control 1 and 2 registers at indexes 80h and 81h.

When using  $\overline{\text{LVDD}}/\overline{\text{LVEE}}$  to control voltage on an LCD display, do not allow the BIOS to force the PMU directly into Suspend mode by writing to the Software Mode Control register at Index 88h. Doing this causes  $\overline{\text{LVDD}}$  and  $\overline{\text{LVEE}}$  to switch off simultaneously, which violates the correct power sequencing for most LCD panels. If a forced shutdown is desired, force the PMU into Sleep mode instead, and allow the PMU timer to sequence the PMU into Suspend mode.

Some LCD panels may require a maximum time between switching off  $\overline{\text{LVEE}}$  and  $\overline{\text{LVDD}}$ . In this case, the Sleep mode transition timer should be programmed to fall within this time.

Table 3-3 Miscellaneous Signals

Signal	Function
$\overline{\text{LVDD}}$	Disable/enable signal for the LCD panel's VDD power (usually +5 V or +3.3 V)
$\overline{\text{LVEE}}$	Disable/enable signal for the LCD panel's VEE power (the high voltage, usually –20 V or +20 V to +40 V)

### 3.2 DISPLAY MEMORY

The ÉlanSC300 microcontroller uses an external 32-Kbit×8-bit SRAM for video information storage. When configured for CGA mode (the default), the SRAM is addressed in the range from 0B8000–0BFFFFh. If configured for HGA mode by setting bit 0 of the Screen Control register at Index 18h, the SRAM is addressed in the range from 0B0000–0B7FFFh. For HGA mode, the SRAM is repeated from 0B8000–0BFFFFh. For the following discussions, assume the video SRAM decodes addresses 0B8000–0BFFFFh. Table 3-4 on page 3-4 shows the DSMA mapping for video-SRAM accesses. All video accesses in the ÉlanSC300 microcontroller occur between the addresses of 0B0000h and 0BFFFFh on the ISA bus.

Table 3-4 SRAM Address Mapping during CPU Access to Video Memory

ISA-Bus Address Signal	SRAM Address Signal
SA0	DSMA0
SA1	DSMA1
SA2	DSMA2
SA3	DSMA3
SA4	DSMA4
SA5	DSMA5
SA6	DSMA6
SA7	DSMA7
SA8	DSMA8
SA9	DSMA9
SA10	DSMA10
SA11	DSMA11
SA12	DSMA12
SA13	DSMA13
SA14	DSMA14
SA15 (= 1)	–
SA23–SA16 (= 0Bh)	–

### 3.3 GRAPHICS MODE

In graphics mode (also called all-points-addressable or APA mode), the ÉlanSC300 microcontroller’s video-memory bits directly represent display pixels; there is no font. The use of either 1 or 2 bits per pixel (bpp) defines the color depth of the image.

The use of 1 bit per pixel (or 640-column mode) means each byte in memory holds 8 pixels of information—each pixel is either on (the bit in memory is 1) or off (the bit is 0). Bit 7 of memory location 0B8000h is displayed as the upper left pixel of the display, bit 6 is the pixel to its right, and so forth. Bit 7 of memory location 0B8001h is 9 pixels from the left on row 0, and so on. The use of 1 bpp implies only two colors or gray scales on the display at a time, but this does not have to be black and white; any one of 16 colors is selected by bits 4–0 of the CGA Color Select register at Port 3D9h for the on pixels.

The use of 2 bits per pixel (or 320-column mode) means each byte in memory holds 4 pixels of information. Each pixel on the display can be one of four colors or gray scales. Bits 7 and 6 of memory location 0B8000h are displayed as the upper left pixel of the display. A 640×200 LCD panel can still display 320-column graphics mode even though each pixel on the panel is a physical location whose size cannot change; each pixel that is displayed is made up of two physical pixels on the panel. For information, see the description for bit 5 in “Control 1 Register (Video Index 20h)” on page 3-34.

Consistent with CGA-graphics-mode conventions, the memory is made up of more than one bank—each row of pixels on the display is stored in alternating banks. The number of banks needed depends on the resolution of the display being used. Each bank is 8 Kbyte deep.

A 640×200 LCD panel requires 2 banks of memory:

- 640 pixels · 200 pixels = 128,000 pixels
- 128,000 pixels ÷ 8 pixels per byte = 16,000 bytes
- 16,000 bytes ÷ 2 banks = 8000 bytes
- 8,000 bytes is less than 8 Kbyte (8 · 1024 = 8,192), so the information fits in only 2 banks.

A 480×320 LCD panel requires 4 banks of memory:

- 480 pixels · 320 pixels = 153,600 pixels
- 153,600 pixels ÷ 8 pixels per byte = 19,200 bytes
- 19,200 bytes ÷ 2 banks = 9,600 bytes
- 9,600 bytes is greater than 8 Kbyte, so the information needs all 4 banks.

**Note:** When using a 640×200 LCD panel with 2 banks of memory, only 16 Kbyte of the SRAM's 32 Kbyte are used. When using a 480×320 LCD with 4 banks of memory, all 32 Kbyte are utilized.

Table 3-5 640×200 and 320×200 Graphics-Mode Video-Memory Data

320×200 (2 bpp)		Pixel 0–Pixel 319		
640×200 (1 bpp)		Pixel 0–Pixel 639		
Display pixel row 0	0B8000h	<b>First Bank</b>	0B804Fh	8 Kbyte
Display pixel row 2	0B8050h		0B809Fh	
Display pixel row 4	0B80A0h	Even-numbered pixel rows	0B80EFh	
.	.	(0, 2, ..., 198)	.	
.	.	.	.	
.	.	.	.	
Display pixel row 198	0B9EF0h	<b>80 bytes each row</b>	0B9F3Fh	
	0B9F40h	Unused (192 bytes)	0B9FFFh	
Display pixel row 1	0BA000h	<b>Second Bank</b>	0BA04Fh	8 Kbyte
Display pixel row 3	0BA050h		0BA09Fh	
Display pixel row 5	0BA0A0h	Odd-numbered pixel rows	0BA0EFh	
.	.	(1, 3, ..., 199)	.	
.	.	.	.	
.	.	.	.	
Display pixel row 199	0BBEF0h	<b>80 bytes each row</b>	0BBF3Fh	
	0BBF40h	Unused (192 bytes)	0BBFFFh	

Table 3-6 640×200 Graphics-Mode Video-Memory Data for a 480×320 LCD Panel

480×320 (1 bpp)		Pixel 0–Pixel 479	
Display pixel row 0	0B8000h	<b>First Bank</b>	0B803Bh
Display pixel row 4	0B803Ch		0B8077h
Display pixel row 8	0B8078h	Pixel rows 0, 4, ..., 316	0B80B3h
.	.	.	
.	.	.	
Display pixel row 316	0B9284h	<b>60 bytes each row</b>	0B92BFh
	0B92C0h	Unused (3,392 bytes)	0B9FFFh
8 Kbyte			
Display pixel row 1	0BA000h	<b>Second Bank</b>	0BA03Bh
Display pixel row 5	0BA03Ch		0BA077h
Display pixel row 9	0BA078h	Pixel rows 1, 5, ..., 317	0BA0B3h
.	.	.	
.	.	.	
Display pixel row 317	0BB284h	<b>60 bytes each row</b>	0BB2BFh
	0BB2C0h	Unused (3,392 bytes)	0BBFFFh
8 Kbyte			
Display pixel row 2	0BC000h	<b>Third Bank</b>	0BC03Bh
Display pixel row 6	0BC03Ch		0BC077h
Display pixel row 10	0BC078h	Pixel rows 2, 6, ..., 318	0BC0B3h
.	.	.	
.	.	.	
Display pixel row 318	0BD284h	<b>60 bytes each row</b>	0BD2BFh
	0BD2C0h	Unused (3,392 bytes)	0BDFFFh
8 Kbyte			
Display pixel row 3	0BE000h	<b>Fourth Bank</b>	0BE03Bh
Display pixel row 7	0BE03Ch		0BE077h
Display pixel row 11	0BE078h	Pixel rows 3, 7, ..., 319	0BE0B3h
.	.	.	
.	.	.	
Display pixel row 319	0BF284h	<b>60 bytes each row</b>	0BF2BFh
	0BF2C0h	Unused (3,392 bytes)	0BFFFFh
8 Kbyte			

Because of memory constraints, a 480×320 LCD panel only operates in 1-bpp mode. 32 Kbyte is not enough memory to store 2 bits per pixel as the following calculations show.

■ 1 bit per pixel:

- 480 pixels · 320 pixels = 153,600 pixels
- 153,600 pixels ÷ 8 pixels per byte (1 bpp) = 19,200 bytes
- 19,200 bytes are needed to store 1 bpp. This works because it is less than the 32 Kbyte available in SRAM.

■ 2 bits per pixel:

- 480 pixels · 320 pixels = 153,600 pixels
- 153,600 pixels ÷ 4 pixels per byte (2 bpp) = 38,400 bytes
- 38,400 bytes are needed to store 2 bpp. This is not possible because it is greater than the 32 Kbyte available in SRAM.

### 3.4 TEXT MODE

In the text (or alphanumeric) mode, display characters are represented in memory as a 2-byte data structure: the character byte is followed by the attribute byte. The video memory is partitioned into a 16-Kbyte block of display data, 12–14 Kbyte of unused off-screen free areas, and 2–4 Kbyte of off-screen font areas. (The device does not use a separate font ROM; the font is stored in the video SRAM.) More information on fonts is given later in this chapter. Tables 3-7–3-10 show how the 16-Kbyte SRAM is partitioned for each text mode.

Table 3-7 CGA Text Mode Using Character Fonts 1 Area

From	To	Size	Usage
0B8000h	0BBFFFh	16 Kbyte	Display data
0BC000h	0BCFFFh	4 Kbyte	Free
0BD000h	0BDFFFh	4 Kbyte	Free
0BE000h	0BEFFFh	4 Kbyte	Free
0BF000h	0BF7FFh	2 Kbyte	Free
0BF800h	0BFFFFh	2 Kbyte	Character fonts 1 (8×8)

Table 3-8 CGA Text Mode Using Character Fonts 2 Area

From	To	Size	Usage
0B8000h	0BBFFFh	16 Kbyte	Display data
0BC000h	0BCFFFh	4 Kbyte	Free
0BD000h	0BDFFFh	4 Kbyte	Free
0BE000h	0BEFFFh	4 Kbyte	Free
0BF000h	0BF7FFh	2 Kbyte	Character fonts 2 (8×8)
0BF800h	0BFFFFh	2 Kbyte	Free

Table 3-9 CGA Text Mode Using Special Character Fonts Area

From	To	Size	Usage
0B8000h	0BBFFFh	16 Kbyte	Display data
0BC000h	0BCFFFh	4 Kbyte	Free
0BD000h	0BDFFFh	4 Kbyte	Special character fonts (8×16)
0BE000h	0BEFFFh	4 Kbyte	Free
0BF000h	0BFFFFh	4 Kbyte	Free

Table 3-10 HGA Text Mode

From	To	Size	Usage
0B8000h	0BBFFFh	16 Kbyte	Display data
0BC000h	0BCFFFh	4 Kbyte	Free
0BD000h	0BDFFFh	4 Kbyte	Free
0BE000h	0BEFFFh	4 Kbyte	HGA character fonts (9×14)
0BF000h	0BFFFFh	4 Kbyte	Free

In the display-data portion of memory, the even addresses contain the character bytes (the ASCII code for the character to be displayed at each screen location), which are used by the video controller to point to the correct font in memory to display that character. The odd addresses contain the attribute bytes, which define the way the character is displayed on the screen. Each even-odd combination of bytes represents one character on the screen. For example, video-memory address 0B8000h is the character byte for the upper left character on the display, and 0B8001h is the attribute for that character; address 0B8002h is the character byte for the next character to the right, and 0B8003h is its attribute.

### 3.4.1 Character Byte

The character byte can be any number from 00h to FFh to designate one of the 256 fonts stored in the font location in memory. For example, the ASCII code for the letter A is 41h, so to display an A in the upper left corner of the screen, 41h is written to 0B8000h. The character byte is used as an address by the video controller to point to the correct font in memory. For information, see “Fonts” on page 3-13.

	7							0
<b>Bit</b>	CA7	CA6	CA5	CA4	CA3	CA2	CA1	CA0

Bit	Name	R/W	Function
7-0	CA7-CA0	R/W	ASCII code for the character to be displayed. Used as an address (CA7-CA0) by the controller to fetch the font for display.

### 3.4.2 Attribute Byte

The attribute byte defines the way the character is displayed on the screen, defining the character’s color, intensity, blinking, and so on. The attribute byte is defined differently for each mode (CGA and HGA). CGA and HGA attribute-byte definitions are shown in the following paragraphs, with Table 3-11 on page 3-10 and Table 3-12 on page 3-10 showing the CGA-attribute-bit logic and Table 3-13 on page 3-11 showing the HGA-attribute-bit logic.

### 3.4.3 CGA Attribute Byte

	7							0
<b>Field</b>		Background Color				Foreground Color		
<b>Bit</b>	BLINK	BAK2	BAK1	BAK0	INTEN	FOR2	FOR1	FOR0

Bit	Name	R/W	Function
7	BLINK	R/W	Blinking if bit 5 of the CGA Mode Control register at Port 3D8h is 1. If bit 5 is 0, this is the background intensity bit (see Table 3-12 on page 3-10). 0 = Not blinking 1 = Blinking
6-4	BAK2-BAK0	R/W	Background color
3	INTEN	R/W	Intensity of character: 0 = Normal intensity 1 = High intensity
2-0	FOR2-FOR0	R/W	Foreground color

Table 3-11 CGA Foreground-Color Bit Values

FOR2	FOR1	FOR0	Color	
Red	Green	Blue	Intensity = 0	Intensity = 1
0	0	0	Black	Dark gray
0	0	1	Blue	Light blue
0	1	0	Green	Light green
0	1	1	Cyan	Light cyan
1	0	0	Red	Light red
1	0	1	Magenta	Light magenta
1	1	0	Brown	Yellow
1	1	1	Light gray	White

Table 3-12 CGA Background-Color Bit Values

BAK2	BAK1	BAK0	Color	
Red	Green	Blue	Bit 5 of Port 3D8h = 1 and Blink = 0 or Bit 5 of Port 3D8h = 0	Bit 5 of Port 3D8h = 1 and Blink = 1
0	0	0	Black	Dark gray
0	0	1	Blue	Light blue
0	1	0	Green	Light green
0	1	1	Cyan	Light cyan
1	0	0	Red	Light red
1	0	1	Magenta	Light magenta
1	1	0	Brown	Yellow
1	1	1	Light Gray	White



### 3.4.4 HGA Attribute Byte

	7					0			
<b>Field</b>		Background Definition				Foreground Definition			
<b>Bit</b>	BLINK	BAK2	BAK1	BAK0	INTEN	FOR2	FOR1	FOR0	

Bit	Name	R/W	Function
7	BLINK	R/W	Blinking if bit 5 of the HGA Mode Control register at Port 3B8h is 1: 0 = Not blinking 1 = Blinking
6-4	BAK2-BAK0	R/W	Background definition
3	INTEN	R/W	Intensity of character: 0 = Normal intensity 1 = High intensity
2-0	FOR2-FOR0	R/W	Foreground definition

Table 3-13 HGA Attribute-Byte Bit Values

BAK2	BAK1	BAK0	FOR2	FOR1	FOR0	Character Displayed As
0	0	0	0	0	0	Nondisplay
0	0	0	0	0	1	Underline
0	0	0	1	1	1	Normal display
1	1	1	0	0	0	Reverse video

### 3.4.5 SRAM Display Data Area Partitioning

The display-data area of SRAM is partitioned as shown in Table 3-14 on page 3-12.

Table 3-14 on page 3-12 shows CGA or HGA 80×25 text mode page 0. Notice that one page (or one screen of information) takes up less than 4 Kbyte of memory (80 columns · 25 rows · 2 bytes per character = 4000 bytes), and there are 16 Kbyte of memory available for display data. Therefore, memory holds up to 4 pages (or screens) of data at one time.

Table 3-14 Text-Mode Video-Memory Data: 640×200 Display, 80×25 Characters

Display row 0	0B8000h	C	A	C	A	...	C	A	0B809Fh
Display row 1	0B80A0h	C	A	C	A	...	C	A	0B813Fh
Display row 2	0B8140h	C	A	C	A	...	C	A	0B81DFh
.	.	.	.	.	.	...	.	.	.
.	.	.	.	.	.	...	.	.	.
.	.	.	.	.	.	...	.	.	.
Display row 24	0B8F00h	C	A	C	A	...	C	A	0B8F9Fh
0B8FA0h		(Unused)			0B8FFFh				

**Note:**

C = Character byte

A = Attribute byte

Selecting between the pages can be done using the registers at video indexes 0Ch and 0Dh (the 6845-compatible Start Address High and Start Address Low registers) and/or by using video indexes 1Ah and 1Bh (the Screen Adjust High and Screen Adjust Low registers).

The 4 pages in 80×25 text mode that are available in memory are located as follows:

- Page 0: 0B8000–0B8FFFh
- Page 1: 0B9000–0B9FFFh
- Page 2: 0BA000–0BAFFFh
- Page 3: 0BB000–0BBFFFh

The 40×25 text mode (CGA only) works the same way except you get 8 pages to work with (because each page is only 2 Kbyte of memory).

Table 3-15 Text-Mode Video-Memory Data: 640×200 Display, 40×25 Characters

Display row 0	0B8000h	C	A	C	A	...	C	A	0B804Fh
Display row 1	0B8050h	C	A	C	A	...	C	A	0B809Fh
Display row 2	0B80A0h	C	A	C	A	...	C	A	0B80EFh
.	.	.	.	.	.	...	.	.	.
.	.	.	.	.	.	...	.	.	.
.	.	.	.	.	.	...	.	.	.
Display row 24	0B8370h	C	A	C	A	...	C	A	0B877Fh
0B8780h		(Unused)						0B87FFh	

**Note:**

C = Character byte

A = Attribute byte

The 8 pages in 40×25 text mode that are available in memory are located as follows:

- Page 0: 0B8000–0B87FFh
- Page 1: 0B8800–0B8FFFh
- Page 2: 0B9000–0B97FFh
- Page 3: 0B9800–0B9FFFh
- Page 4: 0BA000–0BA7FFh
- Page 5: 0BA800–0BAFFFh
- Page 6: 0BB000–0BB7FFh
- Page 7: 0BB800–0BBFFFh

An odd-size LCD panel like the 480×320 panel works the same way as the standard 640×200 panel, but you must adjust the memory partition for it. Taking the 480×320 LCD panel for example, an 8×8 font yields 60×40 characters (480 pixels ÷ 8 pixels per character = 60 characters; 320 rows ÷ 8 rows per character = 40 characters). Therefore, memory partitions are as shown in Table 3-16 on page 3-13.

Table 3-16 Text-Mode Video-Memory Data: 480×320 LCD Panel, 60×40 Characters

Display row 0	0B8000h	C	A	C	A	...	C	A	0B8077h
Display row 1	0B8078h	C	A	C	A	...	C	A	0B80EFh
Display row 2	0B80F0h	C	A	C	A	...	C	A	0B8167h
.	.	.	.	.	.	...	.	.	.
.	.	.	.	.	.	...	.	.	.
.	.	.	.	.	.	...	.	.	.
Display row 39	0B9248h	C	A	C	A	...	C	A	0B92BFh
	0B92C0h	(Unused)							0B9FFFh

**Note:**

*The second page is at 0BA000–0BBFFFh.*

*C = Character byte*

*A = Attribute byte*

## 3.5 FONTS

In the ÉlanSC300 microcontroller, the character fonts are stored in memory as a bitmap representing the shape of the character. Since the character byte in display-data memory is a value between 00h and FFh, this provides 256 characters to choose from in each font area.

### 3.5.1 Font Areas

There are four different font areas in memory: two 8×8 CGA-font areas, one 8×16 CGA-font area, and one 9×14 HGA-font area. The video controller accesses only fonts from one font area at a time depending on the mode selected. The designer could store different fonts in different font areas of memory and use the registers to choose between them, but only one font area would be used at a time. If you switch between an 8×8 font and

any other size font, you must readjust your 6845 registers to correct the timing to the display. The Max Scan Line register at Video Index 09h is used by the video controller to determine the size of the font you have stored. All fonts are 8 bits wide (1 byte of memory per row).

The size of both character font areas are 2 Kbyte each (i.e., 256 fonts · 8 bytes each = 2 Kbyte, 1 row = 1 byte). The special character font is 8×16, so it takes up 4 Kbyte of memory (256 fonts · 16 bytes each = 4 Kbyte). The HGA 9×14 font is also allocated 4 Kbyte for its storage, but each row is still saved as 1 byte; the ninth column is automatically generated by the video controller.

### 3.5.2 Video Controller Font Fetches to SRAM

Table 3-17 on page 3-14 illustrates how the video controller determines which font area is selected. Table 3-18 on page 3-15 illustrates how the video controller reads a character font stored in SRAM, and therefore, how a character font must be stored in SRAM.

Table 3-17 Font Areas

Font Name	Font Size	Memory Size	ISA Address	Registers to Select
Character fonts 1	8×8	2 Kbyte	0BF800–0BFFFFh	Bit 0 of Video Index 18h = 0 (select CGA mode) Bit 3 of Video Index 18h = 0 (select character font 1) Bit 2 of Video Index 20h = 0 (select 8×8 font locations)
Character fonts 2	8×8	2 Kbyte	0BF000–0BF7FFh	Bit 0 of Video Index 18h = 0 (select CGA mode) Bit 3 of Video Index 18h = 1 (select character font 2) Bit 2 of Video Index 20h = 0 (select 8×8 font locations)
Special character font	8×16	4 Kbyte	0BD000–0BDFFFh	Bit 0 of Video Index 18h = 0 (select CGA mode) Bit 3 of Video Index 18h = x (Don't Care) Bit 2 of Video Index 20h = 1 (select special character font locations)
HGA character fonts	9×14	4 Kbyte	0BE000–0BEFFFh	Bit 0 of Video Index 18h = 1 (select HGA mode) Bit 3 of Video Index 18h = x (Don't Care) Bit 2 of Video Index 20h = x (Don't Care)

Table 3-18 SRAM Address Mapping during Video Controller Font Fetches

SRAM Address Signal	Code	Character Fonts 1	Character Fonts 2	Special Character Fonts	HGA Character Fonts
DSMA0	RA0	RA0	RA0	RA0	RA0
DSMA1	RA1	RA1	RA1	RA1	RA1
DSMA2	RA2	RA2	RA2	RA2	RA2
DSMA3	CA0	CA0	CA0	CA0	CA0
DSMA4	CA1	CA1	CA1	CA1	CA1
DSMA5	CA2	CA2	CA2	CA2	CA2
DSMA6	CA3	CA3	CA3	CA3	CA3
DSMA7	CA4	CA4	CA4	CA4	CA4
DSMA8	CA5	CA5	CA5	CA5	CA5
DSMA9	CA6	CA6	CA6	CA6	CA6
DSMA10	CA7	CA7	CA7	CA7	CA7
DSMA11	RA3	1	0	RA3	RA3
DSMA12	DISPMOD	1	1	1	0
DSMA13	SPCHRFBNT	1	1	0	1
DSMA14	1	1	1	1	1

**CA7–CA0** The character address is the character byte (ASCII code) stored in the display data part of memory. It is used by the video controller as an address to point to the position in memory where that character font resides.

**RA3–RA0** The row address is generated by the video controller and is used to point to the row of the font to be displayed. An 8×8 font only has 8 rows, so it only needs RA2–RA0 to point to all rows. The two 8×8 fonts (character fonts 1 and 2) use the inverse of bit 2 of the Screen Control Restore register at Video Index 18h in place of RA3 on DSMA11 to choose between the two font areas. For the larger fonts (8×16 and 9×14), RA3 is the fourth row address signal to point to rows greater than row 7.

**DISPMOD** The inverse of bit 0 of the Screen Control Restore register at Video Index 18h selects between the CGA and HGA font areas.

**SPCHRFBNT** The inverse of bit 2 of the Control 1 register at Video Index 20h selects between using the character font 1 and 2 locations or using the special character font location.

### 3.5.3 Storing Fonts in Video SRAM

When storing the fonts to memory, each row of the character is written as a byte. The location to write is determined by using the ASCII-character code for DSMA10–DSMA3, and the row being written for DSMA11 and DSMA2–DSMA0. For the 8×8 fonts, DSMA11 is not a row address; it is 1 or 0 depending on the font area in memory being written (1 chooses the character fonts 1 area, 0 chooses the character fonts 2 area).

The first eight rows of a font are written sequentially in memory, but the fonts that are larger than eight rows have their remaining rows stored in memory 2 Kbyte away from the beginning of that font area.

The following are examples of how to program an 8×8 character font and an 8×16 character font.

### 3.5.4 Font Example 1: The Letter A

ASCII code of 41h

8×8 font

Character font 1 area

The bitmap for the letter A in an 8×8 font may look like this:

Row	Bit								Hexadecimal Equivalent
	7	6	5	4	3	2	1	0	
0	0	0	1	1	0	0	0	0	30h
1	0	1	1	1	1	0	0	0	78h
2	1	1	0	0	1	1	0	0	CCh
3	1	1	0	0	1	1	0	0	CCh
4	1	1	1	1	1	1	0	0	FCh
5	1	1	0	0	1	1	0	0	CCh
6	1	1	0	0	1	1	0	0	CCh
7	0	0	0	0	0	0	0	0	00h

Each bit represents a pixel turned on (1) or off (0) on the display to produce the letter A. Each of the rows is 8 bits and is saved in the font location of memory as a byte of data, starting with row 0 and moving sequentially through row 7.

The ISA-bus addresses that the CPU needs to generate in order to store the font for A in font area 1 can be determined using Table 3-17 on page 3-14 and Table 3-18 on page 3-15. The following chart shows how the address range 0BFA08–0BFA0Fh was determined. The letter x in bits 2–0 corresponds to RA2–RA0 from Table 3-18 on page 3-15.

Address	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit	0	0	0	0	1	0	1	1	1	1	1	1	1	0	1	0	0	0	0	0	1	x	x	x
	Video memory access 0B8x								1	Character font 1				ASCII code of 41h										

So the CPU stores the font for the letter A in memory as shown in Table 3-19 on page 3-17.

Table 3-19 Storing the Font for the Letter A

To Save Row	Write	To Location
0	30h	0BFA08h
1	78h	0BFA09h
2	CCh	0BFA0Ah
3	CCh	0BFA0Bh
4	FCh	0BFA0Ch
5	CCh	0BFA0Dh
6	CCh	0BFA0Eh
7	00h	0BFA0Fh

Program the controller to select the correct font area when displaying as shown in Table 3-20 on page 3-17.

Table 3-20 Selecting the Font Area for the Letter A

Write	To Port	Comment
18h	3D4h	
00h	3D5h	Bit 0 to select CGA mode, and bit 3 to select the character font 1 area
20h	3D4h	
00h	3D5h	Bit 2 to select 8×16 fonts

Putting 41h in display-data memory at an even byte causes the letter A to appear on the display.

### 3.5.5 Font Example 2: The Letter M

ASCII code of 4Dh

8×16 font

Special character font area

The bitmap for the letter M in an 8×16 font may look like this:

Row	Bit								Hexadecimal Equivalent
	7	6	5	4	3	2	1	0	
0	0	0	0	0	0	0	0	0	00h
1	1	0	0	0	0	0	1	0	82h
2	1	1	0	0	0	1	1	0	C6h
3	1	1	1	0	1	1	1	0	Eeh
4	1	1	1	0	1	1	1	0	Eeh
5	1	1	1	1	1	1	1	0	Feh
6	1	1	1	1	1	1	1	0	Feh
7	1	1	1	1	1	1	1	0	Feh
8	1	1	1	1	1	1	1	0	Feh
9	1	1	0	1	0	1	1	0	D6h
10	1	1	0	1	0	1	1	0	D6h
11	1	1	0	0	0	1	1	0	CCh
12	1	1	0	0	0	1	1	0	CCh
13	1	1	0	0	0	1	1	0	CCh
14	1	1	0	0	0	1	1	0	CCh
15	0	0	0	0	0	0	0	0	00h

Again, the ISA-bus address ranges 0BD268–0BD26Fh and 0BDA68–0BDA6Fh are determined using information from Table 3-17 on page 3-14 and Table 3-18 on page 3-15. Address bit 11 is derived from RA3. Address bits 2–0 are derived from RA2–RA0.

Address	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit	0	0	0	0	1	0	1	1	1	1	0	1	x	0	1	0	0	1	1	0	1	x	x	x
	Video memory access 0B8x								1	Special character font	ASCII code of 4Dh													

The CPU stores the font for the letter M in memory as shown in Table 3-21 on page 3-19.



Table 3-21 Storing the Font for the Letter M

To Save Row	Write	To Location
0	00h	0BD268h
1	82h	0BD269h
2	C6h	0BD26Ah
3	Eeh	0BD26Bh
4	Eeh	0BD26Ch
5	FEh	0BD26Dh
6	FEh	0BD26Eh
7	FEh	0BD26Fh
(2-Kbyte address space)		
8	FEh	0BDA68h
9	D6h	0BDA69h
10	D6h	0BDA6Ah
11	CCh	0BDA6Bh
12	CCh	0BDA6Ch
13	CCh	0BDA6Dh
14	CCh	0BDA6Eh
15	00h	0BDA6Fh

Program the controller to select the correct font area when displaying as shown in Table 3-22 on page 3-19.

Table 3-22 Selecting the Font Area for the Letter M

Write	To Port	Comment
18h	3D4h	
00h	3D5h	Bit 0 to select CGA mode
20h	3D4h	
04h	3D5h	Bit 2 to select special character font area

Now, putting 4Dh in display-data memory at an even byte causes the letter M to appear on the display.

## 3.6 VIDEO REGISTER OVERVIEW

The video-controller register set is mapped into the CPU's I/O address space in the 3D0–3DFh and 3B0–3BFh blocks for CGA and HGA, respectively. CGA and HGA are mutually exclusive configurations; CGA is the default.

**Note:** All registers are fully CGA and HGA compatible, with the following exceptions:

1. There is no support for light-pen functionality.
2. The extended registers at video indexes 12–25h are nonstandard and represent a superset of CGA/HGA.

### 3.6.1 Port Registers

The registers shown in Table 3-23 on page 3-20 are directly accessible by the CPU.

Table 3-23 Video Port Definitions

Port	R/W	Description
3B4h	W	HGA Index Address
3B5h	R/W	HGA Index Data
3B8h	W	HGA Mode Control
3BAh	R	HGA Status
3BFh	W	HGA Configuration
3D4h	W	CGA Index Address
3D5h	R/W	CGA Index Data
3D8h	R/W	CGA Mode Control
3D9h	R/W	CGA Color Select
3DAh	R	CGA Status

### 3.6.2 Index Registers

The video index registers in Table 3-24 on page 3-21 are available in both the CGA and HGA modes, and are accessible indirectly via the Video Index Address and Video Index Data registers (3D4h and 3D5h, or 3B4h and 3B5h) shown in Table 3-23 on page 3-20.

Table 3-24 6845 and Extended Registers

6845 Video Index Registers		Extended Video Index Registers	
Video Index	Description	Video Index	Description
00h	Horizontal Total	12h	Enable Software Switch
01h	Horizontal Displayed	13h	Disable Software Switch
02h	Horizontal Sync Position	14h	Color Mapping
03h	Horizontal Sync Width	15h	Color Mapping
04h	Vertical Total	16h	Color Mapping
05h	Vertical Total Adjust	17h	Color Mapping
06h	Vertical Displayed	18h	Screen Control Restore
07h	Vertical Sync Position	19h	Screen Control 2
08h	Interlace Mode	1Ah	Screen Adjust Lower Byte
09h	Maximum Scan Line	1Bh	Screen Adjust Upper Byte
0Ah	Cursor Start	1Ch	Color Mapping
0Bh	Cursor End	1Dh	Color Mapping
0Ch	Start Address Upper Byte	1Eh	Color Mapping
0Dh	Start Address Lower Byte	1Fh	Color Mapping
0Eh	Cursor Address Upper Byte	20h	Control 1
0Fh	Cursor Address Lower Byte	21h	Text Truncation Start
10h	(Reserved)	22h	Text Truncation Stop
11h	(Reserved)	23h	Graphics Truncation Start
		24h	Graphics Truncation Stop
		25h	LCD Special

**Note:**

Video index registers 00–0Fh are standard 6845-controller registers.

## 3.7 VIDEO PORT REGISTERS

### 3.7.1 HGA Index Address Register (Port 3B4h)

This register is a pointer that provides indirect access to the video index registers. It is to be written with the address of the desired video index register. This video index address register provides HGA compatibility and is the same register accessed at Port 3D4h for CGA.

7

0

Field	Address of Video Index Register							
Reset	0	0	0	0	0	0	0	0

### 3.7.2 HGA Index Data Register (Port 3B5h)

This register contains the data value to be written to, or read from, the video index register currently pointed to by the HGA Index Address register at Port 3B4h. This video index data register provides HGA compatibility and is the same register accessed at Port 3D5h for CGA.

	7							0
<b>Field</b>	Data for/from Video Index Register							
<b>Reset</b>	0	0	0	0	0	0	0	0

### 3.7.3 HGA Mode Control Register (Port 3B8h)

This register configures the operating mode of the HGA display system.

	7						0				
<b>Bit</b>	(Reserved)		TXTBLNK	(Reserved)		VIDCON	(Reserved)		TXTGRP	(Reserved)	
<b>Reset</b>	0	0	0	0	0	0	0	0	0	0	0

Bit	Name	R/W	Function
7–6		W	(Reserved—must be 0)
5	TXTBLNK	W	Text attribute control: 0 = Blinking attribute disabled 1 = Blinking attribute enabled
4		W	(Reserved—must be 0)
3	VIDCON	W	Video control: 0 = Video disabled (screen blank) 1 = Video enabled
2		W	(Reserved—must be 0)
1	TXTGRP	W	HGA text/graphics control: 0 = 80×25 text mode 1 = 720×348 graphics mode
0		W	(Reserved—must be 0)

### 3.7.4 HGA Status Register (Port 3BAh)

This register provides display-timing status to the CPU.

	7					0		
<b>Bit</b>	(Reserved)				VERTRET	(Reserved)		DMSTAT
<b>Reset</b>	0	0	0	0	0	0	0	0

Bit	Name	R/W	Function
7–4		R	(Reserved)
3	VERTRET	R	Vertical retrace status (enabled by bit 0 of Index 25h): 0 = Raster is not in vertical retrace 1 = Raster is in vertical retrace
2–1		R	(Reserved)
0	DMSTAT	R	Display-memory access status: 0 = Display-memory access by the CPU will cause interference 1 = Display-memory access by the CPU will not cause interference

DMSTAT (bit 0) is intended to stop conflicts between the video controller and the CPU when accessing video memory. The original CGA had a problem with these conflicts resulting in interference (also commonly called snow or sparkles) that could be seen on the display. This bit is here for compatibility only; it is not necessary to check for access before writing to the video memory with this controller.

### 3.7.5 HGA Configuration Register (Port 3BFh)

This register provides master control of the HGA-graphics mode, in that an enable or disable in this register overrides a selection in the HGA Mode Control register at Port 3B8h.

	7							0
<b>Bit</b>	(Reserved)							MASTG
<b>Reset</b>	0	0	0	0	0	0	0	0

Bit	Name	R/W	Function
7-1		W	(Reserved—must be 0)
0	MASTG	W	Master text/graphics control: 0 = Prevents graphics mode 1 = Allows graphics mode

### 3.7.6 CGA Index Address Register (Port 3D4h)

This register is a pointer that provides indirect access to the video index registers. It is to be written with the address of the desired video index register. This video index address register provides CGA compatibility and is the same register accessed at Port 3B4h for HGA.

	7							0
<b>Field</b>	Address of Video Index Register							
<b>Reset</b>	0	0	0	0	0	0	0	0

### 3.7.7 CGA Index Data Register (Port 3D5h)

This register contains the data value to be written to, or read from, the video index register currently pointed to by the CGA Index Address register at Port 3D4h. This video index data register provides CGA compatibility and is the same register accessed at Port 3B5h for HGA.

	7							0
<b>Field</b>	Data for/from Video Index Register							
<b>Reset</b>	0	0	0	0	0	0	0	0

### 3.7.8 CGA Mode Control Register (Port 3D8h)

This register configures the operating mode of the CGA display system.

	7						0	
<b>Bit</b>	(Reserved)		TXTBLNK	GRPCON	VIDCON	COLBUR	TGCON	TXTCON
<b>Reset</b>	0	0	0	0	0	0	0	0

Bit	Name	R/W	Function
7–6		R/W	(Reserved—must be 0)
5	TXTBLNK	R/W	Text attribute control: 0 = Blinking attribute disabled (bit 7 of the attribute byte controls background intensity) 1 = Blinking attribute enabled (bit 7 of the attribute byte controls blinking)
4	GRPCON	R/W	CGA graphics control: 0 = 320-column graphics mode 1 = 640-column graphics mode  For nonstandard CGA size, LCD panels interpret this register in the following way: 0 = 2-bpp graphics 1 = 1-bpp graphics
3	VIDCON	R/W	Video control: 0 = Video disable (blank screen) 1 = Video enable
2	COLBUR	R/W	CGA color burst select (affects colors/gray scales to the display in 320-column graphics mode): 0 = Color burst enabled 1 = Color burst disabled
1	TGCON	R/W	CGA text/graphics control: 0 = Text mode 1 = Graphics mode
0	TXTCON	R/W	CGA 80/40-column text select: 0 = 40×25 text mode 1 = 80×25 text mode  For nonstandard CGA size, LCD panels interpret this register in the following way: 0 = Double-width characters (16 pixels wide instead of 8) 1 = Normal-width characters (8 pixels wide)

### 3.7.9 CGA Color Select Register (Port 3D9h)

This register selects the background color for text and 2-bpp graphics modes. For 1-bpp graphics modes, this register defines the color (bits 4–0) for the pixel when on.

	7							0
<b>Bit</b>	(Reserved)		ALTPAL	ALTBK	SBINT	SBRED	SBGREEN	SBBLUE
<b>Reset</b>	0	0	0	0	0	0	0	0

Bit	Name	R/W	Function
7–6		R/W	(Reserved—must be 0)
5	ALTPAL	R/W	This bit selects alternate color sets (palettes) for the 2-bpp graphics-mode display: 0 = Green, red, and yellow palette 1 = Cyan, magenta, and white palette
4	ALTBK	R/W	When set, this bit selects an alternate, intensified color for the 2-bpp graphic-mode display: 0 = Intensity off for colors 1 = Intensity on for colors
3	SBINT	R/W	Screen background color: intensity bit
2	SBRED	R/W	Screen background color: red bit
1	SBGREEN	R/W	Screen background color: green bit
0	SBBLUE	R/W	Screen background color: blue bit

Table 3-25 Screen-Background Color-Bit Logic

SBRED	SBGREEN	SBBLUE	Color	
Red	Green	Blue	SBINT = 0	SBINT = 1
0	0	0	Black	Dark gray
0	0	1	Blue	Light blue
0	1	0	Green	Light green
0	1	1	Cyan	Light cyan
1	0	0	Red	Light red
1	0	1	Magenta	Light magenta
1	1	0	Brown	Yellow
1	1	1	Light gray	White

**Note:**

In 1-bpp graphics mode, bits 3–0 control the foreground gray scale.

### 3.7.10 CGA Status Register (Port 3DAh)

This register provides display-timing status to the CPU.

	7					0		
<b>Bit</b>	(Reserved)				VERTRET	(Reserved)		DMSTAT
<b>Reset</b>	0	0	0	0	0	0	0	0

Bit	Name	R/W	Function
7–4		R	(Reserved)
3	VERTRET	R	Vertical retrace status (enabled by bit 0 of the LCD Special register at Index 25h): 0 = Raster is not in vertical retrace 1 = Raster is in vertical retrace
2–1		R	(Reserved)
0	DMSTAT	R	Display-memory access status 0 = Display-memory access by the CPU causes interference 1 = Display-memory access by the CPU does not cause interference

DMSTAT (bit 0) is intended to allow software to avoid conflicts between the video controller and CPU when accessing video memory. The original CGA had a problem with these conflicts resulting in interference (also commonly called snow or sparkles) that could be seen on the display. This bit is here for compatibility only; it is not necessary to check for access before writing to the video memory with this controller.

## 3.8 STANDARD VIDEO INDEX REGISTERS

The standard 6845 registers have a slightly nonstandard meaning when used to support an LCD panel. The differences between the standard 6845 implementation and the ÉlanSC300 microcontroller's nonstandard implementation are detailed below.

### 3.8.1 Horizontal Registers (Video Indexes 00–03h)

To set up the LCD-panel timing, the Horizontal Total register at Video Index 00h and the Horizontal Displayed register at Video Index 01h must be programmed. Both of these registers should be programmed to the same value because there are no retrace requirements for an LCD panel. The Horizontal Sync Position register at Video Index 02h and the Horizontal Sync Width register at Video Index 03h should be programmed to 00h. They are not used in LCD mode.

In Text mode, the Horizontal Displayed and Horizontal Total registers are programmed with the number of characters per row on the display as follows:

- 640×200 LCD panel
  - 80 for 80×25 mode
  - 40 for 40×25 mode
- 480×320 LCD panel
  - 60 for 60×40 mode
  - 30 for 30×40 mode



In graphics mode, the Horizontal Displayed register is programmed with the number of pixels across the LCD panel divided by 16: 40 for a 640×200 LCD panel, or 80 for a 480×320 LCD panel.

### 3.8.2 The Vertical Registers (Video Indexes 04–07h)

To set up the LCD-panel timing, the Vertical Total register at Video Index 04h, the Vertical Displayed register at Video Index 06h, and the Vertical Sync Position register at Video Index 07h must be programmed. All these registers should be programmed to the same value because there are no retrace requirements for an LCD panel. The Vertical Total Adjust register at Video Index 05h should be programmed to 00h because it is not used in LCD mode.

In Text mode, the Vertical Displayed, Vertical Total, and Vertical Sync Position registers are programmed with the number of rows of characters on the display: 25 for a 640×200 LCD panel in both 80×25 and 40×25 text modes, or 40 for a 480×320 LCD panel in 60×40 and 30×40 text modes.

In Graphics mode, the Vertical Displayed, Vertical Total, and Vertical Sync Position registers represent the number of pixel rows on the display divided by the number of banks into which memory is partitioned: 200 for a 640×200 LCD panel with 2 banks of memory, or 80 for a 480×320 LCD panel with 4 banks of memory.

### 3.8.3 Interlace Mode Register (Video Index 08h)

This register is not used in LCD mode.

### 3.8.4 Max Scan Line Register (Video Index 09h)

This register works the same as the standard 6845 implementation.

In text mode, this register determines the number of scan lines per character row. It is programmed with one less than the number of scan lines. For example, using an 8×8 font, write a value of 07h; for an 8×16 font, write 0Fh.

In graphics mode, this register is programmed for the number of banks into which memory is partitioned. It is programmed for one less than the number of banks (e.g., 01h for 2 banks, or 03h for 4 banks).

### 3.8.5 Cursor Start and End Registers (Video Indexes 0A–0Bh)

These registers work the same as the standard 6845 implementation. The Cursor Start and End registers define which rows in the character are used for the cursor.

### 3.8.6 Start Address Registers (Video Indexes 0C–0Dh)

These registers work the same as the standard 6845 implementation. The Start Address registers contain the address in memory at which to begin the display. For information, see “Screen Adjust Lower Byte Register (Video Index 1Ah)” on page 3-32.

### 3.8.7 Cursor Address Registers (Video Indexes 0E–0Fh)

These registers work the same as the standard 6845 implementation. The Cursor Address registers contain the address in memory at which to position the cursor.

### 3.8.8 Reserved Registers (Video Indexes 10–11h)

These registers were the Light Pen registers in the 6845, but are not supported in any mode here. They are not used in the ÉlanSC300 microcontroller.

## 3.9 EXTENDED VIDEO INDEX REGISTERS

### 3.9.1 Enable Software Switch Register (Video Index 12h)

The video controller must be enabled before software can access the extended video index registers at indexes 14–25h. This is accomplished by writing 12h to the CGA Index Address register at Port 3D4h, immediately followed by reading the CGA Index Data register at Port 3D5h (see Table 3-26 on page 3-28). These registers change to ports 3B4h and 3B5h if the controller is changed to HGA mode via bit 0 of the Screen Control Restore register at Video Index 18h. The power-on default is the CGA mode, so use ports 3D4h and 3D5h when enabling for the first time.

Table 3-26 Enabling the Extended Video-Controller Registers

Perform this Access	To or From this Port	With this Data
Write	3D4h	12h (address value)
Read	3D5h	

### 3.9.2 Disable Software Switch Register (Video Index 13h)

After programming the extended video index registers, they can be disabled to stop accidental changes by software. This is accomplished by writing 13h to the CGA Index Address register at Port 3D4h, immediately followed by reading the CGA Index Data register at Port 3D5h. These registers change to ports 3B4h and 3B5h if the controller is changed to HGA mode via bit 0 of the Screen Control Restore register at Video Index 18h).

Table 3-27 Disabling the Extended Video-Controller Registers

Perform this Access	To or From this Port	With this Data
Write	3D4h	13h (address value)
Read	3D5h	

### 3.9.3 Color Mapping Registers (Video Indexes 14-17h & 1C-1Fh)

The LCD controller supports 16 levels of gray in CGA text mode. However, the effect of the gray scale depends heavily on the quality of the display, on backlighting conditions, and on the application software. To improve viewing perception, a *color-mapping* function is provided to allow software to adjust the mapping of a color to the gray level displayed by the LCD controller. Color mapping is controlled by the Color Mapping registers. This function works in both text and graphics modes. It is enabled by bit 4 of the Screen Control 2 register at Video Index 19h.

	7									0
<b>Field</b>	High Nibble				Low Nibble					
<b>Reset</b>	0	0	0	0	0	0	0	0	0	

The eight Color Mapping registers are located at indexes 14–17h and 1C–1Fh. Each register contains two 4-bit nibbles, for a total of 16 nibbles. Each nibble controls the mapping of one of the 16 CGA gray levels. Each gray level is defined by the 4-bit quantity: I, R, G, B. The mapping is shown in Table 3-28 on page 3-29.

Table 3-28 Gray-Scale Mapping Using the Color-Mapping Registers

I	R	G	B	Video Index Register	Nibble
0	0	0	0	14h	Lower
0	0	0	1	14h	Upper
0	0	1	0	15h	Lower
0	0	1	1	15h	Upper
0	1	0	0	16h	Lower
0	1	0	1	16h	Upper
0	1	1	0	17h	Lower
0	1	1	1	17h	Upper
1	0	0	0	1Ch	Lower
1	0	0	1	1Ch	Upper
1	0	1	0	1Dh	Lower
1	0	1	1	1Dh	Upper
1	1	0	0	1Eh	Lower
1	1	0	1	1Eh	Upper
1	1	1	0	1Fh	Lower
1	1	1	1	1Fh	Upper

The sequence in Table 3-29 on page 3-30 displays the CGA color with IRGB = 0000b as gray scale 1111b and the CGA color with IRGB = 0001b as gray scale 1110b.

Table 3-29 Mapping CGA Colors to Gray Scales

Perform this Access	To or From this Port	With this Data
Write	3D4h	12h
Read	3D5h	
Write	3D4h	14h
Write	3D5h	EFh

### 3.9.4 Screen Control Restore Register (Video Index 18h)

This register is used to set up the mode, select the display type, and select the font area.

Field	LCD Panel		(Reserved)	DSPTYPE	CGATXTF	(Reserved)	AUTBLNK	DSPMOD
Bit	LCDPAN1	LCDPAN0						
Default	0	0	0	0	0	0	0	0

Bit	Name	R/W	Function
7–6	LCDPAN1–LCDPAN0	W	LCD panel select If the LCD panel is not one of these standard resolutions, choose a value based on the LCD panel's data bus size (see Table 3-30 on page 3-31). If you are using a dual-screen LCD panel, see the description of bit 1 in “Function Enable 2 Register (Index B1h)” on page 5-78.
5		W	(Reserved—must be 0)
4	DSPTYPE	W	Select display type: 0 = CRT display (not supported) 1 = LCD display (must be set)
3	CGATXTF	W	Select CGA text mode font (only valid when bit 0 of this register is 0 and bit 2 of the Control 1 register at Video Index 20 is 0): 0 = Select character fonts 1 area (default) 1 = Select character fonts 2 area
2	COLEM	W	(Reserved—must be 0)
1	AUTBLNK	W	Set up auto screen blanking: 0 = Normal display (default) 1 = Enable blanking (be sure to turn off display before activating) For duration setup, see bits 3–2 of the Screen Control 2 register at Video Index 19h.
0	DSPMOD	W	Display mode: 0 = CGA mode (default) 1 = HGA mode

Table 3-30 LCD-Panel Select Logic

Bit 7	Bit 6	LCD Panel
0	0	640×200 dual screen (default) – 8 bits of data to LCD: four upper panel and four lower panel – CGA mode only
0	1	640×200 and smaller single screen – 4 bits of data to LCD – CGA and HGA modes
1	0	(Reserved)
1	1	(Reserved)

### 3.9.5 Screen Control 2 Register (Video Index 19h)

This register is used to control several LCD modes.

Field	7				Screen Blank Timer		0	
Bit	(Reserved)		DSPCON	CLRMPEN	BLNKTIM1	BLNKTIM0	GRAYSC	POWDWN
Reset	0	0	0	0	0	0	0	0

Bit	Name	R/W	Function
7–6		W	(Reserved—must be 0)
5	DSPCON	W	Display-controller enable: 0 = Enable 1 = Disable  When the display controller is disabled, the system cannot access the video ports or the video index registers. The ÉlanSC300 microcontroller must be reset to enable the display controller.
4	CLRMPEN		Color-map enable (enables the Color Mapping registers at video indexes 14–17h and 1C–1Fh to adjust the gray scales to the LCD): 0 = Disable (normal display) 1 = Enable
3–2	BLNKTIM1–BLNKTIM0	W	Auto screen blanking timer setting (see Table 3-31 on page 3-32). To enable, see bit 1 of the Screen Control Restore register at Video Index 18h.
1	GRAYSC	W	Gray-scaling adjustment (CGA 320×200 text mode only): 0 = Disable 1 = Enable  The controller uses a predefined gray-scale mapping that looks best for this mode.
0	POWDWN	W	Power-down mode (stops the control signals to the LCD panel): 0 = Disable 1 = Enable

Table 3-31 Auto Screen-Blanking Timer Setting

Bit 3	Bit 2	CGA (s)	HGA (s)
0	0	135	90
0	1	70	45
1	0	35	20
1	1	135	90

### 3.9.5.1 Auto Screen Blanking

The Auto Screen Blanking (ASB) feature uses the clocks to the LCD to perform the time-out, so the above numbers are only an approximation. The actual values are different depending on the panel resolution and the mode in which it is operated. The ASB timer count down is restarted and the controller leaves ASB mode when a read from, or a write to, the video memory that is active occurs (0B8000–0BFFFFh in CGA mode, or 0B0000–0BFFFFh in HGA mode). When ASB is activated, the clocks to the LCD panel continue to toggle, but the data lines go Low. ASB does not deselect the SRAM either. Therefore, this mode is not used for power saving, but to preserve the screen by preventing a burned-in image. Bit 1 of the Screen Control Restore register at Video Index 18h must be 1 for auto screen blanking to be activated.

### 3.9.5.2 Display Controller Enable

Bit 5 can be used to disable the internal video controller when the system is using an external video controller instead. This bit must be used carefully. Once set, it cannot be cleared until the next power-on reset ( $\overline{\text{RESIN}}$ ) occurs.

Bit 5 stops accesses to the Enable Software Switch register at Video Index 12h so the extended registers cannot be accessed and this bit cannot be cleared. This is especially dangerous because bits 1 and 2 of the Control 1 register at Video Index 20h control the parallel-port address and are not accessible after the display controller is disabled. Instead of disabling the controller with bit 5, consider using bit 0 of the Screen Control Restore register at Video Index 18h to move the registers to HGA mode (3Bxh) so they do not conflict with the external video controller.

### 3.9.6 Screen Adjust Lower Byte Register (Video Index 1Ah)

	7							0
Field	Screen Adjust Lower Byte							
Reset	0	0	0	0	0	0	0	0

### 3.9.7 Screen Adjust Upper Byte Register (Video Index 1Bh)

	7							0
Field	Screen Adjust Upper Byte							
Reset	0	0	0	0	0	0	0	0

These two registers control the first address that the controller sends to the display. The Screen Adjust registers determine which part of the display-data memory is displayed.

These registers work the same as the 6845's Start Address Upper and Lower Byte registers (video indexes 0Ch and 0Dh). The LCD panel scrolls by the sum of the numbers in the Start Address and Screen Adjust registers. These registers only work for a single-screen LCD panel (bit 4 of the Screen Control Restore register at Video Index 18h is 1; bits 7 and 6 are 01b or 11b).

Normally, the video controller starts accessing memory at 0B8000h for the display data. In text mode, the byte at 0B8000h is the upper-left character on the display; in 640-column graphics mode, bit 7 is the upper-left character; and in 320-column graphics mode, bits 6 and 7 are the upper-left pixel. These registers are programmed to cause the controller to start accessing at a different address.

For example, if you write 00h to the Screen Adjust Upper Byte register at Video Index 1Bh and 03h to the Screen Adjust Lower Byte register at Video Index 1Ah, then in text mode, the character byte at address 0B8006h appears in the upper-left position on the display. Each character is defined by 2 bytes—the character and attribute bytes—so 0003h is multiplied by 2 and added to 0B8000h. The number programmed here is the number of the character at which to start displaying.

As a further example, to change to the second page in memory in text mode, write 07h to the Screen Adjust Upper Byte register and D0h to the Screen Adjust Lower Byte register: 07D0h is 2000d, and there are 2000 characters on the display in 80×25 text mode.

This function can be used to quickly change the display by pointing to another page in display-data memory, or to support fast horizontal scrolling by character, row, or page.

The page boundaries for page changes in 80×25 text mode are shown in Table 3-32 on page 3-33.

Table 3-32 Page Boundaries in 80×25 Text Mode

Page	Boundary
0	0000h
1	07D0h
2	0FA0h
3	1770h

These registers also work for graphics mode, but instead of shifting the display by a number of characters, they shift the display by a number of pixels. In both 640- and 320-column graphics modes, the number in the Screen Adjust registers is multiplied by 2 to determine the new starting address. This means that to scroll by two rows in graphics mode on a 640×200 LCD panel (shifting by one row of pixels is not possible because memory is split into two banks), a value of 00h must be written to the Screen Adjust Upper Byte register at Video Index 1Bh, and a value of 28h must be written to the Screen Adjust Lower Byte register at Video Index 1Ah ( $28h \cdot 2 = 50h = 80$  bytes).

A 1-bpp, 640×200 LCD panel uses 80 bytes (640 pixels per row ÷ 8 pixels per byte) of display-data area for one row. A 2-bpp, 320×200 LCD panel also uses 80 bytes (320 pixels per row ÷ 4 pixels per byte) of display-data area for one row. In both cases,

the pixel data for row 0 of the display starts at address 0B8050h in bank 1, and the pixel data for row 1 of the display starts at 0BA050h in bank 2.

### 3.9.8 Control 1 Register (Video Index 20h)

This register controls several miscellaneous functions.

	7						0	
<b>Field</b>							Printer I/O Port	
<b>Bit</b>	(Reserved)		DOTD	TRUNC	PROTDM	SPCHRFN	PRNPRT1	PRNPRT0
<b>Reset</b>	0	0	0	0	0	0	0	1

Bit	Name	R/W	Function
7–6		W	(Reserved—must be 0)
5	DOTD	W	Disable dot doubling (320 graphics mode only): 0 = Enable 1 = Disable  When the display is operating in 320×200 CGA mode, set this bit to display “1 pixel per pixel” versus “2 pixels per pixel.”
4	TRUNC	W	Truncation enable (enables the use of the Truncation registers at video indexes 21–24h for LCD mode): 0 = Disable 1 = Enable
3	PROTDM	W	Protect display-memory sector 0BC000–0BFFFFh in CGA mode (can be used for BIOS to protect the font and free areas from accidental overwrites by software): 0 = Allow CPU read/write access to sector (default) 1 = Deny CPU read/write access to sector
2	SPCHRFN	W	CGA special character font area enable: 0 = Disable (bit 3 of the Screen Control Restore register at Video Index 18 chooses font area in CGA mode) 1 = Enable
1–0	PRNPRT1–PRNPRT0	W	Printer I/O port select (see Table 3-33 on page 3-34).

Table 3-33 Printer I/O Port Select Logic

Bit 1	Bit 0	Printer I/O Port
0	0	(Disabled)
0	1	(default) 3Bxh
1	0	37xh
1	1	27xh



### 3.9.9 Text Truncation Start Register (Video Index 21h)

The Text Truncation registers at video indexes 21–22h are enabled via the following settings:

- Control 1 register at Video Index 20h, bit 4 = 1
- Screen Control Restore register at Video Index 18h, bit 4 = 1
- CGA Control register at Port 3D8h, bit 1 = 0

	7							0
<b>Field</b>	Number of First Truncated Character							
<b>Reset</b>	0	0	0	0	0	0	0	0

### 3.9.10 Text Truncation Stop Register (Video Index 22h)

	7							0
<b>Field</b>	Number of Last Truncated Character							
<b>Reset</b>	0	0	0	0	0	0	0	0

In LCD mode only, these two registers are used together to horizontally truncate the information sent to the LCD panel. They save some power by stopping the shift clock to the LCD panel and deselecting the SRAM during truncation. The Text Truncation Start register at Video Index 21h should be programmed with the character column number of the last column of characters to be displayed, plus one. The Text Truncation Stop register at Video Index 22h should be programmed with the character column number of the next character to display.

If Video Index 21h is greater than Video Index 22h, the following two areas are truncated:

- Text from the column specified in Video Index 22h to the last column
- Text from the first column to the column that precedes the column specified in Video Index 22

For example, if an LCD panel configured for 80 columns×25 rows is filled with numbers as follows:

<b>0</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>	<b>9</b>	<b>10</b>	<b>11</b>	...	<b>74</b>	<b>75</b>	<b>76</b>	<b>77</b>	<b>78</b>	<b>79</b>	<b>80</b>
0	1	2	3	4	5	6	7	8	9	0	1		3	4	5	6	7	8	9
0	1	2	3	4	5	6	7	8	9	0	1		3	4	5	6	7	8	9
0	1	2	3	4	5	6	7	8	9	0	1		3	4	5	6	7	8	9
0	1	2	3	4	5	6	7	8	9	0	1		3	4	5	6	7	8	9
0	1	2	3	4	5	6	7	8	9	0	1		3	4	5	6	7	8	9
												·							
												·							
												·							
0	1	2	3	4	5	6	7	8	9	0	1		3	4	5	6	7	8	9
0	1	2	3	4	5	6	7	8	9	0	1		3	4	5	6	7	8	9

And the Text Truncation registers are programmed as shown in Table 3-34 on page 3-36.

Table 3-34 Truncation-Register Programming Example

Write to this Port	With this Data	Comment
3D4h	21h	Text Truncation Start register
3D5h	0Fh	Truncate columns 14–80
3D4h	22h	Text Truncation Stop register
3D5h	04h	Truncate columns 0–3

Then the resulting display looks like this:

0	1	2	3	4	5	6	7	8	9	10	11	...	74	75	76	77	78	79	80	
4	5	6	7	8	9	0	1	2	3											
4	5	6	7	8	9	0	1	2	3											
4	5	6	7	8	9	0	1	2	3											
4	5	6	7	8	9	0	1	2	3											
4	5	6	7	8	9	0	1	2	3											
4	5	6	7	8	9	0	1	2	3											
												.								
												.								
												.								
4	5	6	7	8	9	0	1	2	3											
4	5	6	7	8	9	0	1	2	3											

The first four columns (0–3) of characters are eliminated, and the display is shifted to the left edge. Everything to the right of the 3 (now in column 9, formerly in column 13) is blank. If the Text Truncation Stop register was 00h, the display would start the count at 0 and extend to column 13.

### 3.9.11 Graphics Truncation Start Register (Video Index 23h)

The Graphics Truncation registers at video indexes 23–24h are enabled via the following settings:

- Control 1 register at Video Index 20h, bit 4 = 1
- Screen Control Restore register at Video Index 18h, bit 4 = 1
- CGA Control register at Port 3D8h, bit 1 = 1

	7							0
<b>Field</b>	Number of First Truncated Pixel							
<b>Reset</b>	0	0	0	0	0	0	0	0

### 3.9.12 Graphics Truncation Stop Register (Video Index 24h)

	7							0
<b>Field</b>	Number of Last Truncated Pixel							
<b>Reset</b>	0	0	0	0	0	0	0	0

The Graphics Truncation registers work the same as the Text Truncation registers except that the Graphics Truncation registers define a pixel instead of a character. The number programmed is multiplied by 16 to determine the number of pixels to either shift or terminate the display.

For example, if the Graphics Truncation Start register at Video Index 23h is written with a value of 0Fh, the Graphics Truncation Stop register at Video Index 24h is written with 00h, and bit 4 of the Control 1 register at Video Index 20h is 1, then a 640×200 LCD panel shows 240 pixels of data from the left side of the display, and the remaining 400 pixels on the right are blank ( $240 = (15 - 0) \cdot 16$ ). Data on the display is accessed normally, beginning with 0B8000h because the Graphics Truncation Stop register reads 00h.

If the Graphics Truncation Stop register is now written with a value of 03h, the display shows 192 pixels of data from the left side of the display, and the remaining 448 pixels are blank ( $192 = (15 - 3) \cdot 16$ ).

Data on the display is accessed beginning with 0B8006h because the Graphics Truncation Stop register now reads 03h. Each byte in memory is 4 or 8 pixels of information on the display, depending on whether 1-bpp or 2-bpp mode is currently active.

### 3.9.13 LCD Special Register (Video Index 25h)

This register controls various functions specific to the LCD panel. The recommended setting for LCD modulation control is bit 1 = 1 and bit 5 = 0 (every 13 lines).

	7							0
<b>Bit</b>	(Reserved)		MODFRQ	(Reserved)	STOPDIS	SRAMCS	LCDMOD	VERTRET
<b>Reset</b>	0	0	0	0	0	0	0	0

Bit	Name	R/W	Function
7–6		W	(Reserved—must be 0)
5	MODFRQ	W	LCD modulation frequency selector (bit 1 must be 1): 0 = Frequency of M = once per 13 lines 1 = Frequency of M = once per 61 lines
4		W	(Reserved—must be 0)
3	STOPDIS	W	Stop display clock (stops all clocks to video controller): 0 = Normal 1 = Stop display clock
2	SRAMCS	W	SRAM $\overline{CS}$ control: 0 = Normal operation 1 = SRAM chip-select remains inactive during Sleep mode
1	LCDMOD	W	LCD modulation control: 0 = Frequency of M = once per frame 1 = Frequency of M dependent on bit 5 of this register
0	VERTRET	W	Vertical-retrace detect option 0 = Normal 1 = Toggle bit 3 of Mode Control registers (ports 3D8h and 3B8h) during read

### 3.9.14 Programming Examples

Tables 3-35–3-39 provide typical display configurations for the LCD controller.

Table 3-35 640×200 LCD Panel Configuration

		<b>CGA (Video Index 18h, Bit 0 = 0)</b>			
<b>Register Name</b>	<b>Port or Index</b>	<b>Text Mode</b>		<b>Graphics Mode</b>	
		<b>40×25</b>	<b>80×25</b>	<b>320×200</b>	<b>640×200</b>
CGA Mode Control	3D8h	28h	29h	0Ah	1Ah
CGA Color Select	3D9h	00h	00h	00h	0Fh
HGA Configuration	3BFh	(NA)	(NA)	(NA)	(NA)
Horizontal Total	00h	28h	50h	28h	28h
Horizontal Displayed	01h	28h	50h	28h	28h
Horizontal Sync Position	02h	00h	00h	00h	00h
Horizontal Sync Width	03h	00h	00h	00h	00h
Vertical Total	04h	19h	19h	64h	64h
Vertical Total Adjust	05h	06h	00h	00h	00h
Vertical Displayed	06h	19h	19h	64h	64h
Vertical Sync Position	07h	19h	19h	64h	64h
Interlace Mode	08h	02h	02h	02h	02h
Maximum Scan Line	09h	07h	07h	01h	01h
Cursor Start	0Ah	06h	06h	00h	00h
Cursor End	0Bh	07h	07h	00h	00h
Start Address Upper Byte	0Ch	00h	00h	00h	00h
Start Address Lower Byte	0Dh	00h	00h	00h	00h
Cursor Address Upper Byte	0Eh	00h	00h	00h	00h
Cursor Address Lower Byte	0Fh	00h	00h	00h	00h

Table 3-36 480×320 LCD Panel Configuration

		CGA (Video Index 18h, Bit 0 = 0)			
Register Name	Port or Index	Text Mode		Graphics Mode	
		30×40	60×40	2 bpp	1 bpp
CGA Mode Control	3D8h	28h	29h	(NA)	1Ah
CGA Color Select	3D9h	00h	00h	(NA)	0Fh
HGA Configuration	3BFh	(NA)	(NA)	(NA)	(NA)
Horizontal Total	00h	1Eh	3Ch	(NA)	1Eh
Horizontal Displayed	01h	1Eh	3Ch	(NA)	1Eh
Horizontal Sync Position	02h	00h	00h	(NA)	00h
Horizontal Sync Width	03h	00h	00h	(NA)	00h
Vertical Total	04h	28h	28h	(NA)	50h
Vertical Total Adjust	05h	00h	00h	(NA)	00h
Vertical Displayed	06h	28h	28h	(NA)	50h
Vertical Sync Position	07h	28h	28h	(NA)	50h
Interlace Mode	08h	00h	00h	(NA)	00h
Maximum Scan Line	09h	07h	07h	(NA)	03h
Cursor Start	0Ah	06h	06h	(NA)	00h
Cursor End	0Bh	07h	07h	(NA)	00h
Start Address Upper Byte	0Ch	00h	00h	(NA)	00h
Start Address Lower Byte	0Dh	00h	00h	(NA)	00h
Cursor Address Upper Byte	0Eh	00h	00h	(NA)	00h
Cursor Address Lower Byte	0Fh	00h	00h	(NA)	00h

Table 3-37 320×240 LCD Panel Configuration

		<b>CGA (Video Index 18h, Bit 0 = 0)</b>			
<b>Register Name</b>	<b>Port or Index</b>	<b>Text Mode</b>		<b>Graphics Mode</b>	
		<b>20×36</b>	<b>40×30</b>	<b>2 bpp</b>	<b>1 bpp</b>
CGA Mode Control	3D8h	28h	29h	0Ah	1Ah
CGA Color Select	3D9h	00h	00h	00h	07h
HGA Configuration	3BFh	(NA)	(NA)	(NA)	(NA)
Horizontal Total	00h	14h	28h	14h	14h
Horizontal Displayed	01h	14h	28h	14h	14h
Horizontal Sync Position	02h	00h	00h	00h	00h
Horizontal Sync Width	03h	00h	00h	00h	00h
Vertical Total	04h	1Eh	1Eh	3Ch	78h
Vertical Total Adjust	05h	00h	00h	00h	00h
Vertical Displayed	06h	1Eh	1Eh	3Ch	78h
Vertical Sync Position	07h	1Eh	1Eh	3Ch	78h
Interlace Mode	08h	00h	00h	00h	00h
Maximum Scan Line	09h	07h	07h	01h	01h
Cursor Start	0Ah	06h	06h	00h	00h
Cursor End	0Bh	07h	07h	00h	00h
Start Address Upper Byte	0Ch	00h	00h	00h	00h
Start Address Lower Byte	0Dh	00h	00h	00h	00h
Cursor Address Upper Byte	0Eh	00h	00h	00h	00h
Cursor Address Lower Byte	0Fh	00h	00h	00h	00h

Table 3-38 320×240 LCD Panel Configuration

		<b>HGA (Video Index 18h, Bit 0 = 1)</b>		
<b>Register Name</b>	<b>Port or Index</b>	<b>Text Mode</b>	<b>Graphics Mode</b>	<b>Graphics Mode</b>
		<b>35.5×30 9×8 Font</b>	<b>320×240 2 Banks</b>	<b>320×240 4 Banks</b>
HGA Mode Control	3B8h	28h	0Ah	0Ah
CGA Color Select	3D9h	(NA)	(NA)	(NA)
HGA Configuration	3BFh	00h	01h	01h
Horizontal Total	00h	28h	14h	14h
Horizontal Displayed	01h	28h	14h	14h
Horizontal Sync Position	02h	00h	00h	00h
Horizontal Sync Width	03h	00h	00h	00h
Vertical Total	04h	1Eh	78h	3Ch
Vertical Total Adjust	05h	00h	00h	00h
Vertical Displayed	06h	1Eh	78h	3Ch
Vertical Sync Position	07h	1Eh	78h	3Ch
Interlace Mode	08h	00h	00h	00h
Maximum Scan Line	09h	07h	01h	03h
Cursor Start	0Ah	00h	00h	00h
Cursor End	0Bh	00h	00h	00h
Start Address Upper Byte	0Ch	00h	00h	00h
Start Address Lower Byte	0Dh	00h	00h	00h
Cursor Address Upper Byte	0Eh	00h	00h	00h
Cursor Address Lower Byte	0Fh	00h	00h	00h

Table 3-39 720×348 LCD Panel Configuration

		<b>HGA (Video Index 18h, Bit 0 = 1)</b>	
<b>Register Name</b>	<b>Port or Index</b>	<b>Text Mode</b>	<b>Graphics Mode</b>
		<b>80×25</b>	<b>720×348</b>
HGA Mode Control	3B8h	28h	0Ah
CGA Color Select	3D9h	(NA)	(NA)
HGA Configuration	3BFh	00h	00h
Horizontal Total	00h	50h	2Dh
Horizontal Displayed	01h	50h	2Dh
Horizontal Sync Position	02h	00h	00h
Horizontal Sync Width	03h	00h	00h
Vertical Total	04h	19h	57h
Vertical Total Adjust	05h	00h	00h
Vertical Displayed	06h	19h	57h
Vertical Sync Position	07h	19h	57h
Interlace Mode	08h	00h	00h
Maximum Scan Line	09h	0Dh	03h
Cursor Start	0Ah	0Bh	00h
Cursor End	0Bh	0Ch	00h
Start Address Upper Byte	0Ch	00h	00h
Start Address Lower Byte	0Dh	00h	00h
Cursor Address Upper Byte	0Eh	00h	00h
Cursor Address Lower Byte	0Fh	00h	00h





The ÉlanSC300 microcontroller contains internal registers used to display the status of various ÉlanSC300 microcontroller internal states, to serve as the target for software commands, to act as data paths to external peripherals, and to access other registers.

All ÉlanSC300 microcontroller's registers have an address. A small number of registers have explicit I/O addresses—that is, their location is fixed in the I/O address space of the Am386SXLV processor. However, most of the registers are addressed via an indirect addressing scheme in which a few registers with actual I/O addresses are used to point to the others. This system is explained below in the sections where it is relevant.

For purposes of this manual, we have divided the ÉlanSC300 microcontroller's register set into four logical groups of various sizes. The four big groups in turn contain subgroups that are explained below.

The four groups consist of two small groups whose registers have explicit I/O addresses, and two much larger groups whose registers are addressed indirectly:

- Core AT-compatible peripheral registers (and miscellaneous AT-compatible registers)
- UART registers
- Configuration registers
- LCD registers (described in Chapter 3, "Video Controller")

Registers in the first two groups are directly addressed in I/O space. Registers in the second two groups have an index value instead of an I/O address.

**Notes:**

1. *When using a logic analyzer to probe the address, data, and control lines of the ÉlanSC300 microcontroller, accesses to the internal registers can be captured with the exception of reads from even addresses. However, when internal registers are accessed,  $\overline{DBUFOE}$  is not asserted.*
2. *In general, the ÉlanSC300 microcontroller's cores decode only up to address bit A9 for I/O accesses. What this means is that the I/O address space is mirrored every 400h bytes. Care should be taken when assigning peripherals I/O addresses above 3FFh, so that conflicts with the mirrored I/O registers are avoided.*

## 4.1 PC/AT-COMPATIBLE PORT REGISTERS

The core peripherals of the ÉlanSC300 microcontroller are four standard integrated circuits and logical components compatible with the IBM PC/AT motherboard:

- 8259A Programmable Interrupt Controller
- 8237A DMA Controller
- 8254 Counter-Timer
- PC/AT Standard Parallel Port

In addition, one other logical component is often grouped with the core peripherals:

- Miscellaneous PC/AT-Compatible registers

In this section and the next, these five components are described. The miscellaneous PC/AT-compatible registers are given their own section because these registers are discussed in detail. Only the I/O addresses are shown for the first three components, and the reader is directed to standard data sheets for more details.

### 4.1.1 Interrupt Controller Registers

The ÉlanSC300 microcontroller has two integrated interrupt controllers. These registers are PC/AT-compatible versions of the industry-standard 8259A programmable interrupt controller. I/O addresses for the interrupt controller are shown in Table 4-1 on page 4-2.

Table 4-1 Interrupt Controller 1

Port	R/W	Description
020h	W	Initialization Control Word 1, IC1 (D4 = 1)
021h	W	Initialization Control Word 3, Master, IC1
021h	W	Initialization Control Word 3, Slave, IC1
021h	W	Initialization Control Word 4, IC1
021h	R/W	Operation Control Word 1, (IMR), IC1
020h	W	Operation Control Word 2, IC1 (D5,4 = 00)
020h	W	Operation Control Word 3, IC1 (D5,4 = 01)
020h	R	Interrupt Request Register (IRR), IC1
020h	R	In Service Register (ISR), IC1; State Register, IC1

**Note:**

Refer to the 8259A Data Sheet for an explanation of the control words.

Table 4-2 Interrupt Controller 2

Port	R/W	Description
0A0h	W	Initialization Control Word 1, IC2 (D4=1)
0A1h	W	Initialization Control Word 2, IC2
0A1h	W	Initialization Control Word 3, Master, IC2
0A1h	W	Initialization Control Word 3, Slave, IC2
0A1h	W	Initialization Control Word 4, IC2
0A1h	R/W	Operation Control Word 1, (IMR), IC2
0A0h	W	Operation Control Word 2, IC2 (D5,4=00)
0A0h	W	Operation Control Word 3, IC2 (D5,4=01)
0A0h	R	Interrupt Request Register (IRR), IC2
0A0h	R	In Service Register (ISR), IC2; State Register, IC2

**Note:**

Refer to the 8259A Data Sheet for an explanation of the control words.

#### 4.1.2 Programmable Interval Timer Registers

The timer that is integrated into the ÉlanSC300 microcontroller is a PC/AT-compatible version of the industry-standard 8254 programmable interval timer. This counter occupies I/O addresses 040–043h. The timer register definition and locations are shown in Table 4-3 on page 4-3.

Table 4-3 System Timer Registers

Port	R/W	Description
040h	R/W	Timer/Counter 0 Count Register
041h	R/W	Timer/Counter 1 Count Register
042h	R/W	Timer/Counter 2 Count Register
043h	R/W	Timer Control Word Register

**Note:**

Refer to the 8254 Data Sheet for an explanation of the control words.

The input clock to the ÉlanSC300 microcontroller's 8254 timer runs at 1.1892 MHz. This deviates from the AT-compatible standard of 1.19318 MHz. This 0.4% difference can result in incorrect DOS clock readings over time if the difference is not taken into account.

In AT-compatible systems, the BIOS loads a value of 0FFFFh into Timer 0, resulting in a timer interrupt every 54.93 ms. To program the ÉlanSC300 microcontroller to generate the Timer-0 interrupt at the same rate, load Timer 0 with a value of 0FF23h.

A second option is to write the BIOS timer tick handler to reload the DOS clock at regular intervals from the real-time clock (RTC), which maintains accurate time.

### 4.1.3 DMA Controller Registers

The two DMA controllers that are integrated in the ÉlanSC300 microcontroller are PC/AT-compatible versions of the industry-standard 8237A DMA controller. DMA controller addresses are shown in Table 4-4 on page 4-4.

Table 4-4 DMA Controller 1

Port	R/W	Description
000h	R/W	CH0 Base Address
001h	R/W	CH0 Base Word Count
002h	R/W	CH1 Base Address
003h	R/W	CH1 Base Word Count
004h	R/W	CH2 Base Address
005h	R/W	CH2 Base Word Count
006h	R/W	CH3 Base Address
007h	R/W	CH3 Base Word Count
008h	R/W	CH3-0 Read Status Register/Write Command Register
009h	R/W	CH3-0 Write Request Register
00Ah	W	CH3-0 Write Single Mask Register Bit
00Bh	W	CH3-0 Write Mode Register
00Ch	W	CH3-0 Clear Byte Pointer Flip-Flop
00Dh	R/W	CH3-0 Read Temporary Register/Write Master Clear
00Eh	R/W	CH3-0 Clear Mask Register
00Fh	W	CH3-0 Write All Mask Register Bits

**Note:**

Refer to the 8237A Data Sheet for an explanation of these registers.

Table 4-5 DMA Controller 2

Port	R/W	Description
0C0h	R/W	CH4 Base Address
0C2h	R/W	CH4 Base Word Count
0C4h	R/W	CH5 Base Address
0C6h	R/W	CH5 Base Word Count
0C8h	R/W	CH6 Base Address
0CAh	R/W	CH6 Base Word Count
0CCh	R/W	CH7 Base Address
0CEh	R/W	CH7 Base Word Count
0D0h	R/W	Read Status Register/Write Command Register
0D2h	W	Write Request Register
0D4h	W	Write Single Mask Register Bit
0D6h	W	Write Mode Register
0D8h	W	Clear Byte Pointer Flip-Flop
0DAh	R/W	Read Temporary Register/Write Master Clear
0DCh	W	Clear Mask Register
0DEh	W	Write All Mask Register Bits

**Note:**

Refer to the 8237A Data Sheet for an explanation of these registers.

#### 4.1.4 DMA Page Registers

DMA page registers provide the upper address bits during DMA transfers. The processor writes the page registers before enabling DMA transfers. Addresses for the page registers are shown in Table 4-6 on page 4-6.

Table 4-6 DMA Page Registers

Address	R/W	Description
0080h	R/W	General Register
0081h	R/W	Channel 2 Page Register
0082h	R/W	Channel 3 Page Register
0083h	R/W	Channel 1 Page Register
0084h	R/W	General Register
0085h	R/W	General Register
0086h	R/W	General Register
0087h	R/W	Channel 0 Page Register
0088h	R/W	General Register
0089h	R/W	Channel 6 Page Register
008Ah	R/W	Channel 7 Page Register
008Bh	R/W	Channel 5 Page Register
008Ch	R/W	General Register
008Dh	R/W	General Register
008Eh	R/W	General Register
008Fh	R/W	General Register

#### 4.1.5 Parallel Port Interface Registers

The parallel port is register-compatible with the industry-standard AT-compatible, EPP-compliant, parallel port. The parallel port can be set up to have a base address of 3BCh, 278h, or 378h. EPP support is only possible when the base address is set to 278h or 378h. EPP-mode enable and bidirectional enable/control are set by bits 2–0 of the Function Enable 1 register at Index B0h. The parallel port interrupt is fixed at IRQ7.

The base address is selected bits 1–0 of the Extended Video Controller Index register, Control 1 register, Index 20h. For more information on this register, see Chapter 3, “Video Controller.” The register is accessed by writing 20h to the CGA Index Address register at Port 3D4h. The index can then be read from or written to using the CGA Index Data register at Port 3D5h.

Note that if the internal video controller is disabled, access to the Extended Video Controller Index registers will be permanently disabled. Therefore, the base address of the parallel port must be set before disabling the video controller.

AT-compatible, parallel-port register definitions are given in the following paragraphs.

#### 4.1.5.1 AT-Compatible Mode

##### 4.1.5.1.1 Parallel Data Port (Ports 278h, 378h, & 3BCh)

7 0

<b>Bit</b>	Data Bits 7–0
<b>Default</b>	

Bit	Name	R/W	Function
7–0		R/W	Parallel Port Data Register

##### 4.1.5.1.2 Parallel Status Port (Ports 279h, 379h, & 3BDh)

7 0

<b>Bit</b>	BUSY	ACK	PE	SLCT	ERR	(Reserved)		
<b>Default</b>	0	0	0	0	0	0	0	0

Bit	Name	R/W	Function
7	BUSY	R	0 = Printer Busy (active Low)
6	ACK	R	0 = Printer Acknowledge. The printer pulses this line Low when it has received a byte of data (active Low).
5	PE	R	1 = Paper End
4	SLCT	R	1 = Printer Selected
3	ERR	R	0 = Printer Error (active Low)
2–0		R	(Reserved)

##### 4.1.5.1.3 Parallel Control Port (Ports 27Ah, 37Ah, & 37Eh)

7 0

<b>Bit</b>	(Reserved)	DIR	IRQEN	SLCTIN	INIT	AUTOFDXT	STROBE
<b>Default</b>	0	0	0	0	0	0	0

Bit	Name	R/W	Function
7–6		R	(Reserved)
5	DIR	R/W	Bidirectional parallel port data direction: 0 = Out (normal printer) 1 = In
4	IRQEN	R/W	1 = Printer IRQ enable. Clearing this bit clears any pending interrupts
3	SLCTIN	R/W	1 = Printer selected 0 = Not selected
2	INIT	R/W	Printer reset signal control: 0 = Hold printer in reset 1 = Release printer from reset
1	AUTOFDXT	R/W	Auto line feed signal control: 1 = AUTOFDXT pin active
0	STROBE	R/W	Printer port strobe signal control: 1 = STROBE pin active

#### 4.1.5.2 EPP-Compliant Mode

##### 4.1.5.2.1 Parallel Data Port (Ports 278h & 378h)

7 0

<b>Bit</b>	Data Bits 7–0
<b>Default</b>	(Undefined)

Bit	Name	R/W	Function
7–0		R/W	Parallel Port Data Register

##### 4.1.5.2.2 Parallel Status Port (Ports 279h & 379h)

7 0

<b>Bit</b>	BUSY	ACK	PE	SLCT	ERR	ACKSTAT	(Reserved)	EPPTO
<b>Default</b>	0	0	0	0	0	0	0	0

Bit	Name	R/W	Function
7	BUSY	R	0 = Printer busy (active Low)
6	ACK	R	0 = Printer acknowledge. The printer pulses this line Low when it has received a byte of data (active Low).
5	PE	R	1 = Paper end
4	SLCT	R	1 = Printer selected
3	ERR	R	0 = Printer error (active Low)
2	ACKSTAT	R	This bit is latched Low when ACK transitions Low to High. Reading this bit sets it to 1.
1		R	(Reserved)
0	EPPTO	R	EPP time-out status: 0 = No time-out 1 = EPP cycle time-out occurred  This bit is reset when either the status register is read, or when EPP mode is enabled.

##### 4.1.5.2.3 Parallel Control Port (Ports 27Ah & 37Ah)

7 0

<b>Bit</b>	(Reserved)	DIR	IRQEN	(Reserved)	INIT	(Reserved)
<b>Default</b>	0	0	0	0	0	0

Bit	Name	R/W	Function
7–6		R	(Reserved)
5	DIR	R/W	Bidirectional parallel port data direction: 0 = Out (normal printer) 1 = In
4	IRQEN	R/W	1 = Printer IRQ enable. Clearing this bit clears any pending interrupts
3		R	(Reserved)
2	INIT	R/W	Printer reset signal control: 0 = Hold printer in reset 1 = Release printer from reset
1–0		R	(Reserved)



4.1.5.2.4 Parallel EPP Address Port (Ports 27Bh & 37Bh)

7 0

<b>Bit</b>	Address Bits 7–0
<b>Default</b>	(Undefined)

Bit	Name	R/W	Function
7–0		R/W	Address Register

4.1.5.2.5 Parallel EPP 32-Bit Data Register (Ports 27C–27Fh & 37C–37Fh)

A 32-bit I/O write to 27Ch or 37Ch causes four back-to-back 8-bit bus cycles to occur to the four EPP data registers. An EPP data strobe is generated for all four bus cycles. A 16-bit I/O write to 27Ch or 37Ch causes two back-to-back 8-bit bus cycles to occur to the EPP data registers 27Ch or 37Ch and 27Dh or 37Dh. In common practice, all write accesses to the parallel port EPP data register (8-, 16-, or 32-bit I/O instructions) should be directed to port 27Ch or 37Ch.

7 0

<b>Bit</b>	Address Bits 7–0
<b>Default</b>	(Undefined)

Bit	Name	R/W	Function
7–0		R/W	Address Register

4.1.6 UART Registers

The UART implemented in the ÉlanSC300 microcontroller is register compatible with the industry-standard 16450 universal asynchronous receiver/transmitter.

4.1.6.1 Transmitter Holding Register (Ports 2F8h & 3F8h)

This write-only register contains the data waiting to be transferred. Bit 7 is the most significant bit; bit 0 is the least significant bit.

7 0

<b>Field</b>	Data to be Transmitted							
<b>Default</b>	0	0	0	0	0	0	0	0

4.1.6.2 Receiver Buffer Register (Ports 2F8h & 3F8h)

This read-only register contains the data received. Bit 7 is the most significant bit; bit 0 is the least significant bit.

7 0

<b>Field</b>	Data to be Received							
<b>Default</b>	0	0	0	0	0	0	0	0

#### 4.1.6.3 Divisor Latch Lower Byte (Ports 2F8h & 3F8h)

This register controls the programmable baud-rate generator that is capable of dividing 1.843 MHz by values from 1 to  $(2^{16} - 1)$ . Two 8-bit registers store the divisor; this read/write register contains the lower byte.

	7							0
<b>Field</b>	Baud-Rate Divisor, Lower Byte							
<b>Default</b>	0	0	0	0	0	0	0	0

#### 4.1.6.4 Divisor Latch Upper Byte (Ports 2F8h & 3F9h)

This register controls the programmable baud-rate generator that is capable of dividing 1.843 MHz (I/P OSC) by values from 1 to  $(2^{16} - 1)$ . Two 8-bit registers store the divisor; this read/write register contains the upper byte.

	7							0
<b>Field</b>	Baud-Rate Divisor, Upper Byte							
<b>Default</b>	0	0	0	0	0	0	0	0

#### 4.1.6.5 Interrupt Enable Register (Ports 2F9h & 3F9h)

This register is used to enable UART functions.

	7					0		
<b>Bit</b>	(Reserved)				EMSI	ELSI	ETDEI	ERDI
<b>Default</b>	0	0	0	0	0	0	0	0

Bit	Name	R/W	Function
7-4		W	(Reserved)
3	EMSI	W	Enable modem status interrupt
2	ELSI	W	Enable receiver line status interrupt
1	ETDEI	W	Enable Transmitter holding register empty interrupt
0	ERDI	W	Enable receiver data ready interrupt

#### 4.1.6.6 Interrupt Identification Register (Ports 2FAh & 3FAh)

This register is used to identify the nature of an interrupt.

	7					0		
<b>Field</b>						Interrupt ID		
<b>Bit</b>	(Reserved)					ID1	ID0	IP
<b>Default</b>	0	0	0	0	0	0	0	

Bit	Name	R/W	Function
7-3		R	(Reserved)
2	ID1	R	Interrupt ID, bit 1
1	ID0	R	Interrupt ID, bit 0
0	IP	R	0 = Interrupt pending

Table 4-7 Interrupt ID Bit Logic

ID1	ID0	Priority	Interrupt Type
1	1	1	Receiver line status
1	0	2	Received data ready
0	1	3	Transmitter holding register empty
0	0	4	Modem status

4.1.6.7 Line Control Register (Ports 2FBh & 3FBh)  
This register controls certain line characteristics.

Field	7						0	
	DLAB	SB	SP	EPS	PE	STP	WLB1	WLB0
Default	0	0	0	0	0	0	0	0

Bit	Name	R/W	Function
7	DLAB	R/W	Divisor latch access bit 0: 1 = Enable divisor programming 0 = Disable divisor programming
6	SB	R/W	Set break bit
5	SP	R/W	Stuck parity: 1 = Enable
4	EPS	R/W	Even parity: 1 = Enable
3	PE	R/W	Parity enable: 1 = Enable
2	STP	R/W	Stop bit: 0 = 1 stop bit 1 = 1.5 stop bits if word length is 5 (2 stop bits if word length is 6, 7, or 8)
1	WLB1	R/W	Word length, bit 1
0	WLB0	R/W	Word length, bit 0

Table 4-8 Word Length Bit Logic

Bit 1	Bit 0	Word Length (Bits)
0	0	5
0	1	6
1	0	7
1	1	8

#### 4.1.6.8 Modem Control Register (Ports 2FCh & 3FCh)

This register governs certain modem characteristics.

	7							0
Bit	(Reserved)			LOOP	$\overline{\text{OUT2}}$	OUT1	RTS	DTR
Default	0	0	0	0	0	0	0	0

Bit	Name	R/W	Function
7–5		R/W	(Reserved)
4	LOOP	R/W	Local loop-back diagnostic test: 1 = Enable loop 0 = Normal operation
3	$\overline{\text{OUT2}}$	R/W	$\overline{\text{OUT2}}$ (must be 1 to enable interrupts)
2	$\overline{\text{OUT1}}$	R/W	OUT1
1	RTS	R/W	$\overline{\text{RTS}}$ : 1 = Enable $\overline{\text{RTS}}$
0	DTR	R/W	$\overline{\text{DTR}}$ : 1 = Enable $\overline{\text{DTR}}$

**Note:**

*If loop-back is enabled, interrupts from the UART are disabled and the four modem control signals ( $\overline{\text{DTR}}$ ,  $\overline{\text{RTS}}$ ,  $\overline{\text{OUT1}}$ , and  $\overline{\text{OUT2}}$ ) are internally connected to  $\overline{\text{DSR}}$ ,  $\overline{\text{CTS}}$ ,  $\overline{\text{RI}}$ , and  $\overline{\text{DCD}}$ , respectively.*

#### 4.1.6.9 Line Status Register (Ports 2FDh & 3FDh)

This register reports the line status. This is a read-only register unless bit 1 of the Control A register at Index 48h is 1, in which case it becomes read/write.

	7							0
Bit	(Reserved)	TEMT	THRE	BI	FE	PE	OE	DR
Default	0	0	0	0	0	0	0	0

Bit	Name	R/W	Function
7		R	(Reserved)
6	TEMT	R	1 = Transmitter empty
5	THRE	R	1 = Transmitter holding register empty
4	BI	R	1 = Break interrupt
3	FE	R	1 = Framing error
2	PE	R	1 = Parity error
1	OE	R	1 = Overrun error
0	DR	R	1 = Data ready

## 4.1.6.10 Modem Status Register (Ports 2FEh &amp; 3FEh)

This register reports more status information.

	7							0
<b>Bit</b>	RLSD	RI	DSR	CTS	DRLSD	TERI	DDSR	DCTS
<b>Default</b>	0	0	0	0	0	0	0	0

Bit	Name	R/W	Function
7	RLSD	R	Status of $\overline{\text{DCD}}$ from modem
6	RI	R	Status of $\overline{\text{RI}}$ from modem
5	DSR	R	Status of $\overline{\text{DSR}}$ from modem
4	CTS	R	Status of $\overline{\text{CTS}}$ from modem
3	DRLSD	R	Delta data carrier detect
2	TERI	R	Trailing edge of ring indicator from modem
1	DDSR	R	Delta data set ready
0	DCTS	R	Delta clear to send

## 4.1.6.11 Scratch Pad Register (Ports 2FFh &amp; 3FFh)

This is a general-purpose register.

	7							0
<b>Field</b>	I/O Port Number							
<b>Default</b>	0	0	0	0	0	0	0	0

Bit	Name	R/W	Function
7-0		R/W	Scratch data

#### 4.1.7

### Real-Time Clock

- Counts seconds, minutes, and hours of day
- Counts days of the week, date, month, and year
- Binary or BCD representation of time, calendar, and alarm
- 12–24-hr clock with A.M. and P.M. in 12-hr mode
- Daylight-savings-time option
- Automatic end-of-month recognition
- Automatic leap-year compensation
- 14 bytes of clock and control registers
- 114 bytes of general-purpose RAM
- Three interrupts are separately software maskable and testable
- Time-of-day alarm (once-per-second to once-per-day)
- Periodic interrupt rates from 122 ms to 500 ms
- End-of-clock update cycle

The real-time clock (RTC) block in the ÉlanSC300 microcontroller is compatible with the Motorola MC146818A device used in PC/AT systems. This block consists of a time-of-day clock with an alarm and a 100-yr calendar. The clock/calendar can be represented in binary or BCD, has a programmable periodic interrupt, and has 114 bytes of static user RAM. The RTC block is powered from the same power as the core logic.

The block also has 10 registers for time, calendar, and alarm data and four general-purpose registers. The interrupt signal from the RTC is an active-Low signal and is routed to interrupt-request line 8.

#### 4.1.7.1

### Addressing

The RTC is accessed using 8-bit-wide I/O cycles at addresses 70h and 71h. The contents of the RTC registers and its 114 bytes of RAM are accessed in an indexed fashion.

Table 4-9 Real-Time Clock

Port	R/W	Register Name	Description
070h	W	RTC Index Address	Index (pointer) to the data in the RTC
071h	R/W	RTC Index Data	Data stored in the RTC

**Note:**

*Bit 7 of the NMI/RTC Index register, I/O address 70h, is the ENMI control bit. If this bit is set, NMIs are inhibited. Bits 6–0 of this register are used to access the RTC registers and RAM.*

4.1.7.2 RTC Registers

The RTC registers and RAM are indexed with the address map shown in Table 4-10 on page 4-15.

Table 4-10 RTC Register Summary

RTC Index	Function
0	Seconds
1	Seconds Alarm
2	Minutes
3	Minutes Alarm
4	Hours
5	Hours Alarm
6	Day of Week
7	Date of Month
8	Month
9	Year
10	Register A
11	Register B
12	Register C
13	Register D
14–127	User Static RAM

4.1.7.2.1 Register A (RTC Index 0Ah)

	7							0
<b>Bit</b>	UIP	DV2	DV1	DV0	RS3	RS2	RS1	RS0
<b>Default</b>	(Undefined)							

Bit	Name	Function
7	UIP	Update in Progress: This bit is a status bit that can be polled by a program to indicate that the clock/calendar is currently being updated.
6–4	DV2–DV0	Divider Chain: These bits select the time base frequency for the clock. (Table 4-12 on page 4-16 describes the operation of these bits.)
3–0	RS3–RS0	Rate Selection: These bits select the periodic interrupt rate. (Table 4-11 on page 4-16 describes the operation of these bits.)

Table 4-11 Register A Periodic-Interrupt Rate-Selection Bits (32.768 kHz)

RS3	RS2	RS1	RS0	Periodic Interrupt Rate
0	0	0	0	(None)
0	0	0	1	(Reserved)
0	0	1	0	(Reserved)
0	0	1	1	122.070 $\mu$ s
0	1	0	0	244.141 $\mu$ s
0	1	0	1	488.281 $\mu$ s
0	1	1	0	976.562 $\mu$ s
0	1	1	1	1.953125 ms
1	0	0	0	3.90625 ms
1	0	0	1	7.8125 ms
1	0	1	0	15.625 ms
1	0	1	1	31.25 ms
1	1	0	0	62.5 ms
1	1	0	1	125 ms
1	1	1	0	250 ms
1	1	1	1	500 ms

Table 4-12 Register A Time-Base Divider-Chain Bits

DV2	DV1	DV0	Time Base Frequency
0	1	0	32.768 kHz (AT-compatible setting)

**Note:**

*Other combinations of divider bits are used for test purposes only.*



## 4.1.7.2.2 Register B (RTC Index 0Bh)

	7							0
<b>Bit</b>	SET	PIE	AIE	UIE	SQWE	DM	24/12	DSE
<b>Default</b>	(Undefined)							

Bit	Name	Function
7	SET	Set Time: This bit prevents the time calendar from being incremented.
6	PIE	Programmable Interrupt Enable: This bit enables the periodic interrupt flag (PF) in Register C to generate an interrupt request.
5	AIE	Alarm Interrupt Enable: This bit enables the alarm flag (AF) in Register C to generate an interrupt request.
4	UIE	Update-Ended Enable: This bit enables the update-ended flag (UF) in Register C to generate an interrupt request.
3	SQWE	Square Wave Enable: This bit is not used.
2	DM	Data Mode: This bit selects between BCD and binary formats for the time and calendar: 1 = BCD 0 = Binary
1	24/12	Hour Format: This bit establishes the format of the hours byte for 24- or 12-hr time: 1 = 24-hr clock 0 = 12-hr clock
0	DSE	Daylight Savings Time: This bit enables the clock to automatically account for daylight savings time adjustments: 1 = Enabled 0 = Disabled

## 4.1.7.2.3 Register C (RTC Index 0Ch)

	7							0
<b>Bit</b>	IRQF	PF	AF	UF	(Reserved)			
<b>Default</b>					0	0	0	0

Bit	Name	Function
7	IRQF	Interrupt Request Flag: This bit is read only and is set when any of these Register C bits are set: PF, AF, or UF.
6	PF	Periodic Interrupt Flag: This bit is read only and is set according to the rate set by the RS3–RS0 bits in Register A.
5	AF	Alarm Flag: This bit is read only and is set when the current item matches the alarm time.
4	UF	Update-Ended Flag: This bit is read only and is set at the end of each time-update cycle.
3–0		(Reserved)

## 4.1.7.2.4 Register D (RTC Index 0Dh)

	7							0
<b>Bit</b>	VRT	(Reserved)						
<b>Default</b>		0	0	0	0	0	0	0

Bit	Name	Function
7	VRT	Valid RAM and Time: This bit is read only and is reset when the $\overline{\text{RESIN}}$ pin is asserted Low. This bit is set by reading Register D.
6–0		(Reserved)

## 4.2 MISCELLANEOUS PC/AT-COMPATIBLE PORT REGISTERS

### 4.2.1 XT Keyboard Data Register (Port 060h)

The XT Keyboard Data register is located at I/O address 060h. This read-only 8-bit register holds the data from the XT keyboard. The XT keyboard is enabled by setting bit 3 (XTKBDEN) of the PMU Control 3 register at Index ADh. While the XT keyboard is enabled, this register is cleared by setting bit 7 of the Port B register.

	7							0
<b>Field</b>	Data Received from the XT Keyboard							
<b>Default</b>	0	0	0	0	0	0	0	0

### 4.2.2 Port B Register (Port 061h)

Port B is an AT-standard, miscellaneous-feature control register which is located at I/O address 061h. The lower 4 bits of the 8-bit register are read/write control bits which enable or disable NMI check-condition sources and sound-generation features. The most significant 4 bits are read-only bits that return status and diagnostic information. Bit 4 (RFD) toggles state with every refresh. Bit 5 (T2OUT) follows the state of the Timer 2 output. Bit 6 (IOCHCK) is set upon detection of a channel check and is cleared upon the reading of this register.

Bits 6 and 7 of this register have alternate functions when the XT keyboard feature is enabled by setting bit 3 of Index ADh.

	7						0	
<b>Bit</b>	(Reserved)	IOCHCK	T2OUT	RFD	EIC	(Reserved)	SPKD	T2G
<b>Default</b>	0	0	0	0	0	0	0	0

Bit	Name	R/W	Function
7		W	(Reserved)
(Alt.)		R/W	1 = Clear XT Keyboard Data register
6	IOCHCK	R	I/O channel check
(Alt.)	KBCLKEN	R/W	1 = Enable KBCLK as an output
5	T2OUT	R	Timer 2 output
4	RFD	R	Refresh detected
3	EIC	W/R	Enable I/O channel check: 0 = Enable
2			(Reserved)
1	SPKD	W/R	Speaker data: 1 = Enable speaker output
0	T2G	W/R	Timer 2 gate (speaker): 1 = Speaker gated on

### 4.2.3 NMI/RTC Index Address Register (Port 070h)

Both the PMU and the I/O channel check are possible sources for the generation of an NMI to the internal CPU. The master NMI enable function can inhibit any NMIs from reaching the CPU regardless of the state of the individual source enables. The NMI enable bit in this register is a write-only bit, and it has an active-0 sense. The default value for the NMI enable bit is 1, which inhibits NMI generation. A write to this I/O address must always be followed by a write to—or read from—address 71h to ensure proper operation of the RTC.

	7							0
Bit	ENMI	RTC Index Address Bits 6–0						
Default	1	0	0	0	0	0	0	0

Bit	Name	R/W	Function
7	ENMI	W	0 = Enable NMI 1 = Disable NMI (default)
6–0		W	RTC index address

**Note:**

For a definition of the index values, see “Real-Time Clock” on page 4-14.

### 4.2.4 RTC Index Data Register (Port 071h)

This register is used to read data from, or write data to, the real-time clock indexed by bits 6–0 in I/O register 070h.

	7							0
Bit	RTC Data Bits 7–0							
Default	0	0	0	0	0	0	0	0

Bit	Name	R/W	Function
7–0		R/W	RTC data bits 7–0

### 4.2.5 Port 92 (Port 092h)

	7							0
Bit	(Reserved)						ALTA20	HOTRST
Default	0	0	0	0	0	0	0	

Bit	Name	R/W	Function
7–2		R	(Reserved)
1	ALTA20	W	A20 gate control: 1 = A20 is CPUA20 0 = A20 is 0 if bit 0 of Index 6Fh is 0 and the A20GATE pin is 0
0	HOTRST	W	A Low-to-High transition in this bit causes CPURST to be asserted for 16 CPU clock cycles.





---

The ÉlanSC300 microcontroller's configuration registers are defined as those registers that control the five central functions:

- General Control
- ISA Bus Control
- Memory Control Unit (MCU)
- Power Management Unit (PMU)
- PCMCIA Control

These registers provide a uniform method of accessing the device's control and configuration parameters. The parameters are mapped into bits and bit fields contained in logical groupings of 8-bit registers.

Most of the configuration registers are set to a default value by a system reset condition. The default values have been specified to allow the device to correctly execute BIOS code out of the local Flash memory or ROM upon exiting the reset state.

## 5.1 CONFIGURATION REGISTER OVERVIEW

There are two basic types registers that are used to configure the ÉlanSC300 microcontroller:

- Port registers
- Index registers

### 5.1.1 Port Registers

The configuration port registers are located in I/O space. They are addressed directly by the ÉlanSC300 microcontroller. The configuration port registers are listed in Table 5-1 on page 5-2. For a description of each port register, see “ConfigurAtion Port Registers” on page 5-12.

Table 5-1 Configuration Port Registers

Port	Register
022h	Configuration Address
023h	Configuration Data

The Configuration Address register points to the index register to be read or written when a data-register I/O cycle occurs. The Configuration Data register is used to read or write the index register.

Reading the Configuration Data register reads the contents of the index register. Writing a value to the Configuration Data register writes that value to the index register.

The contents of the Configuration Address register is static relative to multiple data register reads and writes. This allows for read-modify-write operations. The address of the index register is written to the Configuration Address register, and then the data is read, modified, and written back to the Configuration Data register without rewriting the Configuration Address register.

### 5.1.2 Index Registers

The configuration index registers are located in a separate internal memory space. They are addressed indirectly by the ÉlanSC300 microcontroller. For a description of each index register, see “Configuration Index Registers” on page 5-13.

#### 5.1.2.1 Mandatory Settings

Several of the configuration index registers contain bits that must be written soon after reset with their specified values in order for the device to operate as specified. These mandatory bits control things such as chip-test modes, interrupt mapping, SMI generation, and cycle timings.

Default settings of the microcontroller guarantee that code can be fetched and executed out of the ROM BIOS address space and that the configuration index registers can be accessed upon power up. Setting of these mandatory bits must be done prior to enabling any other functions of the device. Table 5-2 on page 5-3 shows the following:

- Mandatory bit settings for any speed operation
- Additional mandatory bit settings for 33-MHz system speed in a DRAM-configured system
- Mandatory bit settings for 33-MHz and 25-MHz system speed, independent of DRAM or SRAM use

Table 5-2 Mandatory Configuration Bit Settings

Index	Bit Settings (x = Don't Care)
0Fh	1 1 1 1 1 1 1 1
44h	1 0 x x 0 0 0 x
51h	x x x x 0 0 x x
60h	x x 0 x 0 x x x
62h	0 x x x x x x x
63h	x x x x x 0 x
64h	1 0 0 x 1 1 x x
6Ah	0 0 0 0 0 0 0 0
6Bh	0 x x 1 x x x x
74h	x x 0 0 x x x x
80h	x x x x 0 x x x
8Fh	1 x x 0 x x x x
93h	0 0 0 0 0 0 0 0
9Dh	0 1 0 0 0 0 0 0
B1h	x x 0 x x x x x
B4h	x 1 x x x x x x
<b>Additional 33-MHz Bit Settings (Using DRAM)</b>	
60h	x 0 0 x 0 x x x
62h	0 x x 1 x x x x
63h	x 1 1 x x x 0 x
65h	x x 1 x x x x x
<b>Additional 33-MHz and 25-MHz Bit Settings</b>	
6Bh	0 x x 1 x 0 x x

### 5.1.2.2 Recommended Settings

Table 5-3 on page 5-4 contains recommended bit settings that improve the performance of the microcontroller under certain conditions. These bits should be set in addition to the bits listed in Table 5-2 on page 5-3.

Table 5-3 Recommended Configuration Bit Settings

Index	Bit Settings (x = Don't Care)
44h	x x x 1 x x x x
63h	x x x x 1 1 x x
A7h	x x 1 1 x x x 0

### 5.1.3 Configuring the ÉlanSC300 Microcontroller

When you are ready to configure the ÉlanSC300 microcontroller, you program its configuration index registers according to your system's requirements. The following sections identify the index registers—and the bits within them—that you use to control different aspects of the microcontroller.

#### 5.1.3.1 To Control the PC/AT Bus and Its Timing

See	And program bits
"Control A Register (Index 48h)" on page 5-30	2
"Command Delay Register (Index 60h)" on page 5-32	2–0
"I/O Wait State Register (Index 61h)" on page 5-34	5–0
"MMS Memory Wait State 1 Register (Index 62h)" on page 5-35	3–0
"Wait State Control Register (Index 63h)" on page 5-37	3–2
"Miscellaneous 5 Register (Index B3h)" on page 5-81	6, 5–4, and 2
"ROM Configuration 3 Register (Index B8h)" on page 5-84	7–0

#### 5.1.3.2 To Determine the Bus Configuration

See	And program bits
"Memory Configuration 1 Register (Index 66h)" on page 5-41	6–5



## 5.1.3.3 To Control CPU and PC/AT Compatibility

See	And program bits
“Control 1 Register (Video Index 20h)” on page 3-34	1–0
“Port 92 (Port 092h)” on page 4-19	1–0
“Control A Register (Index 48h)” on page 5-30	1
“Miscellaneous 2 Register (Index 6Bh)” on page 5-43	0
“Miscellaneous 1 Register (Index 6Fh)” on page 5-47	1–0
“Control B Register (Index 77h)” on page 5-55	7–4
“UART Clock Enable Register (Index 92h)” on page 5-63	0

## 5.1.3.4 To Control the Speed of the CPU

See	And program bits
“I/O Wait State Register (Index 61h)” on page 5-34	6
“Miscellaneous 2 Register (Index 6Bh)” on page 5-43	2
“Control B Register (Index 77h)” on page 5-55	3
“Auto Low-Speed Control Register (Index 9Fh)” on page 5-66	3–0
“PMU Control 3 Register (Index ADh)” on page 5-76	1–0
“PMU Control 2 Register (Index AFh)” on page 5-77	5
“Function Enable 2 Register (Index B1h)” on page 5-78	4–3

## 5.1.3.5 To Control Direct Memory Accesses

See	And program bits
“I/O Wait State Register (Index 61h)” on page 5-34	7
“Miscellaneous 1 Register (Index 6Fh)” on page 5-47	3–2
“Function Enable 1 Register (Index B0h)” on page 5-77	3

## 5.1.3.6 To Enable Interrupts and Specify How They Are Mapped

See	And program bits
“PCMCIA I/O Card IRQ Redirection Control A Register (Index 06h)” on page 5-14	7–3
“PCMCIA Status Change IRQ Enable Register (Index 0Dh)” on page 5-19	7–0
“PCMCIA Status Change IRQ Redirection Register (Index 0Eh)” on page 5-20	4–0
“PCMCIA I/O Card IRQ Redirection Control B Register (Index 16h)” on page 5-23	7–3
“Control B Register (Index 77h)” on page 5-55	7–6
“NMI/SMI Enable Register (Index 82h)” on page 5-56	7–0
“MMSB Socket Register (Index A9h)” on page 5-73	7
“PMU Control 3 Register (Index ADh)” on page 5-76	6
“Function Enable 1 Register (Index B0h)” on page 5-77	5–4
“PIRQ Configuration Register (Index B2h)” on page 5-79	7–0

### 5.1.3.7 To Set Up Memory Mapping (MMS Windows)

See	And program bits
“ROM Configuration 1 Register (Index 65h)” on page 5-40	6
“MMSA Address Extension 1 Register (Index 67h)” on page 5-42	7–0
“MMS Address Extension 1 Register (Index 6Ch)” on page 5-44	7–0
“MMS Address Register (Index 6Dh)” on page 5-44	7–0
“MMS Address Extension 2 Register (Index 6Eh)” on page 5-46	7–0
“MMSA Device 1 Register (Index 71h)” on page 5-49	7–0
“MMSA Device 2 Register (Index 72h)” on page 5-50	7–0
“MMSB Device Register (Index 73h)” on page 5-51	7–0
“MMSB Control Register (Index 74h)” on page 5-52	1–0
“MMSA Socket Register (Index A8h)” on page 5-72	7–0
“MMSB Socket Register (Index A9h)” on page 5-73	3–0

### 5.1.3.8 To Set Up the System DRAM

See	And program bits
“Miscellaneous 4 Register (Index 44h)” on page 5-27	4
“Command Delay Register (Index 60h)” on page 5-32	6
“MMS Memory Wait State 1 Register (Index 62h)” on page 5-35	4
“Wait State Control Register (Index 63h)” on page 5-37	6–4
“Version Register (Index 64h)” on page 5-38	4 and 1–0
“ROM Configuration 1 Register (Index 65h)” on page 5-40	7 and 5–4
“Memory Configuration 1 Register (Index 66h)” on page 5-41	4–0
“Shadow RAM Enable 1 Register (Index 68h)” on page 5-42	7–0
“Shadow RAM Enable 2 Register (Index 69h)” on page 5-43	7–0
“Miscellaneous 1 Register (Index 6Fh)” on page 5-47	7–4
“Miscellaneous 6 Register (Index 70h)” on page 5-48	0
“PMU Control 1 Register (Index A7h)” on page 5-71	1–0
“Function Enable 2 Register (Index B1h)” on page 5-78	7–6
“Miscellaneous 5 Register (Index B3h)” on page 5-81	3
“PCMCIA Card Reset Register (Index B4h)” on page 5-82	7
“Memory Configuration 2 Register (Index B9h)” on page 5-85	7–0
“Miscellaneous 3 Register (Index BAh)” on page 5-86	2

## 5.1.3.9 To Set Up the System SRAM

See	And program bits
“Wait State Control Register (Index 63h)” on page 5-37	4
“Memory Configuration 1 Register (Index 66h)” on page 5-41	4–0
“Miscellaneous 6 Register (Index 70h)” on page 5-48	0

## 5.1.3.10 To Set Up the Parallel Port

See	And program bits
“Control 1 Register (Video Index 20h)” on page 3-34	1–0
“Function Enable 1 Register (Index B0h)” on page 5-77	2–0
“Miscellaneous 3 Register (Index BAh)” on page 5-86	4–3

## 5.1.3.11 To Set Up the UART

See	And program bits
“Control A Register (Index 48h)” on page 5-30	1
“Control B Register (Index 77h)” on page 5-55	7–4
“UART Clock Enable Register (Index 92h)” on page 5-63	0

## 5.1.3.12 To Set Up the PCMCIA Card Interface

For additional memory setup information, see “To Set Up Memory Mapping (MMS Windows)” on page 5-6.

See	And program bits
“PCMCIA I/O Window A1 Lower Byte Start Register (Index 00h)” on page 5-13	7–0
“PCMCIA I/O Window A1 Lower Byte End Register (Index 01h)” on page 5-13	7–0
“PCMCIA I/O Window A1 Upper Byte Register (Index 02h)” on page 5-13	7–0
“PCMCIA I/O Window A2 Lower Byte Start Register (Index 03h)” on page 5-13	7–0
“PCMCIA I/O Window A2 Lower Byte End Register (Index 04h)” on page 5-14	7–0
“PCMCIA I/O Window A2 Upper Byte Register (Index 05h)” on page 5-14	7–0
“PCMCIA I/O Card IRQ Redirection Control A Register (Index 06h)” on page 5-14	2–0
“PCMCIA VPPA Address Register (Index 07h)” on page 5-15	7–0
“Resume Mask Register (Index 08h)” on page 5-16	7–6 and 1–0
“Resume Status Register (Index 09h)” on page 5-17	1–0
“PCMCIA Data Width Register (Index 0Ah)” on page 5-17	7–0
“PCMCIA Socket B Status Register (Index 0Ch)” on page 5-19	4–0
“PCMCIA Status Change IRQ Enable Register (Index 0Dh)” on page 5-19	7–0
“PCMCIA Status Change IRQ Redirection Register (Index 0Eh)” on page 5-20	4–0
“PCMCIA I/O Window B1 Lower Byte Start Register (Index 10h)” on page 5-21	7–0

(Continued)

See	And program bits
"PCMCIA I/O Window B1 Lower Byte End Register (Index 11h)" on page 5-22	7–0
"PCMCIA I/O Window B1 Upper Byte Register (Index 12h)" on page 5-22	7–0
"PCMCIA I/O Window B2 Lower Byte Start Register (Index 13h)" on page 5-22	7–0
"PCMCIA I/O Window B2 Lower Byte End Register (Index 14h)" on page 5-22	7–0
"PCMCIA I/O Window B2 Upper Byte Register (Index 15h)" on page 5-23	7–0
"PCMCIA I/O Card IRQ Redirection Control B Register (Index 16h)" on page 5-23	2–0
"PCMCIA VPPB Address Register (Index 17h)" on page 5-24	7–0
"MMS Memory Wait State 2 Register (Index 50h)" on page 5-30	7 and 5–3
"Wait State Control Register (Index 63h)" on page 5-37	2
"PCMCIA REGA Address Register (Index 8Ah)" on page 5-59	7–0
"PCMCIA REGB Address Register (Index 9Eh)" on page 5-66	7–0
"PCMCIA Socket A Status Register (Index A2h)" on page 5-68	5–4 and 2–0
" on page 5-70	7–0
"MMSA Socket Register (Index A8h)" on page 5-72	7–0
"MMSB Socket Register (Index A9h)" on page 5-73	3–0
"PCMCIA Card Reset Register (Index B4h)" on page 5-82	7, 5, and 2
"CA24–CA25 Control 1 Register (Index B5h)" on page 5-82	7–0
"CA24–CA25 Control 2 Register (Index B6h)" on page 5-83	7–0
"CA24–CA25 Control 3 Register (Index B7h)" on page 5-83	7–0
"Miscellaneous 3 Register (Index BAh)" on page 5-86	4

## 5.1.3.13 To Set Up the General-Purpose and PMC Pins

See	And program bits
"Miscellaneous 6 Register (Index 70h)" on page 5-48	6
"MMSB Control Register (Index 74h)" on page 5-52	2
"Power Control 1 Register (Index 80h)" on page 5-55	6 and 2
"Power Control 2 Register (Index 81h)" on page 5-56	6 and 2
"General-Purpose I/O 0 Register (Index 89h)" on page 5-59	7–0
"General-Purpose I/O Control Register (Index 91h)" on page 5-61	7–0
"General-Purpose I/O 2 Register (Index 94h)" on page 5-63	7–0
"General-Purpose I/O 3 Register (Index 95h)" on page 5-63	7–0
"General-Purpose I/O 1 Register (Index 9Ch)" on page 5-65	7–0
"CPU Status 0 Register (Index A3h)" on page 5-68	0
"CPU Status 1 Register (Index A4h)" on page 5-69	3
"Power Control 3 Register (Index ABh)" on page 5-74	7–0
"Power Control 4 Register (Index ACh)" on page 5-75	7–0
"PMU Control 3 Register (Index ADh)" on page 5-76	3
"PIRQ Configuration Register (Index B2h)" on page 5-79	7–4 and 3–0
"Miscellaneous 3 Register (Index BAh)" on page 5-86	4–3

## 5.1.3.14 To Control the Clocks (Phase-Locked Loops)

See	And program bits
"I/O Wait State Register (Index 61h)" on page 5-34	6
"Miscellaneous 2 Register (Index 6Bh)" on page 5-43	2
"MMSB Control Register (Index 74h)" on page 5-52	3
"Control B Register (Index 77h)" on page 5-55	3
"Power Control 1 Register (Index 80h)" on page 5-55	7
"Power Control 2 Register (Index 81h)" on page 5-56	7 and 3
"Clock Control Register (Index 8Fh)" on page 5-60	2–0
"UART Clock Enable Register (Index 92h)" on page 5-63	0
"Auto Low-Speed Control Register (Index 9Fh)" on page 5-66	3–0
"PMU Control 3 Register (Index ADh)" on page 5-76	3–0
"PMU Control 2 Register (Index AFh)" on page 5-77	5 and 0
"Function Enable 1 Register (Index B0h)" on page 5-77	6 and 3
"Function Enable 2 Register (Index B1h)" on page 5-78	4–2
"Miscellaneous 3 Register (Index BAh)" on page 5-86	3

### 5.1.3.15 To Control Power Management Activities and Events

See	And program bits
“Resume Mask Register (Index 08h)” on page 5-16	7–0
“Miscellaneous 4 Register (Index 44h)” on page 5-27	0
“PIO Address Register (Index 45h)” on page 5-27	7–0
“PIO Timer Register (Index 46h)” on page 5-27	7–6 and 3–0
“Drive Timer Register (Index 47h)” on page 5-29	7–0
“Miscellaneous 6 Register (Index 70h)” on page 5-48	5
“MMSB Control Register (Index 74h)” on page 5-52	7–6 and 3
“Activity Mask 1 Register (Index 75h)” on page 5-53	7–0
“Activity Mask 2 Register (Index 76h)” on page 5-54	7–0
“Power Control 1 Register (Index 80h)” on page 5-55	7–6 and 2
“Power Control 2 Register (Index 81h)” on page 5-56	7–6 and 3–2
“Software Mode Control Register (Index 88h)” on page 5-58	2–0
“I/O Activity Address 0 Register (Index 8Ch)” on page 5-59	7–0
“I/O Activity Address 1 Register (Index 8Dh)” on page 5-60	7–0
“Memory Write Activity Lower Boundary Register (Index 9Ah)” on page 5-64	7–2 and 0
“Memory Write Activity Upper Boundary Register (Index 9Bh)” on page 5-64	7–4 and 2–0
“Power Control 3 Register (Index ABh)” on page 5-74	7–0
“Power Control 4 Register (Index ACh)” on page 5-75	7–0
“PMU Control 3 Register (Index ADh)” on page 5-76	4
“PMU Control 2 Register (Index AFh)” on page 5-77	0

### 5.1.3.16 To Determine Power Management Status

See	And program bits
“Resume Status Register (Index 09h)” on page 5-17	5–0
“Activity Status 1 Register (Index A0h)” on page 5-67	7–0
“Activity Status 2 Register (Index A1h)” on page 5-67	7–0
“PCMCIA Socket A Status Register (Index A2h)” on page 5-68	7
“CPU Status 0 Register (Index A3h)” on page 5-68	6–1
“CPU Status 1 Register (Index A4h)” on page 5-69	7 and 2–0
“NMI/SMI Control Register (Index A5h)” on page 5-70	7–0
“Miscellaneous 5 Register (Index B3h)” on page 5-81	1–0

## 5.1.3.17 To Control the Power Management State Timers

See	And program bits
“High-Speed to Low-Speed Timer Register (Index 83h)” on page 5-57	7–0
“Low-Speed to Doze Timer Register (Index 84h)” on page 5-57	7–0
“Doze to Sleep Timer Register (Index 85h)” on page 5-57	7–0
“Sleep to Suspend Timer Register (Index 86h)” on page 5-57	7–0
“Suspend to Off Timer Register (Index 87h)” on page 5-58	7–0
“PMU Control 2 Register (Index AFh)” on page 5-77	7–6

## 5.1.3.18 To Map ROM Accesses and Control ROM Cycles

See	And program bits
“MMS Memory Wait State 2 Register (Index 50h)” on page 5-30	6 and 2–0
“ROM Configuration 2 Register (Index 51h)” on page 5-32	1–0
“Command Delay Register (Index 60h)” on page 5-32	7 and 4
“MMS Memory Wait State 1 Register (Index 62h)” on page 5-35	6–5
“ROM Configuration 1 Register (Index 65h)” on page 5-40	3–0
“Miscellaneous 5 Register (Index B3h)” on page 5-81	6–4 and 2
“ROM Configuration 3 Register (Index B8h)” on page 5-84	7–0

## 5.1.3.19 To Control SMIs and Determine Status

See	And program bits
“I/O Timeout Register (Index 40h)” on page 5-24	2–0
“SMI Enable Register (Index 41h)” on page 5-25	4–0
“SMI I/O Status Register (Index 42h)” on page 5-26	3–0
“SMI Status Register (Index 43h)” on page 5-26	6–0
“Wait State Control Register (Index 63h)” on page 5-37	7
“Version Register (Index 64h)” on page 5-38	7
“Miscellaneous 2 Register (Index 6Bh)” on page 5-43	0
“NMI/SMI Enable Register (Index 82h)” on page 5-56	7–0
“PCMCIA Socket A Status Register (Index A2h)” on page 5-68	7–0
“NMI/SMI Control Register (Index A5h)” on page 5-70	7–0
“MMSB Socket Register (Index A9h)” on page 5-73	7–4
“SMI MMS Page Register (Index AAh)” on page 5-73	7–0
“PMU Control 3 Register (Index ADh)” on page 5-76	6
“Function Enable 1 Register (Index B0h)” on page 5-77	5–4
“Miscellaneous 5 Register (Index B3h)” on page 5-81	1–0

5.1.3.20 To Control the Video Display  
 For additional information on controlling the video display, see “Video Port Registers” on page 3-21.

See	And program bits
“Function Enable 2 Register (Index B1h)” on page 5-78	1

## 5.2 CONFIGURATION PORT REGISTERS

### 5.2.1 Configuration Address Register (Port 022h)

This register contains the index of the configuration index register being programmed.

	7							0
Field	Index of Configuration Index Register							
Default	0	0	0	0	0	0	0	0

### 5.2.2 Configuration Data Register (Port 023h)

This register is used to read data from, or write data to, the register pointed to by the Configuration Address register.

	7							0
Field	Data Transferred to/from Configuration Index Register							
Default	0	0	0	0	0	0	0	0



## 5.3 CONFIGURATION INDEX REGISTERS

### 5.3.1 PCMCIA I/O Window A1 Lower Byte Start Register (Index 00h)

This read/write register contains the lower byte of the I/O start address for Window 1 of Socket A. Bit 0 of this register corresponds to address line A0; bit 7 corresponds to address line A7. A system reset clears this register.

	7							0
<b>Field</b>	Socket A I/O Window 1 Start Address Bits A7–A0							
<b>Bit</b>	A7	A6	A5	A4	A3	A2	A1	A0
<b>Default</b>	0	0	0	0	0	0	0	0

### 5.3.2 PCMCIA I/O Window A1 Lower Byte End Register (Index 01h)

This read/write register contains the lower byte of the I/O end address for Window 1 of Socket A. Bit 0 of this register corresponds to address line A0; bit 7 corresponds to address line A7. A system reset clears this register.

	7							0
<b>Field</b>	Socket A I/O Window 1 End Address Bits A7–A0							
<b>Bit</b>	A7	A6	A5	A4	A3	A2	A1	A0
<b>Default</b>	0	0	0	0	0	0	0	0

### 5.3.3 PCMCIA I/O Window A1 Upper Byte Register (Index 02h)

This read/write register contains the upper byte of the I/O start and end addresses for Window 1 of Socket A. Bit 0 of this register corresponds to address line A8; bit 7 corresponds to address line A15. A system reset clears this register.

	7							0
<b>Field</b>	Socket A I/O Window 1 Start and End Address Bits A15–A8							
<b>Bit</b>	A15	A14	A13	A12	A11	A10	A9	A8
<b>Default</b>	0	0	0	0	0	0	0	0

### 5.3.4 PCMCIA I/O Window A2 Lower Byte Start Register (Index 03h)

This read/write register contains the lower byte of the I/O start address for Window 2 of Socket A. Bit 0 of this register corresponds to address line A0; bit 7 corresponds to address line A7. A system reset clears this register.

	7							0
<b>Field</b>	Socket A I/O Window 2 Start Address Bits A7–A0							
<b>Bit</b>	A7	A6	A5	A4	A3	A2	A1	A0
<b>Default</b>	0	0	0	0	0	0	0	0

### 5.3.5 PCMCIA I/O Window A2 Lower Byte End Register (Index 04h)

This read/write register contains the lower byte of the I/O end address for Window 2 of Socket A. Bit 0 of this register corresponds to address line A0; bit 7 corresponds to address line A7. A system reset clears this register.

	7							0
<b>Field</b>	Socket A I/O Window 2 End Address Bits A7–A0							
<b>Bit</b>	A7	A6	A5	A4	A3	A2	A1	A0
<b>Default</b>	0	0	0	0	0	0	0	0

### 5.3.6 PCMCIA I/O Window A2 Upper Byte Register (Index 05h)

This read/write register contains the upper byte of the I/O start and end addresses for Window 2 of Socket A. Bit 0 of this register corresponds to address line A8; bit 7 corresponds to address line A15. A system reset clears this register.

	7							0
<b>Field</b>	Socket A I/O Window 2 Start and End Address Bits A15–A8							
<b>Bit</b>	A15	A14	A13	A12	A11	A10	A9	A8
<b>Default</b>	0	0	0	0	0	0	0	0

### 5.3.7 PCMCIA I/O Card IRQ Redirection Control A Register (Index 06h)

	7								0
<b>Field</b>	ICAJNTIR								
<b>Bit</b>	ICAJNTIR3	ICAJNTIR2	ICAJNTIR1	ICAJNTIR0	INVCIAIRQ	ICAJOEN	ICAJOWIN2	ICAJOWIN1	
<b>Default</b>	0	0	0	0	0	0	0	0	

Bit	Name	R/W	Function
7	ICAJNTIR3	R/W	Socket A interrupt select bit 3
6	ICAJNTIR2	R/W	Socket A interrupt select bit 2
5	ICAJNTIR1	R/W	Socket A interrupt select bit 1
4	ICAJNTIR0	R/W	Socket A interrupt select bit 0
3	INVCIAIRQ	R/W	If set, the interrupt from the I/O card will be inverted
2	ICAJOEN	R/W	Enable Socket A as an I/O card
1	ICAJOWIN2	R/W	Enable Socket A I/O Window 2
0	ICAJOWIN1	R/W	Enable Socket A I/O Window 1

**Bits 7–4** These bits redirect the interrupt generated by the Socket A I/O card (IREQ\_A pin) to the different system interrupts. See Table 5-4 on page 5-15. Interrupts from the I/O card are redirected only if Socket A is configured as an I/O card (bit 2 = 1 and bits 7–4 are not 0).

**Bit 3** This bit selects the polarity of the IREQ line from Socket A to the internal system interrupt controller. If it is 1, the inverted polarity of the IREQ line is selected. If it is 0, the polarity of the IREQ line is not affected and is passed (as is) to the system interrupt controller.

**Bit 2** If this bit is 0, all I/O accesses within the address ranges of Socket A I/O Window 1 and I/O Window 2 do not generate  $\overline{\text{REG\_A}}$ ,  $\overline{\text{MCEL\_A}}$ , and  $\overline{\text{MCEH\_A}}$  to the PCMCIA card regardless of the state of the individual window enable (bits 1 and 0). If this bit is 1, an I/O

access within the address range of either Socket A I/O Window 1 or I/O Window 2 generates  $\overline{\text{REG\_A}}$ ,  $\overline{\text{MCEL\_A}}$ , and  $\overline{\text{MCEH\_A}}$  to the PCMCIA card, but only if the corresponding window enable (bit 0 or 1) is set. This bit should be set only after the I/O window's lower byte start, lower byte end, and upper byte addresses are programmed.

**Bit 1** If this bit is 0, an I/O access within the address range of Socket A I/O Window 2 does not generate  $\overline{\text{REG\_A}}$ ,  $\overline{\text{MCEL\_A}}$ , and  $\overline{\text{MCEH\_A}}$  to the PCMCIA card. If this bit is 1, an I/O access within the address range of Socket A I/O Window 2 generates  $\overline{\text{REG\_A}}$ ,  $\overline{\text{MCEL\_A}}$ , and  $\overline{\text{MCEH\_A}}$  to the PCMCIA card, but only if Socket A is enabled as an I/O card (bit 2 = 1).

**Bit 0** If this bit is 0, an I/O access within the address range of Socket A I/O Window 1 does not generate  $\overline{\text{REG\_A}}$ ,  $\overline{\text{MCEL\_A}}$ , and  $\overline{\text{MCEH\_A}}$  to the PCMCIA card. If this bit is 1, an I/O access within the address range of Socket A I/O Window 1 generates  $\overline{\text{REG\_A}}$ ,  $\overline{\text{MCEL\_A}}$ , and  $\overline{\text{MCEH\_A}}$  to the PCMCIA card, but only if Socket A is enabled as an I/O card (bit 2 = 1).

Table 5-4 Interrupt Redirect Bit Logic

ICAITIR					ICAITIR				
3	2	1	0	IRQ Level Select	3	2	1	0	IRQ Level Select
0	0	0	0	IRQ not selected	1	0	0	0	(Reserved)
0	0	0	1	(Reserved)	1	0	0	1	IRQ9 selected
0	0	1	0	(Reserved)	1	0	1	0	IRQ10 selected
0	0	1	1	IRQ3 selected	1	0	1	1	IRQ11 selected
0	1	0	0	IRQ4 selected	1	1	0	0	IRQ12 selected
0	1	0	1	IRQ5 selected	1	1	0	1	(Reserved)
0	1	1	0	IRQ6 selected	1	1	1	0	IRQ14 selected
0	1	1	1	IRQ7 selected	1	1	1	1	IRQ15 selected

### 5.3.8 PCMCIA VPPA Address Register (Index 07h)

This read/write register holds a programmable I/O address for controlling the VPPA pin to the card. Once this register is programmed, any write to this address changes the state of the VPPA pin to the state of data-bus bit 0.

	7							0
<b>Field</b>	System Address Bus Bits 9–2							
<b>Default</b>	0	0	0	0	0	0	0	0

**Bits 7–0** These bits correspond to bits 9–2 of the system address bus (SA9–SA2). Therefore, this register can be programmed only at a 4-byte boundary in the system I/O address space.

### 5.3.9 Resume Mask Register (Index 08h)

When set, bits 5–0 of this register mask the corresponding function.

	7							0
<b>Bit</b>	ICBRIEN	ICARIEN	RIMSK	IRQ8MSK	IRQ4MSK	IRQ3MSK	ICBL1BMSK	ICBL1AMSK
<b>Default</b>	0	0	0	0	0	0	0	0

Bit	Name	R/W	Function
7	ICBRIEN	R/W	1 = Socket B STSCH/RI input is ring in 0 = Status change
6	ICARIEN	R/W	1 = Socket A STSCH/RI input is ring in 0 = Status change
5	RIMSK	R/W	If this bit is 0 , a ring-in input can wake up the system.
4	IRQ8MSK	R/W	If this bit is 0, IRQ8 can wake up the system.
3	IRQ4MSK	R/W	If this bit is 0, IRQ4 can wake up the system.
2	IRQ3MSK	R/W	If this bit is 0, IRQ3 can wake up the system.
1	ICBL1BMSK	R/W	If this bit is 0, bit 7 (ICBRIEN) is 1, and Socket B is set as I/O by setting bit 2 of the PCMCIA I/O Card IRQ Redirection Control B register at Index 16h, then Socket B STCH/RI(BVD1B) can wake up the system.
0	ICBL1AMSK	R/W	If this bit is 0, bit 6 (ICARIEN) is 1, and Socket A is set as I/O by setting bit 2 of the PCMCIA I/O Card IRQ Redirection Control A register at Index 06h, then Socket A STCH/RI(BVD1A) can wake up the system.

**Bit 7** This bit must be 0 in order for the Socket B status-change interrupt (STSCHG\_B) to take place. If this bit is 1, STSCHG\_B interrupts are disabled, and the BVD1\_B pin can be used as a Ring Indicate wake-up activity, provided that Socket B is configured as I/O and bit 1 of this register is 0.

**Bit 6** This bit must be 0 in order for the Socket A status-change interrupt (STSCHG\_A) to take place. If this bit is 1, STSCHG\_A interrupts are disabled, and the BVD1\_A pin can be used as a Ring Indicate wake-up activity, provided that Socket A is configured as I/O and bit 0 of this register is 0.

**Bit 5** Clearing this bit allows the internal UART Ring Indicate to wake up the system.

**Bit 4** Clearing this bit allows the internal RTC to wake up the system.

**Bit 3** Clearing this bit allows a rising edge on IRQ4 (whether internally or externally generated) to wake up the system. IRQ4 should be held at 1 until the wake-up is complete.

**Bit 2** Clearing this bit allows a rising edge on IRQ3 (whether internally or externally generated) to wake up the system. IRQ3 should be held at 1 until the wake-up is complete.

**Bit 1** Clearing this bit allows a falling edge on the BVD1\_B pin to cause a PMU wake-up event. Bit 7 must be 1 to enable this. The falling edge is latched internally by the PMU.

**Bit 0** Clearing this bit allows a falling edge on the BVD\_A pin to cause a PMU wake-up event. Bit 6 must be 1 to enable this. The falling edge is latched internally.

### 5.3.10 Resume Status Register (Index 09h)

A 1 in any bit in this register indicates that the corresponding function caused the system to wake up. 00h must be written to this register to clear it.

	7							0
<b>Bit</b>	(Reserved)		PRIM_RI	IRQ8	IRQ4	IRQ3	ICB_RI	ICA_RI
<b>Default</b>	0	0	0	0	0	0	0	0

Bit	Name	R/W	Function
7–6		R/W	(Reserved)
5	PRIM_RI	R/W	System was awakened by a Ring-In from the internal UART
4	IRQ8	R/W	System was awakened by IRQ8
3	IRQ4	R/W	System was awakened by IRQ4
2	IRQ3	R/W	System was awakened by IRQ3
1	ICB_RI	R/W	System was awakened by a Ring-In from Socket B
0	ICA_RI	R/W	System was awakened by a Ring-In from Socket A

### 5.3.11 PCMCIA Data Width Register (Index 0Ah)

This register controls PCMCIA memory and I/O accesses.

	7							0
<b>Bit</b>	ICIOA16S0	ICIOA16S1	ICIOB16S0	ICIOB16S1	SCKTB16I	SCKTA16I	SCKTB16M	SCKTA16M
<b>Default</b>	0	0	0	0	0	0	0	0

Bit	Name	R/W	Function
7	ICIOA16S0	R/W	Socket A 16-bit I/O access, selector bit 1
6	ICIOA16S1	R/W	Socket A 16-bit I/O access, selector bit 0
5	ICIOB16S0	R/W	Socket B 16-bit I/O access, selector bit 1
4	ICIOB16S1	R/W	Socket B 16-bit I/O access, selector bit 0
3	SCKTB16I	R/W	Socket B can accept 16-bit I/O accesses
2	SCKTA16I	R/W	Socket A can accept 16-bit I/O accesses
1	SCKTB16M	R/W	Socket B can accept 16-bit memory accesses
0	SCKTA16M	R/W	Socket A can accept 16-bit memory accesses

**Bit 3** If this bit is 0, Socket B only accepts 8-bit I/O accesses. A 16-bit access by the system results in two 8-bit cycles to the card. If this bit is 1, all I/O accesses are performed as shown in Table 5-6 on page 5-18.

**Bit 2** If this bit is 0, Socket A only accepts 8-bit I/O accesses. A 16-bit access by the system results in two 8-bit cycles to the card. If this bit is 1, all I/O accesses are performed as shown in Table 5-5 on page 5-18.

**Bit 1** If this bit is 0, Socket B only accepts 8-bit memory accesses. A 16-bit access by the system results in two 8-bit cycles to the card. If this bit is 1, 16-bit write cycles result in one 16-bit cycle to the card.

**Bit 0** If this bit is 0, Socket A only accepts 8-bit memory accesses. A 16-bit access by the system results in two 8-bit cycles to the card. If this bit is 1, 16-bit write cycles result in one 16-bit cycle to the card.

Table 5-5 ICIOA16S0-ICIOA16S1 Bit Logic

Bit 7	Bit 6	Function
0	0	(Invalid combination)
x	1	(Invalid combination)
1	0	<p>8-bit system accesses are translated into PCMCIA cycles as follows:</p> <p>If <math>\overline{IOIS16A}</math> is Low, all 8-bit I/O accesses at even addresses are performed using <math>\overline{MCEL\_A}</math>, and all 8-bit I/O accesses at odd addresses are performed using <math>\overline{MCEH\_A}</math>. If <math>\overline{IOIS16A}</math> is High, all 8-bit I/O accesses are performed using <math>\overline{MCEL\_A}</math>.</p> <p>16-bit system accesses are translated into PCMCIA cycles as follows:</p> <p>If <math>\overline{IOIS16A}</math> is Low, a single 16-bit cycle is generated to the card using both <math>\overline{MCEL\_A}</math> and <math>\overline{MCEH\_A}</math>. Only even addresses (SA0 = 0 and SBHE = 0) can perform 16-bit I/O accesses. If <math>\overline{IOIS16A}</math> is High, two 8-bit cycles are generated to the card using <math>\overline{MCEL\_A}</math>.</p>

**Note:**

$\overline{IOIS16A}$  is a PCMCIA card-slot signal.

Table 5-6 ICIOB16S0-ICIOB16S1 Bit Logic

Bit 5	Bit 4	Function
0	0	(Invalid combination)
x	1	(Invalid combination)
1	0	<p>8-bit system accesses are translated into PCMCIA cycles as follows:</p> <p>If <math>\overline{IOIS16B}</math> is Low, all 8-bit I/O accesses at even addresses are performed using <math>\overline{MCEL\_B}</math>, and all 8-bit I/O accesses at odd addresses are performed using <math>\overline{MCEH\_B}</math>. If <math>\overline{IOIS16B}</math> is High, all 8-bit I/O accesses are performed using <math>\overline{MCEL\_B}</math>.</p> <p>16-bit system accesses are translated into PCMCIA cycles as follows:</p> <p>If <math>\overline{IOIS16B}</math> is Low, a single 16-bit cycle is generated to the card using both <math>\overline{MCEL\_B}</math> and <math>\overline{MCEH\_B}</math>. Only even addresses (SA0 = 0 and SBHE = 0) can perform 16-bit I/O accesses. If <math>\overline{IOIS16B}</math> is High, two 8-bit cycles are generated to the card using <math>\overline{MCEL\_B}</math>.</p>

**Note:**

$\overline{IOIS16B}$  is a PCMCIA card-slot signal.

### 5.3.12 Reserved Register (Index 0Bh)

This index location is reserved.

### 5.3.13 PCMCIA Socket B Status Register (Index 0Ch)

When Socket B is configured for a memory card, this register contains the status of the socket's WP, BVD1, BVD2, RDY, and card-detect pins. When Socket B is configured for an I/O card, only the card-detect pin status, bit 4, is valid. For information, see "PCMCIA Socket A Status Register (Index A2h)" on page 5-68.

	7						0	
<b>Bit</b>	(Reserved)			CDB	RDYB	BVDB1	BVDB2	WPB
<b>Default</b>	0	0	0	0	0	0	0	0

Bit	Name	R/W	Function
7-5		R	(Reserved)
4	CDB	R	Socket B Card Detect status: 0 = Card detected
3	RDYB	R	Socket B RDY status: 1 = Card ready
2	BVDB1	R	Socket B Battery Low Detect 1 status: 0 = Battery low
1	BVDB2	R	Socket B Battery Low Detect 2 status: 0 = Battery low
0	WPB	R	Socket B Write Protect status: 1 = Card write protected

### 5.3.14 PCMCIA Status Change IRQ Enable Register (Index 0Dh)

This register enables interrupts (IRQs) to be generated when status bits from the PCMCIA card change values. Interrupts generated through these pin changes can be redirected to any of the system interrupt lines using the PCMCIA Status Change IRQ Redirection register at Index 0Eh. Once the interrupt is received, the interrupt handler must clear the corresponding status bit in the PCMCIA Status Change register at Index A6h.

	7							0
<b>Bit</b>	ENBICBL2L	ENAICBL2L	ENBICBL1L	ENAICBL1L	ENBRDY	ENARDY	ENBCDCHG	ENACDCHG
<b>Default</b>	0	0	0	0	0	0	0	0

Bit	Name	R/W	Function
7	ENBICBL2L	R/W	Socket B BVDB2 transition from High to Low generates an IRQ
6	ENAICBL2L	R/W	Socket A BVDA2 transition from High to Low generates an IRQ
5	ENBICBL1L	R/W	Socket B BVDB1 transition from High to Low generates an IRQ
4	ENAICBL1L	R/W	Socket A BVDA1 transition from High to Low generates an IRQ
3	ENBRDY	R/W	Socket B RDY transition from Low to High generates an IRQ
2	ENARDY	R/W	Socket A RDY transition from Low to High generates an IRQ
1	ENBCDCHG	R/W	Socket B $\overline{CD}$ pin change generates an IRQ
0	ENACDCHG	R/W	Socket A $\overline{CD}$ pin change generates an IRQ

All the bits of this register default to 0, disabling all interrupts due to status changes. Software should set the necessary bits for proper interrupt generation.

**Bit 7** If this bit is 1 and the card is configured as memory, then a falling edge on the BVDB2 pin from Socket B generates an interrupt to the system. If the card is configured as I/O, no interrupt is generated regardless of the state of this bit.

**Bit 6** If this bit is 1 and the card is configured as memory, then a falling edge on the BVDA2 pin from Socket A generates an interrupt to the system. If the card is configured as I/O, no interrupt is generated regardless of the state of this bit.

**Bit 5** If this bit is 1, bit 7 in the Resume Mask register at Index 08h is 1, and the card is configured as memory, then a falling edge on the BVDB1 pin from Socket B generates an interrupt to the system. If this bit is 1 and the card is configured as I/O, then a falling edge on the STSCHGB pin from Socket B generates an interrupt to the system. The software must configure STSCHGB (and not RI) through bit 7 of the Resume Mask register in order for this interrupt to take place.

**Bit 4** If this bit is 1, bit 6 in the Resume Mask register at Index 08h is 1, and the card is configured as memory, then a falling edge on the BVDA1 pin from Socket A generates an interrupt to the system. If this bit is 1 and the card is configured as I/O, then a falling edge on the STSCHGA pin from Socket A generates an interrupt to the system. The software must configure STSCHGA (and not RI) through bit 6 of the Resume Mask register in order for this interrupt to take place.

**Bit 3** If this bit is 1 and the card is configured as memory, a rising edge of the RDY pin from Socket B generates an interrupt to the system. If the card is configured as I/O, no interrupt is generated regardless of the state of this bit.

**Bit 2** If this bit is 1 and the card is configured as memory, a rising edge of the RDY pin from Socket A generates an interrupt to the system. If the card is configured as I/O, no interrupt is generated regardless of the state of this bit.

**Bit 1** If this bit is 1, any changes in the  $\overline{CD}$  pin from Socket B generate an interrupt to the system.

**Bit 0** If this bit is 1, any changes in the  $\overline{CD}$  pin from Socket A generate an interrupt to the system.

### 5.3.15 PCMCIA Status Change IRQ Redirection Register (Index 0Eh)

If any interrupt source is enabled by the PCMCIA Status Change IRQ Enable register at Index 0Dh, an interrupt (selected by this register) is generated.

	7							0
<b>Field</b>					ICCHGIR			
<b>Bit</b>	(Reserved)			STSAIRQSEL	ICCHGIR3	ICCHGIR2	ICCHGIR1	ICCHGIR0
<b>Default</b>	0	0	0	0	0	0	0	0

Bit	Name	R/W	Function
7-5			(Reserved)
4	STSAIRQSEL	R/W	Socket (A + B) status change interrupt selection: 0 = IRQ 1 = SMI
3	ICCHGIR3	R/W	Socket (A + B) status change IRQ, select bit 3
2	ICCHGIR2	R/W	Socket (A + B) status change IRQ, select bit 2
1	ICCHGIR1	R/W	Socket (A + B) status change IRQ, select bit 1
0	ICCHGIR0	R/W	Socket (A + B) status change IRQ, select bit 0



**Bit 4** This bit selects the type of interrupt redirection for interrupts generated by pin changes at the card interface. If it is 0, an IRQ-type interrupt is selected, and bits 3–0 select the redirection interrupts to different IRQs. If this bit is 1, selection bits 3–0 are ignored, and an SMI-type interrupt is selected.

**Bits 3–0** These selection bits redirect interrupts generated due to pin changes at the card interface. Bit 4 must be 0 in order for these bits to take effect. A value of 0000b in these bits disables the interrupt to the system. Other values in these bits redirect the interrupts according to Table 5-7 on page 5-21. Once the interrupt is received, the interrupt handler must clear the corresponding status bit in the PCMCIA Status Change register at Index A6h in order for more interrupts to take place. Note that the corresponding status-enable bit to the interface pin (PCMCIA Status Change IRQ Enable register at Index 0Dh) must be enabled in order to generate an interrupt.

Table 5-7 IRQ Select Logic

ICCHGIR					ICCHGIR				
3	2	1	0	IRQ Level Select	3	2	1	0	IRQ Level Select
0	0	0	0	IRQ not selected	1	0	0	0	(Reserved)
0	0	0	1	(Reserved)	1	0	0	1	IRQ9 selected
0	0	1	0	(Reserved)	1	0	1	0	IRQ10 selected
0	0	1	1	IRQ3 selected	1	0	1	1	IRQ11 selected
0	1	0	0	IRQ4 selected	1	1	0	0	IRQ12 selected
0	1	0	1	IRQ5 selected	1	1	0	1	(Reserved)
0	1	1	0	(Reserved)	1	1	1	0	IRQ14 selected
0	1	1	1	IRQ7 selected	1	1	1	1	IRQ15 selected

5.3.16 Reserved Register (Index 0Fh)

This index location is reserved and must be initialized to FFh at boot time.

5.3.17 PCMCIA I/O Window B1 Lower Byte Start Register (Index 10h)

This read/write register contains the lower byte of the I/O start address for Window 1 of Socket B. Bit 0 of this register corresponds to address line A0; bit 7 corresponds to address line A7. A system reset clears this register.

	7							0
Field	Socket B I/O Window 1 Start Address Bits A7–A0							
Bit	A7	A6	A5	A4	A3	A2	A1	A0
Default	0	0	0	0	0	0	0	0

### 5.3.18 PCMCIA I/O Window B1 Lower Byte End Register (Index 11h)

This read/write register contains the lower byte of the I/O end address for Window 1 of Socket B. Bit 0 of this register corresponds to address line A0; bit 7 corresponds to address line A7. A system reset clears this register.

	7							0
<b>Field</b>	Socket B I/O Window 1 End Address Bits A7–A0							
<b>Bit</b>	A7	A6	A5	A4	A3	A2	A1	A0
<b>Default</b>	0	0	0	0	0	0	0	0

### 5.3.19 PCMCIA I/O Window B1 Upper Byte Register (Index 12h)

This read/write register contains the upper byte of the I/O start and end addresses for Window 1 of Socket B. Bit 0 of this register corresponds to address line A8; bit 7 corresponds to address line A15. A system reset clears this register.

	7							0
<b>Field</b>	Socket B I/O Window 1 Start and End Address Bits A15–A8							
<b>Bit</b>	A15	A14	A13	A12	A11	A10	A9	A8
<b>Default</b>	0	0	0	0	0	0	0	0

### 5.3.20 PCMCIA I/O Window B2 Lower Byte Start Register (Index 13h)

This read/write register contains the lower byte of the I/O start address for Window 2 of Socket B. Bit 0 of this register corresponds to address line A0; bit 7 corresponds to address line A7. A system reset clears this register.

	7							0
<b>Field</b>	Socket B I/O Window 2 Start Address Bits A7–A0							
<b>Bit</b>	A7	A6	A5	A4	A3	A2	A1	A0
<b>Default</b>	0	0	0	0	0	0	0	0

### 5.3.21 PCMCIA I/O Window B2 Lower Byte End Register (Index 14h)

This read/write register contains the lower byte of the I/O end address for Window 2 of Socket B. Bit 0 of this register corresponds to address line A0; bit 7 corresponds to address line A7. A system reset clears this register.

	7							0
<b>Field</b>	Socket B I/O Window 2 End Address Bits A7–A0							
<b>Bit</b>	A7	A6	A5	A4	A3	A2	A1	A0
<b>Default</b>	0	0	0	0	0	0	0	0

## 5.3.22 PCMCIA I/O Window B2 Upper Byte Register (Index 15h)

This read/write register contains the upper byte of the I/O start and end addresses for Window 2 of Socket B. Bit 0 of this register corresponds to address line A8; bit 7 corresponds to address line A15. A system reset clears this register.

	7							0
<b>Field</b>	Socket B I/O Window 2 Start and End Address Bits A15–A8							
<b>Bit</b>	A15	A14	A13	A12	A11	A10	A9	A8
<b>Default</b>	0	0	0	0	0	0	0	0

## 5.3.23 PCMCIA I/O Card IRQ Redirection Control B Register (Index 16h)

	7							0
<b>Field</b>	ICBINTIR							
<b>Bit</b>	ICBINTIR3	ICBINTIR2	ICBINTIR1	ICBINTIR0	INVICBIRQ	ICBIOEN	ICBIOWIN2	ICBIOWIN1
<b>Default</b>	0	0	0	0	0	0	0	0

Bit	Name	R/W	Function
7	ICBINTIR3	R/W	Socket B interrupt select bit 3
6	ICBINTIR2	R/W	Socket B interrupt select bit 2
5	ICBINTIR1	R/W	Socket B interrupt select bit 1
4	ICBINTIR0	R/W	Socket B interrupt select bit 0
3	INVICBIRQ	R/W	If set, the interrupt from the I/O card is inverted
2	ICBIOEN	R/W	Enable Socket B as an I/O card
1	ICBIOWIN2	R/W	Enable Socket B I/O Window 2
0	ICBIOWIN1	R/W	Enable Socket B I/O Window 1

**Bits 7–4** These bits redirect the interrupt generated by the Socket B I/O card to the different system interrupts. See Table 5-8 on page 5-24. Interrupts from the I/O card get redirected only if Socket B is configured as an I/O card (bit 2 = 1 and bits 7–4 are not cleared).

**Bit 3** This bit selects the polarity of the IREQ line from Socket B to the internal system interrupt controller. If it is 1, the inverted polarity of the IREQ line is selected. If it is 0, the polarity of the IREQ line is not affected and is passed (as is) to the system interrupt controller.

**Bit 2** If this bit is 0, all I/O accesses within the address ranges of Socket B I/O Window 1 and I/O Window 2 do not generate  $\overline{\text{REG\_B}}$ ,  $\overline{\text{MCEL\_B}}$ , and  $\overline{\text{MCEH\_B}}$  to the PCMCIA card regardless of the state of the individual window enable (bit 0 and bit 1). If this bit is 1, an I/O access within the address range of either Socket B I/O Window 1 or I/O Window 2 generates  $\overline{\text{REG\_B}}$ ,  $\overline{\text{MCEL\_B}}$ , and  $\overline{\text{MCEH\_B}}$  to the PCMCIA card, but only if the corresponding window enable (bit 0 or bit 1) is 1. This bit should be set only after the I/O window's lower byte start, lower byte end, and upper byte addresses are programmed.

**Bit 1** If this bit is 0, an I/O access within the address range of Socket B I/O Window 1 does not generate  $\overline{\text{REG\_B}}$ ,  $\overline{\text{MCEL\_B}}$ , and  $\overline{\text{MCEH\_B}}$  to the PCMCIA card. If this bit is 1, an I/O access within the address range of Socket B I/O Window 1 generates  $\overline{\text{REG\_B}}$ ,  $\overline{\text{MCEL\_B}}$ , and  $\overline{\text{MCEH\_B}}$  to the PCMCIA card, but only if Socket B is enabled as an I/O card (bit 2 = 1).

**Bit 0** If this bit is 0, an I/O access within the address range of Socket B I/O Window 1 does not generate  $\overline{REG\_B}$ ,  $\overline{MCEL\_B}$ , and  $\overline{MCEH\_B}$  to the PCMCIA card. If this bit is 1, an I/O access within the address range of Socket B I/O Window 1 generates  $\overline{REG\_B}$ ,  $\overline{MCEL\_B}$ , and  $\overline{MCEH\_B}$  to the PCMCIA card, but only if Socket A is enabled as an I/O card (bit 2 = 1).

Table 5-8 Interrupt Redirection Bit Logic

ICBINTIR					ICBINTIR				
3	2	1	0	IRQ Level Select	3	2	1	0	IRQ Level Select
0	0	0	0	IRQ not selected	1	0	0	0	(Reserved)
0	0	0	1	(Reserved)	1	0	0	1	IRQ9 selected
0	0	1	0	(Reserved)	1	0	1	0	IRQ10 selected
0	0	1	1	IRQ3 selected	1	0	1	1	IRQ11 selected
0	1	0	0	IRQ4 selected	1	1	0	0	IRQ12 selected
0	1	0	1	IRQ5 selected	1	1	0	1	(Reserved)
0	1	1	0	IRQ6 selected	1	1	1	0	IRQ14 selected
0	1	1	1	IRQ7 selected	1	1	1	1	IRQ15 selected

### 5.3.24 PCMCIA VPPB Address Register (Index 17h)

This read/write register holds a programmable I/O address for controlling the VPPB pin to the card. Once this register is programmed, any write to this address changes the state of the VPPB pin to the state of data-bus bit 0.

	System Address Bus Bits 9–2							
Field								
Bit	A9	A8	A7	A6	A5	A4	A3	A2
Default	0	0	0	0	0	0	0	0

**Bits 7–0** These bits correspond to bits 9–2 of the system address bus (SA9–SA2). Therefore, this register can be programmed only at a 4-byte boundary in the system I/O address space.

### 5.3.25 Reserved Registers (Indexes 18–39h)

These index locations are reserved.

### 5.3.26 I/O Timeout Register (Index 40h)

This register may be used with the SMI Enable register at Index 41h to determine if an I/O-device access generates an SMI. If a bit in this register is 0 and the corresponding bit in the SMI Enable register is 1, the next I/O access to that device causes an SMI.

In addition, writing a 0 to a bit enables an SMI to occur on the next I/O access to that device. Writing a 1 has no effect. For more information, see “Accesses to Powered-Down Device SMI” on page 1-29.

	7					0		
Bit	(Reserved)					PIOTOLTCH	FDTOLTCH	HDTOLTCH
Default	0	0	0	0	0	0	0	0

Bit	Name	R/W	Function
7–3		R/W	(Reserved)
2	PIOTOLTC H	R/W	1 = PIO access caused the SMI; PIO SMIs are masked until this bit is 0
1	FDTOLTCH	R/W	1 = Floppy disk drive access caused the SMI; floppy-disk-drive SMIs are masked until this bit is 0
0	HDTOLTCH	R/W	1 = Hard drive access caused the SMI; hard-drive SMIs are masked until this bit is 0

### 5.3.27 SMI Enable Register (Index 41h)

This register is used to control the generation of individual SMIs upon access to a specific I/O location. A value of 1 in each bit enables an SMI to be generated when the corresponding device is accessed. A value of 0 disables SMI generation for the corresponding device. For bits 2–0, an SMI is generated on the first access to the I/O address after one of the following events occurs:

- The SMI is enabled
- The timer expires
- The corresponding bit in the I/O Timeout Register at Index 40h is 0
- The PMU enters Sleep or Suspend mode

I/O accesses that occur before the timer expires cause the timer to be reloaded. The hard-drive and floppy-disk-drive timers are also reloaded when bit 0 of the Miscellaneous 4 register at Index 44h is 1 and IRQ14 (hard drive) or IRQ6 (floppy disk drive) is asserted. For more information, see “Accesses to Powered-Down Device SMI” on page 1-29.

	7					0				
Bit	(Reserved)					RTCSMIEN	KBSMIEN	ENPIO	ENFD	ENHD
Default	0	0	0	0	0	0	0	0	0	0

Bit	Name	R/W	Function
7–5		R/W	(Reserved)
4	RTCSMIEN	R/W	1 = Enable SMI generation on access to I/O address 07xh
3	KBSMIEN	R/W	1 = Enable SMI generation on 8042 access
2	ENPIO	R/W	1 = Enable SMI generation on PIO access
1	ENFD	R/W	1 = Enable SMI generation on floppy-disk-drive (3F0–3F7h) access
0	ENHD	R/W	1 = Enable SMI generation on hard-drive (1F0–1F7h) access

### 5.3.28 SMI I/O Status Register (Index 42h)

This register contains the states of the CPU bus when an SMI is generated. Software can read this register to determine the type of bus cycle executed. This register is updated when an I/O access generates an SMI that corresponds to bits 4–0 of the SMI Enable register at Index 41h or an EXTSMI pin event.

	7						0	
<b>Bit</b>	(Reserved)				IOW	IOR	BHE	A0
<b>Default</b>	0	0	0	0	0	0	0	0

Bit	Name	R/W	Function
7–4			(Reserved)
3	IOW	R	State of $\overline{IOW}$ when SMI was generated: 1 = $\overline{IOW}$ was active
2	IOR	R	State of $\overline{IOR}$ when SMI was generated: 1 = $\overline{IOR}$ was active
1	BHE	R	State of $\overline{BHE}$ when SMI was generated: 0 = $\overline{BHE}$ was active
0	A0	R	State of address bit 0 when SMI was generated

### 5.3.29 SMI Status Register (Index 43h)

This register contains the status of SMI sources. A 1 in any of these bits indicates that the corresponding device generated an SMI. The programmer may clear this register by writing to it; the data written is irrelevant. For information on the SMI enables that correspond to the bits in this register, see the SMI Enable register at Index 41h, the PCMCIA Status Change IRQ Redirection register at Index 0Eh, and the PMU Control 3 register at Index ADh.

	7							0
<b>Bit</b>	PCMCIASMI	IRQ0SMI	PMCSMI	RTCSMIEN	KBSMIEN	PIOSMI	FDSMI	HDSMI
<b>Default</b>	0	0	0	0	0	0	0	0

Bit	Name	R/W	Function
7	PCMCIASMI	R/W	1 = SMI generation on PCMCIA status change
6	IRQ0SMI	R/W	1 = IRQ0 requested an SMI
5	PMCSMI	R/W	1 = PMU mode change, $\overline{BL}$ event, or $\overline{SUS}/\overline{RES}$ event requested an SMI
4	RTCSMIEN	R/W	1 = SMI generation on access to address 07xh
3	KBSMIEN	R/W	1 = SMI generation on 8042 access
2	PIOSMI	R/W	1 = PIO requested an SMI
1	FDSMI	R/W	1 = Floppy disk drive requested an SMI
0	HDSMI	R/W	1 = Hard drive requested an SMI

### 5.3.30 Miscellaneous 4 Register (Index 44h)

This register is used to clear the hard-drive timer and control data propagation through the ÉlanSC300 microcontroller.

	7				0			
<b>Bit</b>	(Reserved)			DISDEN	(Reserved)			IRQEN
<b>Default</b>	0	0	0	0	0	0	0	0

Bit	Name	R/W	Function
7		W	(Reserved—must be 1)
6		W	(Reserved—must be 0)
5		W	(Reserved)
4	DISDEN	W	1 = Data read from DRAM via a CPU cycle is propagated only through the ÉlanSC300 microcontroller's CPU core 0 = The DRAM data is propagated to all internal ÉlanSC300 microcontroller cores
3–1		W	(Reserved—must be 0)
0	IRQEN	W	If 1, IRQ14 (hard drive) or IRQ6 (floppy disk drive) causes the hard-drive timer or floppy-disk-drive timer to reload (see Index 47h)

### 5.3.31 PIO Address Register (Index 45h)

This register is used to program PIO address bits A9–A2. The PIO base address can be anywhere in the range 000–3FCh and should be programmed on an address boundary that corresponds to the address-decode range specified in bits 6 and 7 of the PIO Timer register at Index 46h.

	7							0
<b>Field</b>	I/O Address Bits 9–2							
<b>Bit</b>	A9	A8	A7	A6	A5	A4	A3	A2
<b>Default</b>	0	0	0	0	0	0	0	0

Bit	Name	R/W	Function
7	A9	R/W	PIO address bit A9
6	A8	R/W	PIO address bit A8
5	A7	R/W	PIO address bit A7
4	A6	R/W	PIO address bit A6
3	A5	R/W	PIO address bit A5
2	A4	R/W	PIO address bit A4
1	A3	R/W	PIO address bit A3
0	A2	R/W	PIO address bit A2

### 5.3.32 PIO Timer Register (Index 46h)

This register is used to program the PIO time-out period. If no accesses are made to the I/O range specified by the PIO Address register at Index 45h in the time-out period specified here, PMC1 is driven Low. This line may be used to control power to an external device. Bits 6 and 7 of this register may be used to mask bits 2–0 of the PIO Address register. For more information, see “SMI Enable Register (Index 41h)” on page 5-25.

		7			0			
Field	Decode Range		(Reserved)		Timer Setting			
Bit	RA1	RA0	(Reserved)		PIOT3	PIOT2	PIOT1	PIOT0
Default	0	0	0	0	0	0	0	0

Bit	Name	R/W	Function
7	RA1	R/W	PIO decode range bit 1
6	RA0	R/W	PIO decode range bit 0
5-4		R/W	(Reserved)
3	PIOT3	R/W	PIO timer setting bit 3
2	PIOT2	R/W	PIO timer setting bit 2
1	PIOT1	R/W	PIO timer setting bit 1
0	PIOT0	R/W	PIO timer setting bit 0

Table 5-9 PIO Timer Setting Logic

PIOT3	PIOT2	PIOT1	PIOT0	Period
0	0	0	0	128 ms
0	0	0	1	256 ms
0	0	1	0	512 ms
0	0	1	1	1 s
0	1	0	0	2 s
0	1	0	1	4 s
0	1	1	0	8 s
0	1	1	1	16 s
1	0	0	0	32 s
1	0	0	1	64 s
1	0	1	0	128 s
1	0	1	1	256 s
1	1	0	0	512 s
1	1	0	1	1024 s
1	1	1	0	2048 s
1	1	1	1	4096 s

Table 5-10 PIO Address Range Decode Logic

RA1	RA0	PIO Address Range
0	0	4 bytes
0	1	8 bytes
1	0	16 bytes
1	1	32 bytes



### 5.3.33 Drive Timer Register (Index 47h)

This register is used to program the hard-drive and floppy-disk-drive time-out periods. If no accesses are made to the hard drive in the time-out period specified here, PMC4 is driven Low. If no accesses are made to the floppy disk drive in the time-out period specified here, PMC0 is driven Low. These lines may be used to control power to those drives. For more details concerning the use of this register, see “SMI Enable Register (Index 41h)” on page 5-25.

	7									0
<b>Field</b>	Floppy Disk Drive Timer Setting				Hard Drive Timer Setting					
<b>Bit</b>	FDT3	FDT2	FDT1	FDT0	HDT3	HDT2	HDT1	HDT0		
<b>Default</b>	0	0	0	0	0	0	0	0		

Bit	Name	R/W	Function
7	FDT3	W	Floppy-disk-drive timer setting bit 3
6	FDT2	W	Floppy-disk-drive timer setting bit 2
5	FDT1	W	Floppy-disk-drive timer setting bit 1
4	FDT0	W	Floppy-disk-drive timer setting bit 0
3	HDT3	W	Hard-drive timer setting bit 3
2	HDT2	W	Hard-drive timer setting bit 2
1	HDT1	W	Hard-drive timer setting bit 1
0	HDT0	W	Hard-drive timer setting bit 0

Table 5-11 Hard Drive and Floppy Disk Drive Timer Setting Bit Logic

HDT3 or FDT3	HDT2 or FDT2	HDT1 or FDT1	HDT0 or FDT0	Period
0	0	0	0	128 ms
0	0	0	1	256 ms
0	0	1	0	512 ms
0	0	1	1	1 s
0	1	0	0	2 s
0	1	0	1	4 s
0	1	1	0	8 s
0	1	1	1	16 s
1	0	0	0	32 s
1	0	0	1	64 s
1	0	1	0	128 s
1	0	1	1	256 s
1	1	0	0	512 s
1	1	0	1	1024 s
1	1	1	0	2048 s
1	1	1	1	4096 s

### 5.3.34 Control A Register (Index 48h)

This register contains miscellaneous control functions. Bit 1 of this register converts the UART Control register at Port 3FDh from read only (the default) to read/write, which is compatible with the 16450. Bit 2 causes the MEMR and MEMW signals to be disabled during on-board memory cycles (e.g., accesses to the local LCD controller).

	7							0
<b>Bit</b>	(Reserved)					DISCMD	LSRWCNTL	(Reserved)
<b>Default</b>	0	0	0	0	0	0	0	0

Bit	Name	R/W	Function
7-3		W	(Reserved)
2	DISCMD	W	1 = Disable $\overline{\text{MEMR}}$ and $\overline{\text{MEMW}}$ signals on the bus during on-board memory cycles
1	LSRWCNTL	W	1 = Enable UART Line Status register (Index 3FDh) for writes (retains 16450 compatibility)
0		W	(Reserved)

### 5.3.35 Reserved Registers (Indexes 49-4Fh)

These index locations are reserved.

### 5.3.36 MMS Memory Wait State 2 Register (Index 50h)

This register is used to specify the command delay and the number of wait states used with 8-bit accesses to ROM DOS or PCMCIA. The reference clock is the internal version of SYSCLK. For more information, see "Wait States and Command Delays" on page 2-23.

	7							0
<b>Field</b>							RDOSWS	
<b>Bit</b>	CARDCMDL	RDOSCMDL	CARDWSEN	CARDWS1	CARDWS0	RDOSWSEN	RDOSWS1	RDOSWS0
<b>Default</b>	0	0	0	0	0	0	0	0

Bit	Name	R/W	Function
7	CARDCMDL	W	8-bit PCMCIA memory command delay selection
6	RDOSCMDL	W	ROMDOS (ROM accessed by $\overline{\text{DOSCS}}$ ) command delay
5	CARDWSEN	W	8-bit PCMCIA memory wait state enable
4	CARDWS1	W	8-bit PCMCIA memory wait states, bit 1
3	CARDWS0	W	8-bit PCMCIA memory wait states, bit 0
2	RDOSWSEN	W	8-bit ROM DOS (ROM accessed by $\overline{\text{DOSCS}}$ ) wait state enable
1	RDOSWS1	W	8-bit ROM DOS (ROM accessed by $\overline{\text{DOSCS}}$ ) wait states, bit 1
0	RDOSWS0	W	8-bit ROM DOS (ROM accessed by $\overline{\text{DOSCS}}$ ) wait states, bit 0

Table 5-12 on page 5-31 and Table 5-13 on page 5-31 describe the wait states and command delays for 8-bit ROM DOS accesses (ROM accessed via  $\overline{\text{DOSCS}}$ ).

Table 5-12 ROM DOS Wait State Select Logic

ENFSTRDOS Bit 7 of Index B8h	RDOSWSEN	RDOSWS1	RDOSWS0	ROM Wait States
0	1	0	0	4
0	1	0	1	3
0	1	1	0	2
0	1	1	1	1
0	0	x	x	Controlled by bits 1–0 in Index 62h
1	x	x	x	Controlled by bits 6–5 in Index B8h

Table 5-13 ROM DOS Command Delay Select Logic

RDOSWSEN Bit 2 of Index 50h	RDOSCMDL Bit 6 of Index 50h	8MCD Bit 2 of Index 60h	ROM DOS Memory-Cycle Command Delay
0	x	0	1 SYSCLK cycle
0	x	1	0.5 SYSCLK cycle
1	1	x	0 delay

**Bit 7** If this bit is 1 and bit 5 is 1, no delay is selected as the command delay for PCMCIA cycles. For all other combinations of these two bits, ISA command delays (specified by bit 2 of the Command Delay register at Index 60h) are used for PCMCIA cycles.

**Bit 5** If this bit is 0 (i.e., PCMCIA memory wait states are disabled), 8-bit ISA memory-cycle wait states are chosen for PCMCIA 8-bit memory cycles. See “MMS Memory Wait State 1 Register (Index 62h)” on page 5-35. Table 5-14 on page 5-31 and Table 5-15 on page 5-32 describe the 8-bit access wait states and command delays for PCMCIA. For an explanation of this bit with respect to command delays, see the description for bit 7, above.

**Bits 4–3** These bits specify the 8-bit wait states for memory accesses to PCMCIA sockets A and B (see Table 5-14 on page 5-31).

Table 5-14 PCMCIA Wait State Select Logic

CARDWSEN	CARDWS1	CARDWS0	8-bit PCMCIA Wait States
1	0	0	4 Wait
1	0	1	3 Wait
1	1	0	2 Wait
1	1	1	1 Wait
0	x	x	Controlled by bits 1–0 in Index 62h

Table 5-15 PCMCIA Command Delay Select Logic

CARDWSEN Bit 5 of Index 50h	CARDCMD Bit 7 of Index 50h	8MCD Bit 2 of Index 60h	8-Bit PCMCIA Memory-Cycle Command Delay
0	x	0	1 SYSCLK cycle
0	x	1	0.5 SYSCLK cycle
x	0	0	1 SYSCLK cycle
x	0	1	0.5 SYSCLK cycle
1	1	x	0 delay

### 5.3.37 ROM Configuration 2 Register (Index 51h)

This register is used to enable ROM BIOS accesses and to set the ROM DOS accesses as 16-bit transfers. For more information on configuration bits for ROM BIOS accesses, see “ROM Configuration 1 Register (Index 65h)” on page 5-40.

Bit	(Reserved)						ROMDOS16	ENROMA
Default	0	0	0	0	0	0	0	

Bit	Name	R/W	Function
7-4		W	(Reserved)
3-2		W	(Reserved—must be 0)
1	ROMDOS16	W	1 = Select 16-bit ROM DOS configuration
0	ENROMA	W	1 = $\overline{ROMCS}$ is active when address is within 0A0000-0AFFFFh

### 5.3.38 Reserved Registers (Indexes 52-5Fh)

These index locations are reserved.

### 5.3.39 Command Delay Register (Index 60h)

This register is used to select different command delays for ISA-bus I/O cycles and wait states for both MMS and non-MMS memory cycles.

Field	(Reserved)						IOCD	
Bit	ROMWS1	REFWS	(Reserved)	ROMWS0	(Reserved)	8MCD	IOCD1	IOCD0
Default	0	0	0	0	0	0	0	0

Bit	Name	R/W	Function
7	ROMWS1	W	ROM BIOS non-MMS-cycle wait states, bit 1
6	REFWS	W	Refresh wait states. Must be 0 for 33-MHz operation.
5		W	(Reserved—must be 0)
4	ROMWS0	W	ROM BIOS non-MMS-cycle wait states, bit 0
3		W	(Reserved—must be 0)
2	8MCD	W	8-bit ISA memory cycle command delay
1	IOCD1	W	8-bit ISA I/O cycle command delay, bit 1
0	IOCD0	W	8-bit ISA I/O cycle command delay, bit 0

Table 5-16 8-Bit ISA I/O Access Command Delay

IOCD1	IOCD0	SYSCLK Cycles Delayed
0	0	1 (default)
0	1	2
1	x	0.5

**Note:**

Acts on addresses from 100h to 3FFh.

Table 5-17 8-bit ISA Memory Access Command Delay

8MCD	SYSCLK Cycles Delayed
0	1 (default)
1	0.5

Table 5-18 ROM BIOS Wait States

ROMWS1	ROMWS0	Number of Wait Cycles
1	0	3
0	1	2
1	1	(Invalid combination)
0	0	3 (default)

**Note:**

These bits control the number of wait cycles for ROM BIOS accesses only during non-MMS cycles. During MMS cycles to ROM BIOS, the number of wait cycles is controlled by the settings for ISA memory cycles found in the MMS Memory Wait State Select 1 register, Index 62h.

Table 5-19 Refresh Cycle Wait States

REFWS	SYSCLK Cycles in Refresh Delay
1	2
0	3 (default)

**Note:**

For 33-MHz operation, this bit must be 0.

### 5.3.40 I/O Wait State Register (Index 61h)

This register defines the number of wait states for I/O cycles to different I/O addresses. It also contains the control for forcing the CPU clock to run at 9.2 MHz from the low-speed PLL during High-Speed PLL mode.

	7						0	
Field			IOWS		HDWS		FDWS	
Bit	DMAMMS	SPEED	IOWS1	IOWS0	HDWS1	HDWS0	FDWS1	FDWS0
Default	0	0	0	0	0	0	0	0

Bit	Name	R/W	Function
7	DMAMMS	W	0 = Disable MMS during DMA 1 = Enable MMS in DMA cycle
6	SPEED	W	CPU clock speed select: 0 = Force all operations to be low speed (9.2 MHz) 1 = High speed
5	IOWS1	W	Other bus I/O wait states, bit 1
4	IOWS0	W	Other bus I/O wait states, bit 0
3	HDWS1	W	Hard-drive wait states, bit 1
2	HDWS0	W	Hard-drive wait states, bit 0
1	FDWS1	W	Floppy-disk-drive wait states, bit 1
0	FDWS0	W	Floppy-disk-drive wait states, bit 0

The number of wait states selected in this register must be greater than the bus I/O command delay specified in the Command Delay register at Index 60h.

After reset, the CPU clock runs at low speed (9.2 MHz). Write a 1 to bit 6 of this register to enable the CPU to run at high speed (the speed set by bits 4–3 of the Function Enable 2 register at Index B1h). The CPU only runs in high speed during DRAM accesses, local bus accesses, fast ROM accesses, and while idle. ISA accesses cause the CPU clock to switch to 9.2 MHz, thus saving power.

Table 5-20 Floppy Disk Drive Wait States

FDWS1	FDWS0	SYSCLK Cycles Delayed
0	0	5 (default)
0	1	4
1	0	3
1	1	2

**Note:**  
Acts on addresses from 3F0–3F7h.

Table 5-21 Hard Drive Wait States

HDWS1	HDWS0	SYSCLK Cycles Delayed
0	0	5 (default)
0	1	4
1	0	3
1	1	2

**Note:**

Acts on addresses from 1F0–1F7h.

Table 5-22 General Bus I/O Wait States

IOWS1	IOWS0	SYSCLK Cycles Delayed
0	0	5 (default)
0	1	4
1	0	3
1	1	2

**Note:**

Acts on addresses from 100h to 3FFh, except 3F0–3F7h, 1F0–1F7h, and the MMS I/O base address.

### 5.3.41 MMS Memory Wait State 1 Register (Index 62h)

This register defines the wait states for different memory accesses.

	7				0			
Field	(Reserved)	NFRDOSEN	NFROMEN	MISOUT	16BMWS		8BMWS	
Bit	(Reserved)	NFRDOSEN	NFROMEN	MISOUT	16BMWS1	16BMWS0	8BMWS1	8BMWS0
Default	0	0	0	0	0	0	0	0

Bit	Name	R/W	Function
7		W	(Reserved—must be 0)
6	NFRDOSEN	W	1 = Enable writes to devices selected by $\overline{DOSCS}$
5	NFROMEN	W	1 = Enable writes to devices selected by $\overline{ROMCS}$ pin
4	MISOUT	W	Page mode bank miss and time-out wait state select (see Table 5-26 on page 5-37): 0 = 3 wait states 1 = 5 wait states  This bit must be 1 for 33-MHz operation.
3	16BMWS1	W	Wait states bit 1 for 16-bit ISA memory cycles
2	16BMWS0	W	Wait states bit 0 for 16-bit ISA memory cycles
1	8BMWS1	W	Wait states bit 1 for 8-bit ISA memory cycles
0	8BMWS0	W	Wait states bit 0 for 8-bit ISA memory cycles

Table 5-23 on page 5-36 describes 8-bit ISA memory-cycle wait states. These bits also apply for any 8-bit cycles to ROM BIOS addressed through the MMS map where  $\overline{\text{MEMCS16}}$  is not asserted.

Table 5-23 8-Bit ISA Memory-Cycle Wait States

8BMWS1	8BMWS0	SYSCLK Cycles Delayed
0	0	(default) 5
0	1	4
1	0	3
1	1	2

Table 5-24 on page 5-36 describes 16-bit ISA memory-cycle wait states. The following applies for cycles when either  $\overline{\text{MEMCS16}}$  is asserted, or ROM DOS or PCMCIA is set up for 16-bit cycles.

Table 5-24 16-Bit ISA Memory-Cycle Wait States

16BMWS1	16BMWS0	SYSCLK Cycles Delayed
0	0	(default) 4
0	1	3
1	0	2
1	1	1



### 5.3.42 Wait State Control Register (Index 63h)

This register defines the wait states for miscellaneous accesses.

	7							0
Bit	SMMSIZE	BKMISS	FCYCWAIT1	FCYCWAIT0	INTIOWAIT	16IOWAIT	(Reserved)	SHUTD
Default	0	0	0	0	0	0	0	0

Bit	Name	R/W	Function
7	SMMSIZE	W	SMM memory range select: 0 = 16 Kbyte 1 = 64 Kbyte
6	BKMISS	W	Set DRAM bank-miss wait state in page mode
5	FCYCWAIT1	W	Set DRAM first-cycle wait state in page mode
4	FCYCWAIT0	W	Set DRAM first-cycle wait state in page mode (also SRAM wait state select). See Table 5-25 on page 5-37 and Table 5-27 on page 5-38.
3	INTIOWAIT	W	This bit controls the number of wait states for I/O addresses to internal cores: 0 = 4 wait states 1 = 2 wait states
2	16IOWAIT	W	Wait state select for 16-bit I/O cycle including PCMCIA: 0 = 4 wait cycles 1 = 3 wait cycles
1		W	(Reserved—must be 0)
0	SHUTD	W	Wait state select for shutdown cycle: 0 = 16 cycles 1 = 32 cycles

**Note:**

Bits 5 and 6 of this register must be set for 33-MHz operation. See also bit 5 of Index 65h.

**Bit 7** This bit controls the size of the MMS page that is mapped to address 60000h when an SMI occurs. Refer to the SMI MMS Page register at Index AAh.

Table 5-25 DRAM First Cycle Wait State Select Logic

PFWS Bit 5 of Index 65h	FCYCWAIT1 Bit 5	FCYCWAIT0 Bit 4	DRAM First-Cycle Wait States in Page Mode
0	x	0	(default) 1
0	x	1	2
1	0	x	2
1	1	x	(33 MHz) 3

Table 5-26 DRAM Bank Miss Wait State Select Logic

BKMISS Bit 6	MISOUT Bit 4 of Index 62h	DRAM Bank-Miss Wait States in Page Mode
x	0	(default) 3
0	1	4
1	1	(33 MHz) 5

Table 5-27 SRAM Wait State Select Logic

Configuration		Number of Wait States		SRAM Speed		
Index 63h Bit 4	Index 66h Bits 1 and 0	Read	Write	20 MHz	25 MHz	33 MHz
x	0 0	0	1	45 ns	35 ns	25 ns
0	0 1	1	1	80 ns	55 ns	35 ns
1	0 1	2	2	120 ns	100 ns	70 ns

**Note:**

Refer to Index 70h, bit 0, to select SRAM versus DRAM.

### 5.3.43 Version Register (Index 64h)

This register displays whether SMI is active. In addition, bits 2–0 of this register can be read to determine the major stepping level information (processor version). Bits 6–3 hold the minor stepping (additional revision) information. Writing to this register selects the refresh rate for DRAM and enables enhanced page mode when using 512-Kbit×8-bit DRAMs.

#### 5.3.43.1 Read Functions

Field	7	Minor Level				Major Level			0
Bit	RSMI								
Default	0	0	0	0	0	0	0	0	

Bit	Name	R/W	Function
7	RSMI	R	SMI active
6–3		R	Minor step level (revision)
2–0		R	Major step level (revision)

Table 5-28 Read Version Stepping Level Decode

Revision	Minor Level Bits 6 5 4 3	Major Level Bits 2 1 0
A	0 0 0 0	0 0 1
B0–B2	0 0 0 0	0 1 0
B3	0 0 0 1	0 1 0

## 5.3.43.2 Write Functions

	7				0			
<b>Bit</b>	(Reserved)			EPMODE	(Reserved)		REFSEL1	REFSEL0
<b>Default</b>	0	0	0	0	0	0	0	0

Bit	Name	R/W	Function
7		W	(Reserved—must be 1)
6–5		W	(Reserved—must be 0)
4	EPMODE	W	Enable enhanced page mode when the value of bits 4–2 of Index 66h is 010b and bit 7 of Index B4h is 0. Bit 0 of Index 66h must also be 1.
3–2		W	(Reserved—must be 1)
1	REFSEL1	W	Refresh interval select, bit 1
0	REFSEL0	W	Refresh interval select, bit 0

**Bits 1–0** These bits are not reset when exiting Micro Power Off mode.

The refresh interval shown in Table 5-29 on page 5-39 is selected only if bits 1–0 of the PMU Control 1 register at Index A7h are both 0.

Table 5-29 Refresh Interval Select Logic

REFSEL1	REFSEL0	Refresh Interval (cycles per second)
0	0	8192
0	1	10922
1	0	16384
1	1	32768

### 5.3.44 ROM Configuration 1 Register (Index 65h)

This register is used to configure ROM accesses. Bits 3–0 are used to set the size of the BIOS ROM that is connected to the  $\overline{\text{ROMCS}}$  signal. When each bit is 1, the  $\overline{\text{ROMCS}}$  signal goes active during accesses within the corresponding address range.  $\overline{\text{ROMCS}}$  must be disabled for regions that are shadowed. Bit 4 enables shadow RAM, and bit 7 is the shadow RAM write protect. This bit must be 1 to allow writes to the shadow RAM.

	7							0
<b>Bit</b>	DISW	ENMMSA	PFWS	SHADOW	ENROMC	ENROMD	ENROME	ENROMF
<b>Default</b>	0	0	0	0	0	0	0	0

Bit	Name	R/W	Function
7	DISW	R/W	0 = Shadow RAM write protect
6	ENMMSA	R/W	1 = MMSA enable
5	PFWS	R/W	Page mode first-cycle wait state select (see Index 63h, bit 5, and Table 5-25 on page 5-37)
4	SHADOW	R/W	1 = Shadow RAM enable
3	ENROMC	R/W	1 = $\overline{\text{ROMCS}}$ is active when address is within range 0C0000–0CFFFFh.
2	ENROMD	R/W	1 = $\overline{\text{ROMCS}}$ is active when address is within range 0D0000–0DFFFFh.
1	ENROME	R/W	1 = $\overline{\text{ROMCS}}$ is active when address is within range 0E0000–0EFFFFh.
0	ENROMF	R/W	0 = $\overline{\text{ROMCS}}$ is active when address is within range 0F0000–0FFFFFFh. This bit reads back the inverse of what was written to it. Writing a 0 enables access.

**Bit 6** This bit enables/disables all windows in MMSA. If this bit is 0, all windows are disabled. If this bit is 1, each window can be individually enabled/disabled via bit 7 of the appropriate Page register.

**Bit 5** This bit must be 1 for 33-MHz operation.

**Bit 0** Read-modify-write operations must invert bit 0 if bit 0 is not to be changed by the operation.

### 5.3.45 Memory Configuration 1 Register (Index 66h)

This register controls memory size and operating mode. Bus configuration is determined by the state of the  $\overline{DTR}$  and  $\overline{RTS}$  pins at reset. Software may read the latched state of SOUT as it was sampled at reset. SOUT may be used as a general-purpose latched input.

	7							0
Field				MS			MD	
Bit	SOUTL	CFG0	CFG1	MS2	MS1	MS0	MOD1	MOD0
Default	0	0	0	0	0	0	0	0

Bit	Name	R/W	Function
7	SOUTL	R	Latched state of SOUT at reset
6	CFG0	R	Latched state of $\overline{RTS}$ at reset
5	CFG1	R	Latched state of $\overline{DTR}$ at reset
4	MS2	R/W	Memory bank configuration, bit 2
3	MS1	R/W	Memory bank configuration, bit 1
2	MS0	R/W	Memory bank configuration, bit 0
1	MOD1	R/W	Must be 1 for DRAM. If SRAM, this bit controls wait states (see Table 5-27 on page 5-38).
0	MOD0	R/W	If configured for DRAM, enhanced page mode is enabled when this bit is set (see Index 64h, bit 4). If configured for SRAM, this bit controls wait states (see Table 5-27 on page 5-38).

Table 5-30 Bus Option Status Table

Bus Selected	CFG1	CFG0
Internal CGA	0	0
2x Clock Local Bus	1	0
Maximum ISA	x	1

Table 5-31 Memory Configuration (DRAM and SRAM)

Bit 7 of Index B4h	MS2	MS1	MS0	Total Memory	Bank 0	Bank 1
0	0	0	1	1 Mbyte	1 Mbyte	–
0	0	1	0	2 Mbyte	1 Mbyte	1 Mbyte
0	0	1	1	2 Mbyte	2 Mbyte	–
0	1	0	0	4 Mbyte	2 Mbyte	2 Mbyte
0	1	0	1	8 Mbyte	8 Mbyte	–
0	1	1	0	16 Mbyte	8 Mbyte	8 Mbyte

**Note:**

See indexes B4h and B1h for additional DRAM configurations.

### 5.3.46 MMSA Address Extension 1 Register (Index 67h)

This is the MMSA register containing address-extension bits 21–22 for pages 4–7. Bit 1 of the MMSB Control register at Index 74h must be 1 prior to writing this register.

	7				0			
Field	Page 7		Page 6		Page 5		Page 4	
Bit	E7A22	E7A21	E6A22	E6A21	E5A22	E5A21	E4A22	E4A21
Default	0	0	0	0	0	0	0	0

Bit	Name	R/W	Function
7	E7A22	R/W	MMSA page 7 address extension bit 22
6	E7A21	R/W	MMSA page 7 address extension bit 21
5	E6A22	R/W	MMSA page 6 address extension bit 22
4	E6A21	R/W	MMSA page 6 address extension bit 21
3	E5A22	R/W	MMSA page 5 address extension bit 22
2	E5A21	R/W	MMSA page 5 address extension bit 21
1	E4A22	R/W	MMSA page 4 address extension bit 22
0	E4A21	R/W	MMSA page 4 address extension bit 21

### 5.3.47 Shadow RAM Enable 1 Register (Index 68h)

This register controls the shadow-RAM mapping range. Mapping is disabled by default.

	7				0			
Bit	SDCF	SD8B	SD47	SD03	SCCF	SC8B	SC47	SC03
Default	0	0	0	0	0	0	0	0

Bit	Name	R/W	Function
7	SDCF	R/W	Enable shadow RAM at range 0DC000–0DFFFFh
6	SD8B	R/W	Enable shadow RAM at range 0D8000–0DBFFFh
5	SD47	R/W	Enable shadow RAM at range 0D4000–0D7FFFh
4	SD03	R/W	Enable shadow RAM at range 0D0000–0D3FFFh
3	SCCF	R/W	Enable shadow RAM at range 0CC000–0CFFFFh
2	SC8B	R/W	Enable shadow RAM at range 0C8000–0CBFFFh
1	SC47	R/W	Enable shadow RAM at range 0C4000–0C7FFFh
0	SC03	R/W	Enable shadow RAM at range 0C0000–0C3FFFh

### 5.3.48 Shadow RAM Enable 2 Register (Index 69h)

This register controls the shadow-RAM mapping range. Mapping is disabled by default.

	7							0
<b>Bit</b>	SFCF	SF8B	SF47	SF03	SECF	SE8B	SE47	SE03
<b>Default</b>	0	0	0	0	0	0	0	0

Bit	Name	R/W	Function
7	SFCF	R/W	Enable shadow RAM at range 0FC000–0FFFFFh
6	SF8B	R/W	Enable shadow RAM at range 0F8000–0FBFFFh
5	SF47	R/W	Enable shadow RAM at range 0F4000–0F4000h
4	SF03	R/W	Enable shadow RAM at range 0F0000–0F3FFFh
3	SECF	R/W	Enable shadow RAM at range 0EC000–0EFFFFh
2	SE8B	R/W	Enable shadow RAM at range 0E8000–0EBFFFh
1	SE47	R/W	Enable shadow RAM at range 0E4000–0E7FFFh
0	SE03	R/W	Enable shadow RAM at range 0E0000–0E3FFFh

### 5.3.49 Reserved Register (Index 6Ah)

This register is reserved and must be 00h.

### 5.3.50 Miscellaneous 2 Register (Index 6Bh)

This register contains several miscellaneous control bits.

	7							0
<b>Bit</b>	(Reserved)					CPU_IDLE	(Reserved)	A20SMI
<b>Default</b>	0	0	0	0	0	0	0	0

Bit	Name	R/W	Function
7		W	(Reserved—must be 0)
6–5		R	(Reserved)
4		R/W	(Reserved—must be 1)
3		R	(Reserved)
2	CPU_IDLE	W	Selects whether CPU idle cycles are dynamically switched to use the high-speed clock frequency: 0 = High speed 1 = Low speed (9.2 MHz)  This bit must be 0 for 33-MHz and 25-MHz operation.
1			(Reserved)
0	A20SMI	W	Controls gate A20 during SMI access: 1 = A20 propagates 0 = A20 Low

### 5.3.51 MMS Address Extension 1 Register (Index 6Ch)

This is the MMS register that contains address-extension bit 23 for all pages.

	7							0
Bit	E7A23	E6A23	E5A23	E4A23	E3A23	E2A23	E1A23	E0A23
Default	0	0	0	0	0	0	0	0

Bit	Name	R/W	Function
7	E7A23	R/W	MMSA page 7 address extension bit 23
6	E6A23	R/W	MMSA page 6 address extension bit 23
5	E5A23	R/W	MMSA page 5 address extension bit 23
4	E4A23	R/W	MMSA page 4 address extension bit 23
3	E3A23	R/W	MMSA/MMSB page 3 address extension bit 23
2	E2A23	R/W	MMSA/MMSB page 2 address extension bit 23
1	E1A23	R/W	MMSA/MMSB page 1 address extension bit 23
0	E0A23	R/W	MMSA/MMSB page 0 address extension bit 23

**Bits 3–0** Provide mapped address-extension bit 23 for both MMSA and MMSB (pages 0–3). Before programming this register, software must select the region to program (i.e., MMSA or MMSB). This selection is performed via bit 1 of the MMSB Control register at Index 74h.

### 5.3.52 MMS Address Register (Index 6Dh)

This register selects the base I/O addresses and page addresses.

	7							0
Field	Base Address				Page Address			
Bit	EMBA3	EMBA2	EMBA1	EMBA0	EMIO3	EMIO2	EMIO1	EMIO0
Default	0	0	0	0	0	0	0	0

Bit	Name	R/W	Function
7	EMBA3	R/W	MMSA base address, bit 3
6	EMBA2	R/W	MMSA base address, bit 2
5	EMBA1	R/W	MMSA base address, bit 1
4	EMBA0	R/W	MMSA base address, bit 0
3	EMIO3	R/W	MMSA/B page register(s) I/O address, bit 3
2	EMIO2	R/W	MMSA/B page register(s) I/O address, bit 2
1	EMIO1	R/W	MMSA/B page register(s) I/O address, bit 1
0	EMIO0	R/W	MMSA/B page register(s) I/O address, bit 0

**Bits 3–0** Provide selection of page-register I/O addresses for both MMSA (pages 0–7) and MMSB (pages 0–3). These page registers get mapped at different I/O locations as listed in Table 5-32 on page 5-45. Once the I/O address spaces for these registers are programmed, writes to these address spaces store mapped address-extension bits 20–14 for the windows in MMSA and MMSB as indicated in Table 5-33 on page 5-45.



Table 5-32 MMSA/B Page Register I/O Addresses

Bit				Page Register I/O Address							
3	2	1	0	Page 0	Page 1	Page 2	Page 3	Page 4	Page 5	Page 6	Page 7
0	0	0	0	208h	2208h	4208h	6208h	8208h	A208h	C208h	E208h
0	0	0	1	218h	2218h	4218h	6218h	8218h	A218h	C218h	E218h
0	1	0	1	258h	2258h	4258h	6258h	8258h	A258h	C258h	E258h
0	1	1	0	268h	2268h	4268h	6268h	8268h	A268h	C268h	E268h
1	0	1	0	2A8h	22A8h	42A8h	62A8h	82A8h	A2A8h	C2A8h	E2A8h
1	0	1	1	2B8h	22B8h	42B8h	62B8h	82B8h	A2B8h	C2B8h	E2B8h
1	1	1	0	2E8h	22E8h	42E8h	62E8h	82E8h	A2E8h	C2E8h	E2E8h

Table 5-33 Page Register Contents Description

Bit	Name	R/W	Function
7	PAGEEN	R/W	0 = Page disable 1 = Page enable
6	EA20	R/W	MMSA/B translate address bit A20
5	EA19	R/W	MMSA/B translate address bit A19
4	EA18	R/W	MMSA/B translate address bit A18
3	EA17	R/W	MMSA/B translate address bit A17
2	EA16	R/W	MMSA/B translate address bit A16
1	EA15	R/W	MMSA/B translate address bit A15
0	EA14	R/W	MMSA/B translate address bit A14

**Bits 7–4** Provide selection of starting addresses of memory windows in MMSA (pages 0–7). These windows get mapped at different system-memory address locations as listed in Table 5-34 on page 5-46. Once the system-memory address spaces for these windows are programmed (i.e., the base address is selected) and the page register I/O addresses are selected, the software can program the page registers. Software must also program the other address extension registers (MMS Address Extension 2 register at Index 6Eh, MMSA Address Extension 1 register at Index 67h, and MMS Address Extension 1 register at Index 6Ch) before a page in the MMSA or the MMSB is enabled. See the ROM Configuration 1 register at Index 65h and the MMSB Control Register at Index 74h to enable the MMSA and the MMSB. Note that the MMSB base address is fixed at 0A0000h and is not under software control.

Table 5-34 MMSA Base Addresses

Bit				Base Address							
7	6	5	4	Page 0	Page 1	Page 2	Page 3	Page 4	Page 5	Page 6	Page 7
0	0	0	0	C0000h	C4000h	C8000h	CC000h	D0000h	D4000h	D8000h	DC000h
0	0	0	1	C4000h	C8000h	CC000h	D0000h	D4000h	D8000h	DC000h	E0000h
0	0	1	0	C8000h	CC000h	D0000h	D4000h	D8000h	DC000h	E0000h	E4000h
0	0	1	1	CC000h	D0000h	D4000h	D8000h	DC000h	E0000h	E4000h	E8000h
0	1	0	0	D0000h	D4000h	D8000h	DC000h	E0000h	E4000h	E8000h	EC000h
0	1	0	1	D4000h	D8000h	DC000h	E0000h	E4000h	E8000h	EC8000h	F0000h

### 5.3.53 MMS Address Extension 2 Register (Index 6Eh)

This is the MMS register containing address-extension bits 21–22 for pages 0–3.

	7						0	
Field	Page 3		Page 2		Page 1		Page 0	
Bit	E3A22	E3A21	E2A22	E2A21	E1A22	E1A21	E0A22	E0A21
Default	0	0	0	0	0	0	0	0

Bit	Name	R/W	Function
7	E3A22	R/W	MMSA/MMSB page 3 address extension bit 22
6	E3A21	R/W	MMSA/MMSB page 3 address extension bit 21
5	E2A22	R/W	MMSA/MMSB page 2 address extension bit 22
4	E2A21	R/W	MMSA/MMSB page 2 address extension bit 21
3	E1A22	R/W	MMSA/MMSB page 1 address extension bit 22
2	E1A21	R/W	MMSA/MMSB page 1 address extension bit 21
1	E0A22	R/W	MMSA/MMSB page 0 address extension bit 22
0	E0A21	R/W	MMSA/MMSB page 0 address extension bit 21

**Bits 7–0** Provide mapped address-extension bits 22–21 for both the MMSA and the MMSB (pages 0–3).

Before programming this register, software must select the region to program (i.e., MMSA or MMSB). This selection is performed via bit 1 of the MMSB Control register at Index 74h.

### 5.3.54 Miscellaneous 1 Register (Index 6Fh)

This register is used to control ISA and MCU functions.

	7	MMSZ				0			
Field									
Bit	MMSZ3	MMSZ2	MMSZ1	MMSZ0	MRDLY	DMWS	RESCPU	GATEA20	
Default	0	0	0	0	0	0	0	0	

Bit	Name	R/W	Function
7	MMSZ3	R/W	MMS memory range, bit 3
6	MMSZ2	R/W	MMS memory range, bit 2
5	MMSZ1	R/W	MMS memory range, bit 1
4	MMSZ0	R/W	MMS memory range, bit 0
3	MRDLY	R/W	MEMR delay: 1 = Delay MEMR by 1 DMACK cycle 0 = No delay for MEMR
2	DMWS	R/W	DMA wait states: 1 = 2 wait cycles 0 = 1 wait cycle
1	RESCPU	R/W	CPU reset: A Low-to-High transition in this bit automatically resets the CPU. The reset lasts for 16 PROCLK cycles.
0	GATEA20	R/W	A20 gate control: 1 = A20 is CPUA20 0 = A20 is 0 if bit 0 of Port 92h and the A20GATE pin are also 0

Table 5-35 MMS Memory Range Select Logic

Bit				MMS Memory Range	Bit				MMS Memory Range
7	6	5	4		7	6	5	4	
0	0	0	0	No MMS	1	0	0	0	8 Mbyte
0	0	0	1	1 Mbyte	1	0	0	1	9 Mbyte
0	0	1	0	2 Mbyte	1	0	1	0	10 Mbyte
0	0	1	1	3 Mbyte	1	0	1	1	11 Mbyte
0	1	0	0	4 Mbyte	1	1	0	0	12 Mbyte
0	1	0	1	5 Mbyte	1	1	0	1	13 Mbyte
0	1	1	0	6 Mbyte	1	1	1	0	14 Mbyte
0	1	1	1	7 Mbyte	1	1	1	1	15 Mbyte

MMS memory range provides a method for disabling on-board memory accesses when directly accessing memory (non-MMS cycles). The programmed MMS memory range is subtracted from the amount of on-board memory configured by bits 4–2 of the Memory Configuration 1 register at Index 66h or bits 7–6 of the Function Enable 2 register at Index B1h. If the programmed MMS memory range is equal to or greater than the configured amount of on-board memory, then on-board memory is disabled and all memory cycles are transferred on the ISA bus.

For example, assume that 8 Mbyte of DRAM have been configured as on-board main memory. If the user desires to have 10 Mbyte of linearly addressed DOS ROM, then bits 7–4 of the Miscellaneous 1 register at Index 6Fh should be programmed to 0010b. This allows all accesses in the range 000000–5FFFFFFh to transfer to DRAM, and all accesses in the range 600000–FFFFFFh to be ISA bus transfers. For more information, see “ROM DOS Memory” on page 2-9.

### 5.3.55 Miscellaneous 6 Register (Index 70h)

This register is used to control MCU and PMU functions.

	7							0
<b>Bit</b>	(Reserved)	PGP0DIR	SACIN	(Reserved)				MTS
<b>Default</b>	0	0	0	0	0	0	0	0

Bit	Name	R/W	Function
7		R/W	(Reserved)
6	PGP0DIR	R/W	PGP0 pin direction: 0 = Input 1 = Output
5	SACIN	R/W	Software ACIN input: 1 = PMU behaves as if ACIN pin was asserted
4–1		R/W	(Reserved)
0	MTS	R/W	Memory type: 0 = DRAM 1 = SRAM

**Bit 5** Setting this bit is equivalent to asserting the ACIN pin, except that the  $\overline{BL4}$ – $\overline{BL0}$  pins are not gated with this bit. Therefore, the  $\overline{BL4}$ – $\overline{BL0}$  pins can still change the state of the PMU.

### 5.3.56 MMSA Device 1 Register (Index 71h)

This register selects the peripheral device that each MMSA page controls.

	7						0	
Field	Page 3 Device		Page 2 Device		Page 1 Device		Page 0 Device	
Bit	EMDP31	EMDP30	EMDP21	EMDP20	EMDP11	EMDP10	EMDP01	EMDP00
Default	0	0	0	0	0	0	0	0

Bit	Name	R/W	Function
7	EMDP31	R/W	MMSA page 3 device type, bit 1
6	EMDP30	R/W	MMSA page 3 device type, bit 0
5	EMDP21	R/W	MMSA page 2 device type, bit 1
4	EMDP20	R/W	MMSA page 2 device type, bit 0
3	EMDP11	R/W	MMSA page 1 device type, bit 1
2	EMDP10	R/W	MMSA page 1 device type, bit 0
1	EMDP01	R/W	MMSA page 0 device type, bit 1
0	EMDP00	R/W	MMSA page 0 device type, bit 0

**Note:**

PCMCIA CARD\_A or CARD\_B is selected by the MMSA Socket register at Index A8h.

Page 3 Device Select		Page 2 Device Select	
Bit 7 6	Selected Device	Bit 5 4	Selected Device
0 0	MMS ROM DOS	0 0	MMS ROM DOS
0 1	MMS on-board main memory	0 1	MMS on-board main memory
1 0	PCMCIA	1 0	PCMCIA
1 1	MMS BIOS	1 1	MMS BIOS

Page 1 Device Select		Page 0 Device Select	
Bit 3 2	Selected Device	Bit 1 0	Selected Device
0 0	MMS ROM DOS	0 0	MMS ROM DOS
0 1	MMS on-board main memory	0 1	MMS on-board main memory
1 0	PCMCIA	1 0	PCMCIA
1 1	MMS BIOS	1 1	MMS BIOS

### 5.3.57 MMSA Device 2 Register (Index 72h)

Bits of this register select the peripheral device that each MMSA page controls.

	7						0	
Field	Page 7 Device		Page 6 Device		Page 5 Device		Page 4 Device	
Bit	EMDP71	EMDP70	EMDP61	EMDP60	EMDP51	EMDP50	EMDP41	EMDP40
Default	0	0	0	0	0	0	0	0

Bit	Name	R/W	Function
7	EMDP71	R/W	MMSA page 7 device type, bit 1
6	EMDP70	R/W	MMSA page 7 device type, bit 0
5	EMDP61	R/W	MMSA page 6 device type, bit 1
4	EMDP60	R/W	MMSA page 6 device type, bit 0
3	EMDP51	R/W	MMSA page 5 device type, bit 1
2	EMDP50	R/W	MMSA page 5 device type, bit 0
1	EMDP41	R/W	MMSA page 4 device type, bit 1
0	EMDP40	R/W	MMSA page 4 device type, bit 0

**Note:**

PCMCIA CARD\_A or CARD\_B is selected by the MMSA Socket register at Index A8h.

Page 7 Device Select		Page 6 Device Select	
Bit 7 6	Selected Device	Bit 5 4	Selected Device
0 0	MMS ROM DOS	0 0	MMS ROM DOS
0 1	MMS on-board main memory	0 1	MMS on-board main memory
1 0	PCMCIA	1 0	PCMCIA
1 1	MMS BIOS	1 1	MMS BIOS

Page 5 Device Select		Page 4 Device Select	
Bit 3 2	Selected Device	Bit 1 0	Selected Device
0 0	MMS ROM DOS	0 0	MMS ROM DOS
0 1	MMS on-board main memory	0 1	MMS on-board main memory
1 0	PCMCIA	1 0	PCMCIA
1 1	MMS BIOS	1 1	MMS BIOS

### 5.3.58 MMSB Device Register (Index 73h)

Bits of this register select the peripheral device that each MMSB page controls.

	7						0	
Field	Page 3 Device		Page 2 Device		Page 1 Device		Page 0 Device	
Bit	EMDP31	EMDP30	EMDP21	EMDP20	EMDP11	EMDP10	EMDP01	EMDP00
Default	0	0	0	0	0	0	0	0

Bit	Name	R/W	Function
7	EMDP31	R/W	MMSB page 3 device type, bit 1
6	EMDP30	R/W	MMSB page 3 device type, bit 0
5	EMDP21	R/W	MMSB page 2 device type, bit 1
4	EMDP20	R/W	MMSB page 2 device type, bit 0
3	EMDP11	R/W	MMSB page 1 device type, bit 1
2	EMDP10	R/W	MMSB page 1 device type, bit 0
1	EMDP01	R/W	MMSB page 0 device type, bit 1
0	EMDP00	R/W	MMSB page 0 device type, bit 0

**Note:**

PCMCIA CARD\_A or CARD\_B is selected by the MMSB Socket register at Index A9h.

Page 3 Device Select		Page 2 Device Select	
Bit 7 6	Selected Device	Bit 5 4	Selected Device
0 0	MMS ROM DOS	0 0	MMS ROM DOS
0 1	MMS on-board main memory	0 1	MMS on-board main memory
1 0	PCMCIA	1 0	PCMCIA
1 1	MMS BIOS	1 1	MMS BIOS

Page 1 Device Select		Page 0 Device Select	
Bit 3 2	Selected Device	Bit 1 0	Selected Device
0 0	MMS ROM DOS	0 0	MMS ROM DOS
0 1	MMS on-board main memory	0 1	MMS on-board main memory
1 0	PCMCIA	1 0	PCMCIA
1 1	MMS BIOS	1 1	MMS BIOS

### 5.3.59 MMSB Control Register (Index 74h)

This register controls MCU and PMU functions.

	7						0	
Bit	NENLB4	NENLB2	(Reserved)		ENPMCIRQ0	PGP1DIR	MMSABSEL	ENMMSB
Default	0	0	0	0	0	0	0	0

Bit	Name	R/W	Function
7	NENLB4	R/W	1 = $\overline{BL4}$ going active does not cause the PMU to transition to Suspend mode.
6	NENLB2	R/W	1 = $\overline{BL2}$ going active does not cause the PMU to transition to Sleep mode.
5		R/W	(Reserved—must be 0)
4		R/W	(Reserved—must be 0)
3	ENPMCIRQ0	R/W	Enable IRQ0 active in Doze mode: 1 = Enabled.  If this bit is 1 and bit 0 of the PMU Control 2 register at Index AFh is 0, the CPUCLK signal is active while IRQ0 is High in Doze mode. If this bit is 1 and bit 0 of the PMU Control 2 register is 1, the CPUCLK signal remains active for an additional 64 refresh cycles after IRQ0 is deasserted.
2	PGP1DIR	R/W	PGP1 pin direction: 0 = Input 1 = Output
1	MMSABSEL	R/W	MMSA and MMSB select bit: 0 = MMSB 1 = MMSA
0	ENMMSB	R/W	Enable MMSB: 1 = Enabled

**Bit 7** If bit 7 = 0 and ACIN = 0, the PMU transitions to Suspend mode when  $\overline{BL4} = 0$ .

**Bit 6** If bit 6 = 0 and ACIN = 0, the PMU transitions to Sleep mode when  $\overline{BL2} = 0$ . This bit has no effect on whether an SMI for  $\overline{BL2}$  is generated. It also does not have an effect on the status read at the CPU Status 0 register at Index A3h.

**Bit 1** Since the MMSA and the MMSB use the same I/O address for the page registers and address extension registers for pages 0–3, this bit selects either the MMSA or the MMSB for programming. In other words, this bit directs the I/O address to either the MMSA or the MMSB. If this bit is 1, an I/O cycle accesses the MMSA; otherwise, an I/O cycle accesses the MMSB.

**Bit 0** This bit enables/disables all windows in the MMSB. If this bit is 0, all windows are disabled. If this bit is 1, each window can be individually enabled/disabled via bit 7 of the appropriate page register.



### 5.3.60 Activity Mask 1 Register (Index 75h)

This register is used in conjunction with the Activity Mask 2 register at Index 76h and the Resume Mask register at Index 08h to enable which activities are detected by the PMU. Each of these bits masks out the corresponding activity. A 1 means the activity is masked; a 0 means it is counted. For information on status and enabling ACIN activity, see “Activity Status 1 Register (Index A0h)” on page 5-67 and “PMU Control 3 Register (Index ADh)” on page 5-76.

	7							0
<b>Bit</b>	INT	ACIN	MMS	KB	DRQ3	DRQ2	DRQ1	DRQ7–DRQ5
<b>Default</b>	0	0	0	0	0	0	0	0

Bit	Name	R/W	Function
7	INT	R/W	Interrupt recognition: 1 = Disabled 0 = Enabled
6	ACIN	R/W	AC adapter input or Low-to-High transition of bit 5 of the Miscellaneous 6 register at Index 70h recognition: 1 = Disabled 0 = Enabled
5	MMS	R/W	MMS cycle activity recognition: 1 = Disabled 0 = Enabled
4	KB	R/W	Keyboard interrupt (IRQ1) activity recognition: 1 = Disabled 0 = Enabled
3	DRQ3	R/W	DRQ3 activity recognition: 1 = Disabled 0 = Enabled
2	DRQ2	R/W	DRQ2 activity recognition: 1 = Disabled 0 = Enabled
1	DRQ1	R/W	DRQ1 activity recognition: 1 = Disabled 0 = Enabled
0	DRQ7–DRQ5	R/W	DRQ7–DRQ5 activity recognition: 1 = Disabled 0 = Enabled

**Note:**

*Activities are not detected during the execution of SMLs or NMIs.*

**Bit 7** INT means that all interrupts from IRQ2 to IRQ15 can serve as PMU activity that cause a PMU transition from either Low-Speed PLL or Doze mode to High-Speed PLL mode. This bit does not allow the above stated IRQ levels to wake up the processor from Sleep, Suspend, or Off mode.

IRQ3, IRQ4, and IRQ8 can be programmed individually by the Resume Mask register at Index 08h to act as wake-up events. Unmasking these events allows their occurrence to wake up the system from Sleep, Suspend, or Off mode into High-Speed PLL mode.

**Bit 6** Bit 4 of the PMU Control 3 register at Index ADh must also be set to permit ACIN going active to count as activity.

**Bits 6 and 3–0** These activities also wake up the system from Sleep, Suspend, or Off mode into High-Speed PLL mode.

**Bit 4** Unlike other IRQs, IRQ1 activity cannot be masked by the 8259 PIC.

### 5.3.61 Activity Mask 2 Register (Index 76h)

This register is used in conjunction with the Activity Mask 1 register at Index 75h and the Resume Mask register at Index 08h to enable which activities are detected by the PMU. Each of these bits masks the corresponding activity when set. For more information on activity status reporting, see “Activity Status 2 Register (Index A1h)” on page 5-67.

	7							0
<b>Bit</b>	PMW	VD	PIO1	PIO0	COM	HD	FD	LPT
<b>Default</b>	0	0	0	0	0	0	0	0

Bit	Name	R/W	Function
7	PMW	R/W	Memory address range (defined in the Memory Write Activity Lower and Upper Boundary registers at indexes 9Ah and 9Bh) recognition: 1 = Disabled 0 = Enabled
6	VD	R/W	Video memory write recognition: 1 = Disabled 0 = Enabled
5	PIO1	R/W	I/O address range (defined in the I/O Activity Address 1 register at Index 8Dh) recognition: 1 = Disabled 0 = Enabled
4	PIO0	R/W	I/O address range (defined in the I/O Activity Address 0 register at Index 8Ch) recognition: 1 = Disabled 0 = Enabled
3	COM	R/W	COM1–COM2 read/write recognition: 1 = Disabled 0 = Enabled
2	HD	R/W	Hard drive read/write recognition: 1 = Disabled 0 = Enabled
1	FD	R/W	Floppy disk drive read/write recognition: 1 = Disabled 0 = Enabled
0	LPT	R/W	LPT1–LPT3 read/write recognition: 1 = Disabled 0 = Enabled

**Note:**

*Activities are not detected during SMI or NMI execution.*

### 5.3.62 Control B Register (Index 77h)

This register controls various general functions.

	7							0
Bit	UART_IR4	UART_IR3	UART_IOP	UART_EN	AUTLOW	(Reserved)		
Default	0	0	0	0	0	0	0	0

Bit	Name	R/W	Function
7	UART_IR4	R/W	Set internal UART IRQ to IRQ4
6	UART_IR3	R/W	Set internal UART IRQ to IRQ3
5	UART_IOP	R/W	0 = Set internal UART I/O address to 3F8–3FFh 1 = Set internal UART I/O address to 2F8–2FFh
4	UART_EN	R/W	1 = Enable internal UART
3	AUTLOW	R/W	1 = Enable Auto Low-Speed
2–0		R/W	(Reserved)

**Bit 6** If the ÉlanSC300 microcontroller is in Local Bus or Internal Video modes, program the PIRQ Configuration register at Index B2h so it does not conflict with the IRQ selection for the internal UART.

**Bit 3** This bit is only useful if bit 6 of the I/O Wait State register at Index 61h is 1, which puts the CPU clock into High-Speed PLL mode. Otherwise, the CPU clock is always operating at the low-speed PLL frequency. This function is not dependent on any activity (see “Auto Low-Speed Control Register (Index 9Fh)” on page 5-66).

### 5.3.63 Reserved Registers (Indexes 78–7Fh)

These index locations are reserved.

### 5.3.64 Power Control 1 Register (Index 80h)

This register controls the PMC2 output pin in High-Speed PLL mode, Low-Speed PLL mode, and Doze mode. It also enables/disables the low-speed PLL and video PLL in Doze mode.

	7							0
Bit	0CLK_DOZ	DZ2	(Reserved)			FO2	(Reserved)	
Default	0	0	0	0	0	0	0	0

Bit	Name	R/W	Function
7	0CLK_DOZ	R/W	1 = Low-speed PLL and video PLL are shut down in Doze mode
6	DZ2	R/W	1 = State of PMC2 pin in Doze mode
5–4		R/W	(Reserved)
3		R/W	(Reserved—must be 0)
2	FO2	R/W	1 = State of PMC2 pin in High-Speed PLL and Low-Speed PLL modes
1–0		R/W	(Reserved)

**Note:**

The state of PMC2 after power-on is Low. When the bit is 0, the corresponding PMC output is Low. For details, see Chapter 1, “Power Management.”

### 5.3.65 Power Control 2 Register (Index 81h)

This register activates the PMC2 output pin in Sleep, Suspend, and Off modes. It also enables/disables the low-speed PLL and video PLL in Sleep, Suspend, and Off modes.

	7							0
<b>Bit</b>	OCLK_SUS	SU2			OCLK_SLP	SP2		
<b>Default</b>	0	0	0	0	0	0	0	0

Bit	Name	R/W	Function
7	OCLK_SUS	R/W	1 = Low-speed PLL and Video PLL are shut down in Suspend and Off modes
6	SU2	R/W	1 = State of PMC2 pin in Suspend and Off modes
5–4		R/W	(Reserved)
3	OCLK_SLP	R/W	1 = Low-speed PLL and video PLL are shut down in Sleep mode
2	SP2	R/W	1 = State of PMC2 pin in Sleep mode
1–0		R/W	(Reserved)

**Note:**

The state of PMC2 after power-on is Low. When the bit is 0, the corresponding PMC output is Low. For details, see Chapter 1, “Power Management.”

### 5.3.66 NMI/SMI Enable Register (Index 82h)

This register is used to enable the generation of NMIs or SMIs during certain conditions, such as mode changes or battery-low conditions. By default, NMIs and SMIs are disabled. The choice of SMIs or NMIs is selected by bit 7 of the MMSB Socket register at Index A9h.

	7							0
<b>Bit</b>	BL3	BL2	BL1	SUS	SLP	DZ	ON	RESU
<b>Default</b>	0	0	0	0	0	0	0	0

Bit	Name	R/W	Function
7	BL3	R/W	1 = Battery low warning 3 generates an NMI or SMI
6	BL2	R/W	1 = Battery low warning 2 generates an NMI or SMI
5	BL1	R/W	1 = Battery low warning 1 generates an NMI or SMI
4	SUS	R/W	1 = PMU generates NMI or SMI before entering Off mode from Suspend mode
3	SLP	R/W	1 = PMU generates NMI or SMI before entering Suspend mode from Sleep mode
2	DZ	R/W	1 = PMU generates NMI or SMI before entering Sleep mode from Doze mode
1	ON	R/W	1 = PMU generates NMI or SMI before entering Doze mode from Low-Speed PLL mode
0	RESU	R/W	1 = $\overline{\text{SUS}}/\overline{\text{RES}}$ pin will generate NMI or SMI

**Note:**

Bit 7 of AT-compatible port 70h must be 0 for NMIs to occur.

### 5.3.67 High-Speed to Low-Speed Timer Register (Index 83h)

This read/write register is used to program the time-out period from High-Speed PLL mode to Low-Speed PLL mode. The minimum period is 1/512 s. The maximum period is 15.94 s. A value of 00h disables the PMU timer. If any activities are detected during the timer counting period, the PMU timer is reset to 00h automatically.

	7							0
<b>Field</b>	Time-Out Period in Multiples of 1/512 s or 1/16 s							
<b>Default</b>	0	0	0	0	0	0	0	0

**Note:** The timer granularity can be changed to 1/16 s by setting bit 6 of the PMU Control 2 register at Index AFh.

### 5.3.68 Low-Speed to Doze Timer Register (Index 84h)

This read/write register is used to program the time-out period from Low-Speed PLL mode to Doze mode. The minimum period is 1/16 s; the maximum is 63.75 s. A value of 00h disables the PMU timer. If any activities are detected during the timer counting period, the PMU timer is reset to 00h automatically and returns to High-Speed PLL mode.

	7							0
<b>Field</b>	Time-Out Period in Multiples of 1/16 s or 1/4 s							
<b>Default</b>	0	0	0	0	0	0	0	0

**Note:** The timer granularity can be changed to 1/4 s by setting bit 7 of the PMU Control 2 register at Index AFh.

### 5.3.69 Doze to Sleep Timer Register (Index 85h)

This read/write register is used to program the time-out period from Doze mode to Sleep mode. The minimum period is 4 s; the maximum is 1024 s. A value of 00h disables the PMU timer. If any activities are detected during the timer counting period, the PMU timer is reset to 00h automatically and returns to High-Speed PLL mode.

	7							0
<b>Field</b>	Time-Out Period in Multiples of 4 s							
<b>Default</b>	0	0	0	0	0	0	0	0

### 5.3.70 Sleep to Suspend Timer Register (Index 86h)

This read/write register is used to program the time-out period from Sleep mode to Suspend mode. The minimum period is 1/16 s; the maximum is 16 s. A value of 00h disables the PMU timer. If a wake-up or the Resume key is detected during the timer counting period, the PMU timer is reset to 00h automatically and the system returns to High-Speed PLL mode.

	7							0
<b>Field</b>	Time-Out Period in Multiples of 1/16 s							
<b>Default</b>	0	0	0	0	0	0	0	0

### 5.3.71 Suspend to Off Timer Register (Index 87h)

This read/write register is used to program the time-out period from Suspend mode to Off mode. The minimum period is 1 min; the maximum is 256 min. A value of 00h disables the PMU timer. If a wake-up or the Resume key is detected during the timer counting period, the counter is reset to 00h automatically and the system returns to High-Speed PLL mode.

	7							0
<b>Field</b>	Time-Out Period in Multiples of 1 Min							
<b>Default</b>	0	0	0	0	0	0	0	0

### 5.3.72 Software Mode Control Register (Index 88h)

This register allows software to force the system into a particular mode.

	7				0			
<b>Field</b>						SPC		
<b>Bit</b>						SPC2	SPC1	SPC0
<b>Default</b>	0	0	0	0	0	0	0	0

Bit	Name	R/W	Function
7–3		W	(Reserved)
2	SPC2	W	Software command control bit 2
1	SPC1	W	Software command control bit 1
0	SPC0	W	Software command control bit 0

**Notes:**

*Do not force the PMU into Suspend mode when using an LCD panel. Doing so causes LVDD and LVEE to switch off simultaneously, violating LCD-panel power-sequencing requirements. See Chapter 1, “Power Management.”*

*Do not force the PMU from a clock-stopped state (Sleep, Suspend, or Off mode) into the Low-Speed PLL mode or Doze mode. Instead, force the PMU into High-Speed PLL mode. This ensures that internal flags are properly reset for future PMU transitions.*

Table 5-36 PMU Mode Select Logic

SPC2	SPC1	SPC0	Mode
0	0	0	High-Speed PLL
1	0	0	Low-Speed PLL
0	0	1	Doze
0	1	1	Sleep
0	1	0	Suspend

### 5.3.73 General-Purpose I/O 0 Register (Index 89h)

This is a write-only register. This register is used to control the PGP0 pin in either direct-control mode or address-decode mode when PGP0 is configured as an output. In direct-control mode, the state of PGP0 is controlled by bit 7. When bit 7 is 1, the PGP0 output is Low. When bit 7 is 0, PGP0 is High. In address-decode mode, PGP0 functions as a simple address decode. PGP0 is High until the SA9–SA3 signals match bits 6–0 of this register, at which time PGP0 goes Low for as long as the signals match. PGP0 can also be gated internally with the I/O Write command signal. The General-Purpose I/O Control register at Index 91h is used to select how PGP0 operates as an output. Bit 6 of the Miscellaneous 6 register at Index 70h is used for PGP0 direction control.

	7							0
<b>Field</b>	Address Bits 9–3							
<b>Bit</b>	DX	A9	A8	A7	A6	A5	A4	A3
<b>Default</b>	0	0	0	0	0	0	0	0

### 5.3.74 PCMCIA REGA Address Register (Index 8Ah)

This write-only register holds a programmable I/O address for controlling the  $\overline{\text{REG\_A}}$  pin to the card. Once this register is programmed, any write to this programmed address changes the state of the  $\overline{\text{REG\_A}}$  pin to the inverted state of data-bus bit 0. This functionality is provided to access the attribute memory in PCMCIA Socket A. Normally  $\overline{\text{REG\_A}}$  is High during the common-memory cycles, but it should be Low during attribute-memory cycles. Software can access this register to make  $\overline{\text{REG\_A}}$  Low and then perform memory cycles to access the attribute memory.

	7							0
<b>Field</b>	I/O Address, Bits 9–2							
<b>Bit</b>	A9	A8	A7	A6	A5	A4	A3	A2
<b>Default</b>	0	0	0	0	0	0	0	0

**Note:** Bits 7–0 correspond to bits 9–2 of the system-address bus. Therefore, the PCMCIA REGA Address register can be programmed only at a 4-byte boundary in the system I/O address space.

### 5.3.75 Reserved Register (Index 8Bh)

This index location is reserved.

### 5.3.76 I/O Activity Address 0 Register (Index 8Ch)

This register is used by the PMU software to program the I/O address that the activity monitor checks (bit 4 of the Activity Mask 2 register at Index 76h is the mask). This is a write-only register. Status is read from bit 4 of the Activity Status 2 register at Index A1h.

	7							0
<b>Field</b>	I/O Address, Bits 9–3							
<b>Bit</b>	(Reserved)	A9	A8	A7	A6	A5	A4	A3
<b>Default</b>	0	0	0	0	0	0	0	0

### 5.3.77 I/O Activity Address 1 Register (Index 8Dh)

This register is used by the PMU software to program the I/O address that the activity monitor checks (bit 5 of the Activity Mask 2 register at Index 76h is the mask). This is a write-only register. Status is read from bit 5 of the Activity Status 2 register at Index A1h.

	7								0
<b>Field</b>		I/O Address, Bits 9–3							
<b>Bit</b>	(Reserved)	A9	A8	A7	A6	A5	A4	A3	
<b>Default</b>	0	0	0	0	0	0	0	0	

### 5.3.78 Reserved Register (Index 8Eh)

This index location is reserved.

### 5.3.79 Clock Control Register (Index 8Fh)

This register is used to program the crystal restart-delay time and CPU restart-delay time. Bits 7–0 of this register are not reset when exiting Micro Power Off mode.

	7						0		
<b>Bit</b>	(Reserved)					XST2	XST1	XST0	
<b>Default</b>	0	0	0	0	0	0	0	0	

Bit	Name	R/W	Function
7		W	(Reserved—must be 1)
6–5		W	(Reserved)
4		W	(Reserved—must be 0)
3		W	(Reserved)
2	XST2	W	PLL restart delay time control bit 2
1	XST1	W	PLL restart delay time control bit 1
0	XST0	W	PLL restart delay time control bit 0



Table 5-37 PLL Restart Time Select Logic

XST2	XST1	XST0	PLL Restart Time
0	0	0	4 ms
0	0	1	8 ms
0	1	0	16 ms
0	1	1	32 ms
1	0	0	64 ms
1	0	1	128 ms
1	1	0	256 ms
1	1	1	1 s

**Note:**

A 256-ms restart time is recommended when clocks are started from a PMU state where the low-speed PLL is disabled. A 128-ms restart time is recommended when clocks are started from a PMU state where the low-speed PLL is enabled. Results are not guaranteed if values less than these are used. This restart value applies to both High-Speed PLL and Low-Speed PLL mode restarts.

## 5.3.80 Reserved Register (Index 90h)

This index location is reserved.

## 5.3.81 General-Purpose I/O Control Register (Index 91h)

This register is used to control the PGP3–PGP0 pins. PGP0 and PGP1 can be driven directly, gated by I/O commands, or driven by simple address decodes. PGP2 and PGP3 can be automatically switched to a programmed level when the Power Management Unit switches to Off mode, gated by I/O commands, or driven by simple address decodes. When implemented as simple-address decodes, the PGP pin goes active when the address bus matches the address bits (bits 9–3) programmed into the General-Purpose I/O register for that PGP pin. All remaining address bits are Don't Cares. It is up to the system designer to externally qualify this pin with the  $\overline{IOR}$ ,  $\overline{IOW}$ ,  $\overline{MEMR}$ , or  $\overline{MEMW}$  command signal. For more information, see the descriptions for each of the General-Purpose I/O registers in this chapter (indexes 89h, 94h, 95h, and 9Ch).

	7						0	
Field	PGP Pin 3		PGP Pin 2		PGP Pin1		PGP Pin 0	
Bit	PG3IO1	PG3IO0	PG2IO1	PG2IO0	PG1IO1	PG1IO0	PG0IO1	PG0IO0
Default	0	0	0	0	0	0	0	0

Bit	Name	R/W	Function
7	PG3IO1	W	Program general-purpose pin 3 gate control bit 1
6	PG3IO0	W	Program general-purpose pin 3 gate control bit 0
5	PG2IO1	W	Program general-purpose pin 2 gate control bit 1
4	PG2IO0	W	Program general-purpose pin 2 gate control bit 0
3	PG1IO1	W	Program general-purpose pin 1 gate control bit 1
2	PG1IO0	W	Program general-purpose pin 1 gate control bit 0
1	PG0IO1	W	Program general-purpose pin 0 gate control bit 1
0	PG0IO0	W	Program general-purpose pin 0 gate control bit 0

**PG0IO1 PG0IO0 PGP0 Output**

0	0	PGP0 direct control mode
1	0	PGP0 gates with I/O Write command
0	1	PGP0 acts as an address decode only
1	1	PGP0 acts as an address decode only

**PG1IO1 PG1IO0 PGP1 Output**

0	0	PGP1 direct control mode
1	0	PGP1 acts as an address decode only
0	1	PGP1 gates with I/O Read command
1	1	PGP1 acts as an address decode only

**PG2IO1 PG2IO0 PGP2 Output**

0	0	PGP2 automatically switches to the inverse of bit 7 of the General-Purpose I/O 2 register at Index 94h when the PMU is switched to Off mode
1	0	PGP2 gates with I/O Write command
0	1	PGP2 acts as an address decode only
1	1	PGP2 acts as an address decode only

**PG3IO1 PG3IO0 PGP3 Output**

0	0	PGP3 automatically switches to the inverse of bit 7 of the General-Purpose I/O 3 register at Index 95h when the PMU is switched to Off mode
1	0	PGP3 acts as an address decode only
0	1	PGP3 gates with I/O Read command
1	1	PGP3 acts as an address decode only

## 5.3.82 UART Clock Enable Register (Index 92h)

This register is used to control the UART clock.

	7							0
<b>Bit</b>	(Reserved)							ENCLK
<b>Default</b>	0	0	0	0	0	0	0	

Bit	Name	R/W	Function
7-1		W	(Reserved)
0	ENCLK	W	1 = Enable clock to internal 16450 UART

## 5.3.83 Reserved Register (Index 93h)

This index location is reserved and must be 0.

## 5.3.84 General-Purpose I/O 2 Register (Index 94h)

This is a write-only register. This register is used to control the PGP2 pin in either power-management mode or address-decode mode. In power-management mode, the state of PGP2 is High when the PMU is not in Off mode. When the PMU transitions to Off mode, the state of PGP2 is determined by bit 7. When bit 7 is 1, PGP2 is Low. When bit 7 is 0, PGP2 is High. In address-decode mode, PGP2 functions as a simple address decode. PGP2 is High until the SA9–SA3 signals match bits 6–0 of this register, at which time PGP2 goes Low for as long as the signals match. PGP2 can also be gated internally with the I/O Write command signal. The General-Purpose I/O Control register at Index 91h is used to select how PGP2 operates as an output. PGP2 cannot operate as an input.

	7							0
<b>Field</b>	Address Bits 9–3							
<b>Bit</b>	DX	A9	A8	A7	A6	A5	A4	A3
<b>Default</b>	0	0	0	0	0	0	0	0

## 5.3.85 General-Purpose I/O 3 Register (Index 95h)

This is a write-only register. This register is used to control the PGP3 pin in either power-management mode or address-decode mode. In power-management mode, the state of PGP3 is High when the PMU is not in Off mode. When the PMU transitions to Off mode, the state of PGP3 is determined by bit 7. When bit 7 is 1, PGP3 is Low. When bit 7 is 0, PGP3 is High. In address-decode mode, PGP3 functions as a simple address decode. PGP3 is High until the SA9–SA3 signals match bits 6–0 of this register, at which time PGP3 goes Low for as long as the signals match. PGP3 can also be gated internally with the I/O Read command signal. The General-Purpose I/O Control register at Index 91h is used to select how PGP3 operates as an output. PGP3 cannot operate as an input.

	7							0
<b>Field</b>	Address Bits 9–3							
<b>Bit</b>	DX	A9	A8	A7	A6	A5	A4	A3
<b>Default</b>	0	0	0	0	0	0	0	0

### 5.3.86 Reserved Registers (Indexes 96–99h)

These index locations are reserved.

### 5.3.87 Memory Write Activity Lower Boundary Register (Index 9Ah)

This register specifies the low memory-address boundary that the activity monitor counts as activity (addresses A23–A20 are all 0). The high memory address is defined by the Programmable Memory Write Activity Upper Boundary register at Index 9Bh. This activity is enabled/masked by bit 7 of the Activity Mask 2 register at Index 76h. This is a write-only register.

An activity is generated when a memory write to any address greater than the low address and less than the high address occurs. Address bits A13–A0 are Don't Cares when determining if the memory address is within the programmable range.

	7							0
<b>Bit</b>	LSA19	LSA18	LSA17	LSA16	LSA15	LSA14	(Reserved)	ENHIT
<b>Default</b>	0	0	0	0	0	0	0	0

Bit	Name	R/W	Function
7	LSA19	W	Lower boundary memory address SA19
6	LSA18	W	Lower boundary memory address SA18
5	LSA17	W	Lower boundary memory address SA17
4	LSA16	W	Lower boundary memory address SA16
3	LSA15	W	Lower boundary memory address SA15
2	LSA14	W	Lower boundary memory address SA14
1		W	(Reserved)
0	ENHIT	W	Hit count function enable

### 5.3.88 Memory Write Activity Upper Boundary Register (Index 9Bh)

This register specifies the high memory-address boundary that the activity monitor counts as activity (addresses A23–A20 are all 0). The low memory-address boundary is specified by the Programmable Memory Write Activity Lower Boundary register at Index 9Ah. This activity is enabled/masked by bit 7 of the Activity Mask 2 register at Index 76h. This is a write-only register.

An activity is generated when a memory write to any address greater than the low address and less than the high address occurs.

	7							0
<b>Field</b>						Hit Count Limit		
<b>Bit</b>	HSA19	HSA18	HSA17	HSA16	(Reserved)	SC2	SC1	SC0
<b>Default</b>	0	0	0	0	0	0	0	0

Bit	Name	R/W	Function
7	HSA19	W	Upper boundary memory address SA19
6	HSA18	W	Upper boundary memory address SA18
5	HSA17	W	Upper boundary memory address SA17
4	HSA16	W	Upper boundary memory address SA16
3		W	(Reserved)
2–0	SC2–SC0	W	Hit count limit, bits 2–0

Table 5-38 Hit-Count Limit Bit Logic

Bit			Hit-Count Limit
2	1	0	
0	0	0	1
0	0	1	2
0	1	0	4
0	1	1	8
1	0	0	16
1	0	1	32
1	1	0	64
1	1	1	128

**Notes:**

The hit count is cleared when a PMU state-transition counter expires or any other activity occurs. Thus, all memory writes must occur in the same PMU state, without other activity, before they count as activity.

If the hit-count limit is 2 and two or more memory write cycles fall into the range specified by the address range, it counts as activity. If bit 0 of the Memory Write Activity Lower Boundary register at Index 9Ah is not set, then this function is disabled.

## 5.3.89 General-Purpose I/O 1 Register (Index 9Ch)

This is a write-only register. This register is used to control the PGP1 pin in either direct-control mode or address-decode mode when PGP1 is configured as an output. In direct-control mode, the state of PGP1 is controlled by bit 7. When bit 7 is 1, PGP1 is Low. When bit 7 is 0, PGP1 is High. In address-decode mode, PGP1 functions as a simple address decode. PGP1 is High until the SA9–SA3 signals match bits 6–0 of this register, at which time PGP1 goes Low for as long as the signals match. PGP1 can also be gated internally with the I/O Read command signal. The General-Purpose I/O Control register at Index 91h is used to select how PGP1 operates as an output. Bit 2 of the MMSB Control register at Index 74h is used for PGP1 direction control.

	7	Address Bits 9–3						0
Field	DX	A9	A8	A7	A6	A5	A4	A3
Default	0	0	0	0	0	0	0	0

### 5.3.90 Reserved Register (Index 9Dh)

This index location is reserved and must be set to 40h.

### 5.3.91 PCMCIA REGB Address Register (Index 9Eh)

This write-only register holds a programmable I/O address for controlling the  $\overline{\text{REG\_B}}$  pin to the card. Once this register is programmed, any write to this programmed address changes the state of the  $\overline{\text{REG\_B}}$  pin to the inverted state of data-bus bit 0. This functionality is provided to access the attribute memory in PCMCIA Socket B. Normally, the  $\overline{\text{REG\_B}}$  signal is High during the common-memory cycles, but it should be Low during attribute-memory cycles. Software can access this register to make the  $\overline{\text{REG\_B}}$  pin Low and then perform memory cycles to access the attribute memory.

	7							0	
Field	I/O Address, Bits 9–2								
Bit	A9	A8	A7	A6	A5	A4	A3	A2	
Default	0	0	0	0	0	0	0	0	

**Note:**

Bits 7–0 of this register correspond to bits 9–2 of the system-address bus. Therefore, the PCMCIA REGB Address register can be programmed only at a 4-byte boundary in the system I/O address space.

### 5.3.92 Auto Low-Speed Control Register (Index 9Fh)

This register controls the auto low-speed trigger and duration period. Bit 3 of the Control B register at Index 77h disables or enables the trigger; bit 6 of the I/O Wait State register at Index 61h enables the high-speed CPU clock.

	7				0			
Field	(Reserved)				Low Speed		Trigger Period	
Bit	(Reserved)				LOW1	LOW0	T1	T0
Default	0	0	0	0	0	0	0	0

Bit	Name	R/W	Function
7–4			(Reserved)
3	LOW1	W	Low-speed duration bit 1
2	LOW0	W	Low-speed duration bit 0
1	T1	W	Trigger period bit 1
0	T0	W	Trigger period bit 0

Table 5-39 Trigger Period Select Logic

T1	T0	Trigger Period
0	0	4 s
0	1	8 s
1	0	16 s
1	1	32 s

Table 5-40 Low-Speed Duration Period Select Logic

LOW1	LOW0	Low-Speed Period
0	0	0.25 s
0	1	0.5 s
1	0	1 s
1	1	2 s

### 5.3.93 Activity Status 1 Register (Index A0h)

This register contains the activity status of system peripherals and signals. A 1 indicates activity. Software can clear this register by writing any data to it.

	7							0
Bit	INT	ACIN	MMS	KB	DRQ3	DRQ2	DRQ1	DRQ0
Default	0	0	0	0	0	0	0	0

Bit	Name	R/W	Function
7	INT	R/W	1 = Interrupt was detected active
6	ACIN	R/W	1 = AC adapter input or bit 5 of the Miscellaneous 6 register at Index 70h was detected active
5	MMS	R/W	1 = MMS was detected active
4	KB	R/W	1 = Keyboard was detected active
3	DRQ3	R/W	1 = DRQ3 was detected active
2	DRQ2	R/W	1 = DRQ2 was detected active
1	DRQ1	R/W	1 = DRQ1 was detected active
0	DRQ0	R/W	1 = DRQ5, DRQ6, or DRQ7 was detected active

**Note:**

*INT includes all interrupts from IRQ2 to IRQ15.*

### 5.3.94 Activity Status 2 Register (Index A1h)

This register contains the activity status of system peripherals and signals. A 1 indicates activity. Software can clear this register by writing any data to it.

	7							0
Bit	PMW	VD	PIO1	PIO0	COM	HD	FD	LPT
Default	0	0	0	0	0	0	0	0

Bit	Name	R/W	Function
7	PMW	R	1 = Programmable memory range write access was detected active
6	VD	R	1 = Video memory write (0B0000–0BFFFFh) was detected active
5	PIO1	R	1 = General-Purpose I/O 1 register was detected active
4	PIO0	R	1 = General-Purpose I/O 0 register was detected active
3	COM	R/W	1 = COM1 or COM2 was detected active
2	HD	R/W	1 = Hard drive was detected active
1	FD	R/W	1 = Floppy disk drive was detected active
0	LPT	R/W	1 = LPT1, LPT2, or LPT3 was detected active

### 5.3.95 PCMCIA Socket A Status Register (Index A2h)

When Socket A is configured for a memory card, this register contains the status of the WP, BVD1, BVD2, RDY, and Card Detect pins. When Socket A is configured as an I/O card, only the Card Detect pin status (bit 5) is valid. For information on the Socket B status, see “PCMCIA Socket B Status Register (Index 0Ch)” on page 5-19.

Writing any value to this register resets all PMU SMI requests. For more information on SMIs, see “SMI and NMI Control” on page 1-25.

	7							0
<b>Bit</b>	LPH1	(Reserved)	CD	RDY	(Reserved)	BVDA1	BVDA2	WP
<b>Default</b>	0	0	0	0	0	0	0	0

Bit	Name	R/W	Function
7	LPH1	R	Status of LPH pin
6		R	(Reserved)
5	CD	R	PCMCIA Socket A Card detect status (0 = Card detected)
4	RDY	R	PCMCIA Socket A RDY status (1 = Card ready)
3		R	(Reserved)
2	BVDA1	R	PCMCIA Socket A battery low detect 1 (0 = Battery low)
1	BVDA2	R	PCMCIA Socket A battery low detect 2 (0 = Battery low)
0	WP	R	PCMCIA Socket A write-protect status (1 = Card write protect)

### 5.3.96 CPU Status 0 Register (Index A3h)

	7							0
<b>Field</b>		Last Mode			Battery Low			
<b>Bit</b>	(Reserved)	LIND2	LIND1	LIND0	BL3IN	BL2IN	BL1IN	PG0IN
<b>Default</b>	0	0	0	0	0	0	0	0

Bit	Name	R/W	Function
7		R	(Reserved)
6	LIND2	R	Last PMU state indicator 2
5	LIND1	R	Last PMU state indicator 1
4	LIND0	R	Last PMU state indicator 0
3	BL3IN	R	Battery low detect pin 2 (BL3) input data
2	BL2IN	R	Battery low detect pin 2 (BL2) input data
1	BL1IN	R	Battery low detect pin 1 (BL1) input data
0	PG0IN	R	Programmable general-purpose I/O pin 0 (PGP0) input data

**Note:**

The last PMU mode indicator bits are not updated for PMU state transitions caused by writes to the Software Mode Control register at Index 88h.



Table 5-41 Last PMU Mode Indicator Bits

LIND2	LIND1	LIND0	Last Mode
0	0	0	High-Speed PLL
1	0	0	Low-Speed PLL
0	0	1	Doze
0	1	1	Sleep
0	1	0	Suspend

## 5.3.97 CPU Status 1 Register (Index A4h)

The last mode status is accurate only between the time when an event transitions the PMU to High-Speed PLL mode and when the timer expires in High-Speed PLL mode. This function is intended to allow the system to know what state the PMU was in when an event brought it out of a lower-power mode than High-Speed PLL mode. This register must be read before the timer expires in High-Speed PLL mode.

Field	7			0				
Bit	ACIN	(Reserved)			PG1IN	Present State		
Default	0	0	0	0	0	PIND2	PIND1	PIND0
	0	0	0	0	0	0	0	0

Bit	Name	R/W	Function
7	ACIN	R	ACIN input status
6-4		R	(Reserved)
3	PG1IN	R	Status of the PGP1 pin
2	PIND2	R	PMU Indicator bit 2
1	PIND1	R	PMU Indicator bit 1
0	PIND0	R	PMU Indicator bit 0

Table 5-42 Present PMU Mode Indicator Bits

PIND2	PIND1	PIND0	Present Mode
0	0	0	High-Speed PLL
1	0	0	Low-Speed PLL
0	0	1	Doze
0	1	1	Sleep
0	1	0	Suspend

### 5.3.98 NMI/SMI Control Register (Index A5h)

Reading this register returns the status information on the source of an NMI or SMI. Writing this register allows the mode change to occur on the next refresh cycle after the write. For example, if the ÉlanSC300 microcontroller is programmed to generate an NMI or SMI when the mode changes from Low-Speed PLL mode to Doze mode, then the PMU I stays in Low-Speed PLL mode after the generation of an NMI or SMI until software writes to this register to enable the mode change.

	7							0
<b>Bit</b>	BL3	BL2	BL1	SU	SP	DZ	ON	RESUME
<b>Default</b>	0	0	0	0	0	0	0	0

Bit	Name	R/W	Function
7	BL3	R/W	1 = The $\overline{BL3}$ (battery low third warning) generated an NMI or SMI
6	BL2	R/W	1 = The $\overline{BL2}$ (battery low second warning) generated an NMI or SMI
5	BL1	R/W	1 = The $\overline{BL1}$ (battery low first warning) generated an NMI or SMI
4	SU	R/W	1 = The NMI or SMI was generated from Suspend mode to Off mode
3	SP	R/W	1 = The NMI or SMI was generated from Sleep mode to Suspend mode
2	DZ	R/W	1 = The NMI or SMI was generated from Doze mode to Sleep mode
1	ON	R/W	1 = The NMI or SMI was generated from Low-Speed PLL mode to Doze mode
0	RESUME	R/W	1 = The NMI or SMI was generated by the $\overline{SUS/RES}$ pin

### 5.3.99 PCMCIA Status Change Register (Index A6h)

After receiving any interrupt due to a pin change at the card interface, software must clear the bit in this register that corresponds to the pin that generated the interrupt. To clear any of these bits, software must write a 0 followed by a 1 to the corresponding bit. See “PCMCIA Status Change IRQ Enable Register (Index 0Dh)” on page 5-19 and “PCMCIA Status Change IRQ Redirection Register (Index 0Eh)” on page 5-20.

For example, to clear all the bits, write 00h followed by FFh to the PCMCIA Status Change register. To clear only bit 0, write FEh followed by FFh to this register.

	7							0
<b>Bit</b>	BICBL2CHG	AICBL2CHG	BICBL1CHG	AICBL1CHG	BRDYCHG	ARDYCHG	ICCDBCHG	ICCDACHG
<b>Default</b>	0	0	0	0	0	0	0	0

Bit	Name	R/W	Function
7	BICBL2CHG	R	Socket B BVD2B bit changed
6	AICBL2CHG	R	Socket A BVD2A bit changed
5	BICBL1CHG	R	Socket B BVD1B bit changed
4	AICBL1CHG	R	Socket A BVD1A bit changed
3	BRDYCHG	R	Socket B RDY bit changed
2	ARDYCHG	R	Socket A RDY bit changed
1	ICCDBCHG	R	Socket B $\overline{CD}$ bit changed
0	ICCDACHG	R	Socket A $\overline{CD}$ bit changed

**Bit 7** If the card is configured as memory, a falling edge of the BVDB2 pin from Socket B sets this bit. If the card is configured as I/O, this bit is not affected.

**Bit 6** If the card is configured as memory, a falling edge of the BVDA2 pin from Socket A sets this bit. If the card is configured as I/O, this bit is not affected.

**Bit 5** If the card is configured as memory, a falling edge of the BVDB1 pin from Socket B sets this bit. If the card is configured as I/O, a falling edge of the STSCHG\_B pin from Socket A sets this bit.

**Bit 4** If the card is configured as memory, a falling edge of the BVDA1 pin from Socket A sets this bit. If the card is configured as I/O, a falling edge of the STSCHG\_A pin from Socket A sets this bit.

**Bit 3** If the card is configured as memory, a rising edge of the RDY pin from Socket B sets this bit. If the card is configured as I/O, this bit is not affected.

**Bit 2** If the card is configured as memory, a rising edge of the RDY pin from Socket A sets this bit. If the card is configured as I/O, this bit is not affected.

**Bit 1** If this bit is set,  $\overline{CD\_B}$  has changed from High to Low or from Low to High.

**Bit 0** If this bit is set,  $\overline{CD\_A}$  has changed from High to Low or from Low to High.

### 5.3.100 PMU Control 1 Register (Index A7h)

This register is used to control various PMU functions.

	7				0			
<b>Bit</b>	(Reserved)		ENADIN2	ENADIN1	(Reserved)		SLREF	REFSEL
<b>Default</b>	0	0	0	0	0	0	0	0

Bit	Name	R/W	Function
7–6		R/W	(Reserved)
5	ENADIN2	R/W	1 = ÉlanSC300 microcontroller disables data propagation to UART and PMU controller in memory cycle
4	ENADIN1	R/W	1 = ÉlanSC300 microcontroller disables data propagation to display controller in memory cycle
3–2		R/W	(Reserved)
1	SLREF	R/W	Enable slow refresh for DRAM (if bit 0 is set, this bit's setting has no meaning): 0 = Enable slow refresh. When slow refresh is enabled, the Version register at Index 64h programs the 32-kHz divisor for the source. 1 = Disable slow refresh. The refresh rate is 65536/s unless bit 0 is set. This bit is not reset when exiting Micro Power Off mode.
0	REFSEL	R/W	Select DRAM refresh source (this bit setting overrides any setting for bit 1): 0 = Use 32-kHz clock multiplied by 2 (65536/s) as refresh source 1 = Use 8254 as refresh source

**Note:**

*If the PMU is enabled to stop the 8254 clock, setting bit 0 will cause the DRAM refresh to be lost.*

### 5.3.101 MMSA Socket Register (Index A8h)

This register is used to map MMSA pages to Sockets A or B.

	7							0
Bit	MMSP7	MMSP6	MMSP5	MMSP4	MMSP3	MMSP2	MMSP1	MMSP0
Default	0	0	0	0	0	0	0	0

Bit	Name	R/W	Function
7	MMSP7	R/W	MMSA page 7 mapping: 0 = Socket A 1 = Socket B
6	MMSP6	R/W	MMSA page 6 mapping: 0 = Socket A 1 = Socket B
5	MMSP5	R/W	MMSA page 5 mapping: 0 = Socket A 1 = Socket B
4	MMSP4	R/W	MMSA page 4 mapping: 0 = Socket A 1 = Socket B
3	MMSP3	R/W	MMSA page 3 mapping: 0 = Socket A 1 = Socket B
2	MMSP2	R/W	MMSA page 2 mapping: 0 = Socket A 1 = Socket B
1	MMSP1	R/W	MMSA page 1 mapping: 0 = Socket A 1 = Socket B
0	MMSP0	R/W	MMSA page 0 mapping: 0 = Socket A 1 = Socket B

## 5.3.102 MMSB Socket Register (Index A9h)

This register is used to map MMSB pages to Sockets A or B.

	7							0
Bit	GENSMI	(Reserved)	SMIA23	SMIA22	MMS1P3	MMS1P2	MMS1P1	MMS1P0
Default	0	0	0	0	0	0	0	0

Bit	Name	R/W	Function
7	GENSMI	R/W	1 = ÉlanSC300 microcontroller generates an SMI instead of an NMI for the sources enabled in the NMI/SMI Enable register at Index 82h. (Bit 7 of the AT-Compatible Port 70h must be 0 for NMIs to occur.)
6		R/W	(Reserved)
5	SMIA23	R/W	SMI MMS translate address A23 for SMI only. See “SMI MMS Page Register (Index AAh)” on page 5-73.
4	SMIA22	R/W	SMI MMS translate address A22 for SMI only. See “SMI MMS Page Register (Index AAh)” on page 5-73.
3	MMS1P3	R/W	MMSB page 3 mapping: 0 = Socket A 1 = Socket B
2	MMS1P2	R/W	MMSB page 2 mapping: 0 = Socket A 1 = Socket B
1	MMS1P1	R/W	MMSB page 1 mapping: 0 = Socket A 1 = Socket B
0	MMS1P0	R/W	MMSB page 0 mapping: 0 = Socket A 1 = Socket B

## 5.3.103 SMI MMS Page Register (Index AAh)

This register is used to control the SMI MMS page register.

	7							0
Bit	SMIA21	SMIA20	SMIA19	SMIA18	SMIA17	SMIA16	SMIA15	SMIA14
Default	0	0	0	0	0	0	0	0

**Note:**

SMI MMS is active only in System Management Mode (SMM).

Bit	Name	R/W	Function
7	SMIA21	R/W	SMI MMS translate address bit A21
6	SMIA20	R/W	SMI MMS translate address bit A20
5	SMIA19	R/W	SMI MMS translate address bit A19
4	SMIA18	R/W	SMI MMS translate address bit A18
3	SMIA17	R/W	SMI MMS translate address bit A17
2	SMIA16	R/W	SMI MMS translate address bit A16
1	SMIA15	R/W	SMI MMS translate address bit A15
0	SMIA14	R/W	SMI MMS translate address bit A14

**Note:**

A special MMS with page address 060000h is used for the SMI function. SMI accesses that use the MMS always map to on-board memory.

### 5.3.104 Power Control 3 Register (Index ABh)

This register activates the PMC3 and PMC4 output pins in High-Speed PLL mode, Low-Speed PLL mode, Doze mode, Sleep mode, and Suspend mode. The PMC4 pin can be programmed to toggle inactive when the hard-drive timer expires. For more information on using PMC4 with the hard-drive timer, see “Accesses to Powered-Down Device SMI” on page 1-29.

	7							0
<b>Bit</b>	SU4	SP4	DZ4	FO4	SU3	SP3	DZ3	FO3
<b>Default</b>	0	0	0	0	0	0	0	0

Bit	Name	R/W	Function
7	SU4	R/W	State of PMC4 pin in Suspend mode
6	SP4	R/W	State of PMC4 pin in Sleep mode
5	DZ4	R/W	State of PMC4 pin in Doze mode
4	FO4	R/W	State of PMC4 pin in High-Speed PLL and Low-Speed PLL modes
3	SU3	R/W	Inverse state of PMC3 pin in Suspend mode
2	SP3	R/W	Inverse state of PMC3 pin in Sleep mode
1	DZ3	R/W	Inverse state of PMC3 pin in Doze mode
0	FO3	R/W	Inverse state of PMC3 pin in High-Speed PLL and Low-Speed PLL modes

**Notes:**

*The initial state of PMC4 after power-on is Low. When the bit is 0, the corresponding PMC output for that PMU mode is Low.*

*The initial state of PMC3 after power-on is High. When the bit is 0, the corresponding PMC output for that PMU mode is High.*

*PMC4 is the pin associated with the dedicated hard-drive address activity decode and timer. If PMC4 is not used for the hard drive, the hard-drive timer and associated control must be disabled.*

*Bits 7–4 of this register are cleared when the hard-drive timer expires.*

## 5.3.105 Power Control 4 Register (Index ACh)

This register activates the PMC0 and PMC1 output pins in High-Speed PLL mode, Low-Speed PLL mode, Doze mode, Sleep mode, and Suspend mode. The PMC0 pin can be programmed to toggle inactive when the floppy-disk-drive timer expires. The PMC1 pin can be programmed to toggle inactive when the PIO timer expires. For more information on using the PMC pins with the timers, see “Accesses to Powered-Down Device SMI” on page 1-29.

	7							0
<b>Bit</b>	SU1	SP1	DZ1	FO1	SU0	SP0	DZ0	FO0
<b>Default</b>	0	0	0	0	0	0	0	0

Bit	Name	R/W	Function
7	SU1	R/W	State of PMC1 pin in Suspend mode
6	SP1	R/W	State of PMC1 pin in Sleep mode
5	DZ1	R/W	State of PMC1 pin in Doze mode
4	FO1	R/W	State of PMC1 pin in High-Speed PLL and Low-Speed PLL modes
3	SU0	R/W	State of PMC0 pin in Suspend mode
2	SP0	R/W	State of PMC0 pin in Sleep mode
1	DZ0	R/W	State of PMC0 pin in Doze mode
0	FO0	R/W	State of PMC0 pin in High-Speed PLL and Low-Speed PLL modes

**Notes:**

*The initial state of PMC0 and PMC1 after power-on is Low. When the bit is 0, the corresponding PMC output for that PMU mode is Low.*

*PMC0 is associated with the dedicated floppy-disk-drive address activity decode and timer. If PMC0 is not used for the floppy disk drive, the floppy-disk-drive timer and associated control must be disabled.*

*PMC1 is associated with the dedicated PIO address activity decode and timer. If PMC1 is not used for the PIO, the PIO timer and associated control must be disabled.*

**Bits 7–4** These bits are cleared as a result of the PIO timer expiring.

**Bits 3–0** These bits are cleared as a result of the floppy-disk-drive timer expiring.

### 5.3.106 PMU Control 3 Register (Index ADh)

This register controls several PMU functions. When switching the low-speed frequency, the low-speed PLL is divided. This divided frequency is selected for use when the PMU is not in the Low-Speed PLL mode. The Low-Speed Clock Select bits (bits 1 and 0 of the PMU Control 3 register) should be modified only when the PMU is not in Low-Speed PLL mode.

	7						0	
<b>Field</b>							Clock Select	
<b>Bit</b>	(Reserved)	IRQ0SMIEN	(Reserved)	ENACIN	XTKBDEN	MAINOFF	ONCLK1	ONCLK0
<b>Default</b>	0	0	0	0	0	0	0	0

Bit	Name	R/W	Function
7		R/W	(Reserved)
6	IRQ0SMIEN	R/W	1 = 8254 channel 0 generates an SMI instead of the normal IRQ0. Note that the IRQ0 in the PIC must be enabled to allow this condition.
5		R/W	(Reserved)
4	ENACIN	R/W	1 = ACIN is treated as activity
3	XTKBDEN	R/W	1 = Allows the 8042CS and SYSCLK pins to become three-stated so they may be used as inputs, all other requirements being met. It also qualifies an internal decode so that port 60h is read as an internal port. It further switches a multiplexer to vector IRQ1 from the external pin to the output of this circuitry.
2	MAINOFF	R/W	1 = ÉlanSC300 microcontroller turns off the high-speed PLL in Low-Speed PLL mode; otherwise, the high-speed PLL is turned off in Doze mode.
1	ONCLK1	R/W	Low-Speed PLL mode CPU clock bit 1
0	ONCLK0	R/W	Low-Speed PLL mode CPU clock bit 0

**Bit 4** This bit is used in conjunction with bit 6 of the Activity Mask 1 register at Index 75h. If bit 4 of the PMU Control 3 register is 0, bit 6 of the Activity Mask 1 register has no function.

**Bits 1–0** These bits only have an effect in Low-Speed PLL mode.

Table 5-43 Low-Speed PLL Mode CPU Clock Speed Select

ONCLK1	ONCLK0	Clock Frequency to CPU	Internal CPU Operation Speed
0	0	9.216 MHz	4.608 MHz
0	1	4.608 MHz	2.304 MHz
1	0	2.304 MHz	1.152 MHz
1	1	1.152 MHz	0.576 MHz

### 5.3.107 Reserved Register (Index AEh)

This register is reserved.



## 5.3.108 PMU Control 2 Register (Index AFh)

This register controls several PMU operations.

	7							0
Bit	CHGON	CHGFUSET	BL1LOWSP	(Reserved)				EXTIR0ACT
Default	0	0	0	0	0	0	0	0

Bit	Name	R/W	Function
7	CHGON	R/W	Low-Speed to Doze Mode Timer register (Index 84h) unit value: 1 = 1/4 s 0 = 1/16 s
6	CHGFUSET	R/W	High-Speed to Low-Speed Mode Timer register (Index 83h) unit value: 1 = 1/16 s 0 = 1/512 s
5	BL1LOWSP	R/W	1 = Set CPU clock speed to low speed (9.2 MHz) in High-Speed PLL mode if $\overline{BL1}$ is Low and ACIN is Low.
4–1		R/W	(Reserved)
0	EXTIR0ACT	R/W	1 = Extend CPU run time for an additional 64 refresh cycles following IRQ0 while in Doze mode. This bit is effective only if bit 3 of the MMSB Control register at Index 74h is 1.

## 5.3.109 Function Enable 1 Register (Index B0h)

	7							0
Bit	(Reserved)	X1SEL	EXTSMIEDG	EXTSMIEN	DMASTCLK	EPPMODE	PPISBI	PPBIENB
Default	0	0	0	0	0	0	0	0

Bit	Name	R/W	Function
7		R/W	(Reserved)
6	X1SEL	W	X1OUT clock select: 0 = X1OUT determined by bit 2 of Index B1h 1 = X1OUT driven by BAUDOUT
5	EXTSMIEDG	R/W	External SMI active edge 1 = Active-Low external SMI 0 = Active-High external SMI
4	EXTSMIEN	R/W	1 = Enable external SMI
3	DMASTCLK	R/W	1 = DMA stop clock (power-save mode) enable
2	EPPMODE	R/W	EPP mode enable for parallel port
1	PPISBI	R/W	Bidirectional configuration enable for parallel port
0	PPBIENB	R/W	Bidirectional enable for parallel port

**Bit 3** When this bit is set, the DMA clock runs only when a DMA access is happening; it is stopped between DMA transfers.

**Bit 1** This bit is used to configure the ÉlanSC300 microcontroller's parallel-port control outputs for the parallel-port hardware interface implemented on the system board. When bit 1 is 0,  $\overline{DBUFOE}$  is not generated during parallel-port accesses. This bit should only be 0 if the system is implemented without a system buffer. This bit must be 1 for systems that have a system buffer and implement a parallel port with either an input buffer and output latch, or just an output latch.

**Bit 0** This bit is used to enable the bidirectional control for the parallel port. This bit functions only if bit 1 is 1. When bit 0 is 0, the parallel-port data-latch output-enable signal from the ÉlanSC300 microcontroller's CPU,  $\overline{PPOEN}$ , is forced Low, causing the parallel-port data-latch to drive the parallel-port data bus. When bit 0 is 1, the  $\overline{PPOEN}$  signal is controlled by bit 5 of the Parallel Port Control register.

Table 5-44 Latch and Buffer Logic

PPBIENB	PPISBI	EPPMODE	Mode
0	0	0	Output latch, no system buffer, no input buffer, PP output only
0	0	1	(Invalid combination)
0	1	0	Output latch, system buffer, no input buffer, PP output only
0	1	1	Output latch, system buffer, no input buffer, EPP mode
1	0	0	(Invalid combination)
1	0	1	(Invalid combination)
1	1	0	Output latch, system buffer, input buffer, PP is bidirectional
1	1	1	Output latch, system buffer, input buffer, EPP mode

### 5.3.110 Function Enable 2 Register (Index B1h)

	7							0
<b>Bit</b>	EB_RMMD1	EB_RMMD0	(Reserved)	HSPLLQ1	HSPLLQ0	XTALUSE	LCDDUEN	(Reserved)
<b>Default</b>	0	0	0	0	0	0	0	0

Bit	Name	R/W	Function
7	EB_RMMD1	R/W	RAM mode select bit 0
6	EB_RMMD0	R/W	RAM mode select bit 1
5		R/W	(Reserved—must be 0)
4	HSPLLQ1	R/W	High-speed PLL frequency select, bit 1
3	HSPLLQ0	R/W	High-speed PLL frequency select, bit 0
2	XTALUSE	R/W	Crystal interface: 1 = 14.336 MHz out on X1OUT pin 0 = Three-state X1OUT pin (see bit 6 of the Function Enable 1 register at Index B0h)
1	LCDDUEN	R/W	Enable dual-panel LCD mode
0			(Reserved)

**Bit 7–6** When bit 7 of the PCMCIA Card Reset register at Index B4h is 1, the values of bits 6 and 7 of this register determine the DRAM mode as shown in Table 5-46 on page 5-79.

**Bit 4–3** Do not change these bits while running in the High-Speed PLL mode. To change these bits, first clear bit 6 of the I/O Wait State register at Index 61h, then wait for the next refresh to occur. Then change the high-speed PLL frequency and return to using the high-speed PLL.

**Bit 1** This bit should be set to support dual scan LCD panels. When set, the  $\overline{\text{IOCS16}}$ ,  $\overline{\text{MEMCS16}}$ ,  $\overline{\text{IRQ14}}$ , and  $\overline{\text{SBHE}}$  signals are reconfigured as the upper nibble LCD panel data.

Table 5-45 High-Speed PLL Frequency Select

HSPLLFQ 1 0	High-Speed PLL (CLK2) Frequency
0 0	40 MHz
0 1	50 MHz
1 0	66 MHz
1 1	(Reserved)

**Note:**

*In order to meet DRAM timing at 33 MHz, the following registers need to be programmed as indicated:*

*Clear bit 6 of the Command Delay register at Index 60h.*

*Set bit 4 of the MMS Memory Wait State 1 register at Index 62h.*

*Set bit 5 of the ROM Configuration 1 register at Index 65h.*

*Set bits 5 and 6 of the Wait State Control register at Index 63h.*

Table 5-46 RAM Mode Decode Logic

RAM Mode	Index B4h Bit 7	Index B1h Bits 7–6	Total Memory	Bank 0	Bank 1
	0	x x	Controlled by bits 4–2 of Index 66h		
256-Kbit×4-bit DRAM	1	0 0	512 Kbyte	512 Kbyte	–
256-Kbit×4-bit DRAM	1	0 1	1 Mbyte	512 Kbyte	512 Kbyte
Asymmetric 1-Mbit×16-bit DRAM	1	1 0	2 Mbyte	2 Mbyte	–
Asymmetric 1-Mbit×16-bit DRAM	1	1 1	4 Mbyte	2 Mbyte	2 Mbyte

**Notes:**

*If bit 7 of the PCMCIA Card Reset register at Index B4h is 1, bits 4–2 of the Memory Configuration 1 register at Index 66h are disabled. Bit 0 of the Memory Configuration 1 register determines page mode or enhanced page mode.*

*If the 4-Mbyte memory configuration is selected, the enhanced page mode must be selected by setting bit 0 of the Memory Configuration 1 register. Page mode is illegal for this configuration.*

### 5.3.111 PIRQ Configuration Register (Index B2h)

This register selects the IRQ level to which the PIRQ0 and PIRQ1 pins are connected. See Table 5-47 on page 5-80 for the valid IRQ selections. In Full ISA Bus mode, this register has no effect. This register is valid for Local Bus and Internal Video modes only. In Full ISA Bus mode, PIRQ0 is connected to IRQ3, and PIRQ1 is connected to IRQ6.

	7				0			
Field	PIRQ1 Steering				PIRQ0 Steering			
Bit	PIRQ1SL3	PIRQ1SL2	PIRQ1SL1	PIRQ1SL0	PIRQ0SL3	PIRQ0SL2	PIRQ0SL1	PIRQ0SL0
Default	0	0	0	0	0	0	0	0

Bit	Name	R/W	Function
7	PIRQ1SL3	R/W	PIRQ1 IRQ select, bit 3
6	PIRQ1SL2	R/W	PIRQ1 IRQ select, bit 2
5	PIRQ1SL1	R/W	PIRQ1 IRQ select, bit 1
4	PIRQ1SL0	R/W	PIRQ1 IRQ select, bit 0
3	PIRQ0SL3	R/W	PIRQ0 IRQ select, bit 3
2	PIRQ0SL2	R/W	PIRQ0 IRQ select, bit 2
1	PIRQ0SL1	R/W	PIRQ0 IRQ select, bit 1
0	PIRQ0SL0	R/W	PIRQ0 IRQ select, bit 0

**Note:**

Do not program either of the PIRQ pins such that they conflict with other IRQs. For example, do not program both PIRQ pins to the same level or to the IRQ level used by the internal UART. Interrupt sharing is not supported.

Table 5-47 Interrupt Redirect Logic

PIRQ1SL3–PIRQ1SL0 or PIRQ0SL3–PIRQ0SL0				PIRQ1 or PIRQ0
3	2	1	0	IRQ Selected
0	0	0	0	(None)
0	0	0	1	(Reserved)
0	0	1	0	(Reserved)
0	0	1	1	IRQ3
0	1	0	0	IRQ4
0	1	0	1	IRQ5
0	1	1	0	IRQ6
0	1	1	1	IRQ7
1	0	0	0	(Reserved)
1	0	0	1	IRQ9
1	0	1	0	IRQ10
1	0	1	1	IRQ11
1	1	0	0	IRQ12
1	1	0	1	(Reserved)
1	1	1	0	IRQ14
1	1	1	1	IRQ15

## 5.3.112 Miscellaneous 5 Register (Index B3h)

This register is used to obtain miscellaneous status information.

	7							0
Bit	32KHZSTE	ENFSTROMCS	FSTROMWS1	FSTROMWS0	ENSELREF	ENBROMCS	EXTSMISTE	(Reserved)
Default	0	0	0	0	0	0	0	0

Bit	Name	R/W	Function
7	32KHZSTE	R	State of the 32-kHz clock
6	ENFSTROMCS	R/W	1 = $\overline{ROMCS}$ ROM accesses enabled to run at the high-speed clock rate
5	FSTROMWS1	R/W	Fast BIOS ROM wait-state select 1
4	FSTROMWS0	R/W	Fast BIOS ROM wait-state select 0
3	ENSELREF	R/W	Self-refresh DRAM mode when CPUCLK is halted: 0 = CAS-before-RAS refresh (default) 1 = Self-refresh  This bit is not reset when exiting Micro Power Off mode.
2	ENBROMCS	R/W	1 = $\overline{ROMCS}$ enabled as an address decode, but not qualified with $\overline{MEMR}$ or $\overline{MEMW}$
1	EXTSMISTE	R	State of external SMI pin
0			(Reserved)

**Bit 6** This bit should not be set for systems that assert  $\overline{MCS16}$  because running the ISA bus at the high-speed PLL frequency violates  $\overline{MCS16}$  timing. When bit 6 is set, the Maximum ISA Bus signal, BALE, is not asserted for cycles to the  $\overline{ROMCS}$  decode space. This may cause ISA decode conflicts for devices that use BALE to catch LA23–LA17.

**Bits 5–4** When the  $\overline{ROMCS}$  ROM accesses are enabled to run at high speed (bit 6), these bits control the number of wait states for these cycles as shown in Table 5-48 on page 5-81.

**Bit 1** This bit indicates the state of the external SMI pin; it should be read to determine if an external SMI is being generated. The SMI handler should poll this bit and verify that it goes inactive prior to exiting the handler.

Table 5-48 ROM BIOS Enable and Wait-State Select Logic

ENFSTROMCS	FSTROMWS1–FSTROMWS0	Number of Wait States for $\overline{ROMCS}$ Cycle
0	x x	(Fast $\overline{ROMCS}$ disabled)
1	0 0	4
1	0 1	3
1	1 0	2
1	1 1	1

### 5.3.113 PCMCIA Card Reset Register (Index B4h)

This register is used to control miscellaneous PCMCIA functions and to enable DRAM modes.

	7							0
<b>Bit</b>	ENRAME2	(Reserved)	CBRESET	(Reserved)	(Reserved)	CARESET	(Reserved)	
<b>Default</b>	0	0	0	0	0	0	0	0

Bit	Name	R/W	Function
7	ENRAME2	R/W	1 = Enable additional DRAM modes corresponding to bits 7–6 of the Function Enable 2 register at Index B1h
6			(Reserved—must be 1)
5	CBRESET	W	Card B reset (RST_B)
4–3			(Reserved)
2	CARESET	W	Card A reset (RST_A)
1–0			(Reserved)

**Bit 7** When this bit is 0, RAM mode is determined by bits 4–2 of the Memory Configuration 1 register at Index 66h. Setting this bit enables additional DRAM modes configured via bits 7–6 of the Function Enable 2 register at Index B1h.

**Bit 5** Writing a 1 to this bit issues an RST\_B signal to Socket B. Software must write a 1 followed by a 0 to complete the reset command. The PCMCIA 2.1 standard requires RESET to be High for a minimum of 10 nsh.

**Bit 2** Writing a 1 to this bit issues an RST\_A signal to Socket A. Software must write a 1 followed by a 0 to complete the reset command. The PCMCIA 2.1 standard requires RESET to be High for a minimum of 10 ns.

### 5.3.114 CA24–CA25 Control 1 Register (Index B5h)

The CA24–CA25 Control registers control the CA24 and CA25 output pins. Using these registers, it is possible to map accesses to different pages to accesses anywhere in the PCMCIA 64-Mbyte memory map. The CA24–CA25 Control 1 register maps the output of PCMCIA pins 24 and 25 to MMSA pages 0–3.

		7						0
<b>Field</b>	MMSAPG3		MMSAPG2		MMSAPG1		MMSAPG0	
<b>Bit</b>	MMSAPG31	MMSAPG30	MMSAPG21	MMSAPG20	MMSAPG11	MMSAPG10	MMSAPG01	MMSAPG00
<b>Default</b>	0	0	0	0	0	0	0	0

Bit	Name	R/W	Function
7	MMSAPG31	R/W	MMSA page 3 CA25
6	MMSAPG30	R/W	MMSA page 3 CA24
5	MMSAPG21	R/W	MMSA page 2 CA25
4	MMSAPG20	R/W	MMSA page 2 CA24
3	MMSAPG11	R/W	MMSA page 1 CA25
2	MMSAPG10	R/W	MMSA page 1 CA24
1	MMSAPG01	R/W	MMSA page 0 CA25
0	MMSAPG00	R/W	MMSA page 0 CA24

## 5.3.115 CA24–CA25 Control 2 Register (Index B6h)

The CA24–CA25 Control registers control the CA24 and CA25 output pins. Using these registers, it is possible to map accesses to different pages to accesses anywhere in the PCMCIA 64-Mbyte memory map. The CA24–CA25 Control 2 register maps the output of PCMCIA pins 24 and 25 to MMSA pages 4–7.

	7				0			
<b>Field</b>	MMSAPG7		MMSAPG6		MMSAPG5		MMSAPG4	
<b>Bit</b>	MMSAPG71	MMSAPG70	MMSAPG61	MMSAPG60	MMSAPG51	MMSAPG50	MMSAPG41	MMSAPG40
<b>Default</b>	0	0	0	0	0	0	0	0

Bit	Name	R/W	Function
7	MMSAPG71	R/W	MMSA page 7 CA25
6	MMSAPG70	R/W	MMSA page 7 CA24
5	MMSAPG61	R/W	MMSA page 6 CA25
4	MMSAPG60	R/W	MMSA page 6 CA24
3	MMSAPG51	R/W	MMSA page 5 CA25
2	MMSAPG50	R/W	MMSA page 5 CA24
1	MMSAPG41	R/W	MMSA page 4 CA25
0	MMSAPG40	R/W	MMSA page 4 CA24

## 5.3.116 CA24–CA25 Control 3 Register (Index B7h)

The CA24–CA25 Control registers control the CA24 and CA25 output pins. Using these registers, it is possible to map accesses to different pages to accesses anywhere in the PCMCIA 64-Mbyte memory map. The CA24–CA25 Control 3 register maps the output of PCMCIA pins 24 and 25 to MMSB pages 0–3.

	7				0			
<b>Field</b>	MMSBPG3		MMSBPG2		MMSBPG1		MMSBPG0	
<b>Bit</b>	MMSBPG31	MMSBPG30	MMSBPG21	MMSBPG20	MMSBPG11	MMSBPG10	MMSBPG01	MMSBPG00
<b>Default</b>	0	0	0	0	0	0	0	0

Bit	Name	R/W	Function
7	MMSBPG31	R/W	MMSB page 3 CA25
6	MMSBPG30	R/W	MMSB page 3 CA24
5	MMSBPG21	R/W	MMSB page 2 CA25
4	MMSBPG20	R/W	MMSB page 2 CA24
3	MMSBPG11	R/W	MMSB page 1 CA25
2	MMSBPG10	R/W	MMSB page 1 CA24
1	MMSBPG01	R/W	MMSB page 0 CA25
0	MMSBPG00	R/W	MMSB page 0 CA24

### 5.3.117 ROM Configuration 3 Register (Index B8h)

This register controls the size of ROM DOS and the number of wait states for a  $\overline{\text{DOSCS}}$  cycle.

	7							0
Bit	ENFSTRDOS	FRDOSWS1	FRDOSWS0	ENRDOSCS	RDOSSIZ3	RDOSSIZ2	RDOSSIZ1	RDOSSIZ0
Default	0	0	0	0	0	0	0	0

Bit	Name	R/W	Function
7	ENFSTRDOS	R/W	1 = Enable $\overline{\text{DOSCS}}$ accesses to run at the high-speed CPU clock rate
6	FRDOSWS1	R/W	$\overline{\text{DOSCS}}$ wait state select bit 0
5	FRDOSWS0	R/W	$\overline{\text{DOSCS}}$ wait state select bit 1
4	ENRDOSCS	R/W	1 = Enable $\overline{\text{DOSCS}}$ as an address decode not qualified with $\overline{\text{MEMR}}$ or $\overline{\text{MEMW}}$
3	RDOSSIZ3	R/W	$\overline{\text{DOSCS}}$ size select bit 3
2	RDOSSIZ2	R/W	$\overline{\text{DOSCS}}$ size select bit 2
1	RDOSSIZ1	R/W	$\overline{\text{DOSCS}}$ size select bit 1
0	RDOSSIZ0	R/W	$\overline{\text{DOSCS}}$ size select bit 0

**Bit 7** This bit should not be set when decoding an 8-bit device in systems that assert  $\overline{\text{MCS16}}$  because running the ISA bus at the high-speed PLL frequency violates  $\overline{\text{MCS16}}$  timing. Bit 7 can be used if  $\overline{\text{DOSCS}}$  accesses a 16-bit device by setting bit 1 of the ROM Configuration 2 register at Index 51h. Also, when bit 7 is 1, the Maximum ISA Bus signal, BALE, is not asserted for cycles to the  $\overline{\text{DOSCS}}$  decode space. This may cause ISA decode conflicts for devices that use BALE to latch LA23–LA17.

**Bits 6–5** When the  $\overline{\text{DOSCS}}$  ROM accesses are enabled to run at high speed (bit 7), these bits control the number of wait states for these cycles as shown in Table 5-49 on page 5-84.

Table 5-49 ROM DOS Enable and Wait-State Select Logic

ENFSTRDOS	FRDOSWS1–FRDOSWS0	Number of Wait States for $\overline{\text{DOSCS}}$ Cycle
0	x x	(See Index 50h)
1	0 0	4
1	0 1	3
1	1 0	2
1	1 1	1

**Bits 3–0** These bits decode to the sizes shown in Table 5-50 on page 5-85.



Table 5-50 ROM DOS Linear Address Decode Size Select Logic

RDOSSIZ3–RDOSSIZ0	$\overline{\text{DOSCS}}$ Address Decode	DOS ROM Size
0 0 0 0	( $\overline{\text{DOSCS}}$ uses MMS mapping)	(NA)
0 0 0 1	F00000–FEFFFFh	1 Mbyte–64 Kbyte
0 0 1 0	E00000–FEFFFFh	2 Mbyte–64 Kbyte
0 0 1 1	D00000–FEFFFFh	3 Mbyte–64 Kbyte
0 1 0 0	C00000–FEFFFFh	4 Mbyte–64 Kbyte
0 1 0 1	B00000–FEFFFFh	5 Mbyte–64 Kbyte
0 1 1 0	A00000–FEFFFFh	6 Mbyte–64 Kbyte
0 1 1 1	900000–FEFFFFh	7 Mbyte–64 Kbyte
1 0 0 0	800000–FEFFFFh	8 Mbyte–64 Kbyte
1 0 0 1	700000–FEFFFFh	9 Mbyte–64 Kbyte
1 0 1 0	600000–FEFFFFh	10 Mbyte–64 Kbyte
1 0 1 1	500000–FEFFFFh	11 Mbyte–64 Kbyte
1 1 0 0	400000–FEFFFFh	12 Mbyte–64 Kbyte
1 1 0 1	300000–FEFFFFh	13 Mbyte–64 Kbyte
1 1 1 0	200000–FEFFFFh	14 Mbyte–64 Kbyte
1 1 1 1	100000–FEFFFFh	15 Mbyte–64 Kbyte

**Note:**

The linear  $\overline{\text{DOSCS}}$  decode range must not overlap the on-board memory decode range unless the on-board memory in the overlapping range is disabled via bits 7–4 of the Miscellaneous 1 register at Index 6Fh.

### 5.3.118 Memory Configuration 2 Register (Index B9h)

This register disables refresh in Off mode and selects programmable drive strengths, as shown in Table 5-51 on page 5-86 and Table 5-52 on page 5-86.

	7							0
<b>Bit</b>	DISREFOFF	(Reserved)	MEMDATS1	MEMDATS0	MEMADRS1	MEMADRS0	MEMCTLS1	MEMCTLS0
<b>Default</b>	0	0	0	0	0	0	0	0

Bit	Name	R/W	Function
7	DISREFOFF	R/W	Disable refresh in the PMU Off mode
6		R/W	(Reserved)
5	MEMDATS1	R/W	D15–D0 drive strength select 1
4	MEMDATS0	R/W	D15–D0 drive strength select 0
3	MEMADRS1	R/W	MA10–MA0/SA23–SA13 and $\overline{\text{MWE}}$ drive strength select 1
2	MEMADRS0	R/W	MA10–MA0/SA23–SA13 and $\overline{\text{MWE}}$ drive strength select 0
1	MEMCTLS1	R/W	$\overline{\text{RAS0}}$ , $\overline{\text{RAS1}}$ drive strength select 1
0	MEMCTLS0	R/W	$\overline{\text{RAS0}}$ , $\overline{\text{RAS1}}$ drive strength select 0

Table 5-51 Output Drive Strength Select Logic

Drive Strength Select Bit Value	Output Drive Strength
1 0	
0 0	E (default)
0 1	C
1 0	D
1 1	Three-state output

Table 5-52 I/O Drive Type Description

Drive Type	VCCIO (mA)	IOL <sub>TTL</sub> (mA)	IOH <sub>TTL</sub> (mA)
C	3.0–4.5	7.7–10.8	–8.6 to –34.2
D	3.0–4.5	7.7–10.8	–10.3 to –40.8
E	3.0–4.5	10.2–14.1	–13.6 to –53.9

**Note:**

Current out of a pin is given as a negative value.

### 5.3.119 Miscellaneous 3 Register (Index BAh)

This register enables PCMCIA MEM commands on parallel-port pins, enables the 14-MHz clock output on  $\overline{AFDT}$ , and preserves DRAM during Micro Power Off mode.

Bit	(Reserved)			EN_MCE_PP	EN_14M_PP	EN_MPOM	(Reserved)	
Default	0	0	0	0	0	0	0	0

Bit	Name	R/W	Function
7–5		R/W	(Reserved)
4	EN_MCE_PP	R/W	1 = Enable PCMCIA MEM commands on parallel-port pins 84 and 89.
3	EN_14M_PP	R/W	1 = Enable 14.336-MHz clock output on $\overline{AFDT}$
2	EN_MPOM	R/W	1 = Preserve DRAM during Micro Power Off mode
1–0		R/W	(Reserved)

**Bit 4** When this bit is 1, the PCMCIA output enable ( $\overline{PCMCOE}$ ) and write enable ( $\overline{PCMCWE}$ ) signals are enabled on pins 84 and 89, respectively.

**Bit 3** When this bit is 1, the 14.336-MHz clock output is enabled on pin 80.

**Bit 2** When this bit is 1, DRAM timing is maintained when entering Micro Power Off mode. It should be set if the contents of DRAM are to be preserved during Micro Power Off mode.

**Bits 1–0** These bits are not reset when exiting Micro Power Off mode.

Table 5-53 Parallel Port Pin Redefinition

Pin No.	Name	New Function
80	AFDT	14.336-MHz clock
84	SLCTIN	PCMCOE
89	INIT	PCMCWE





# A CONFIGURATION INDEX REGISTER REFERENCE

Configuration index registers are used to set up and monitor the system configuration of the ÉlanSC300 microcontroller. The following table lists the configuration index registers alphabetically by name. Use it when you know the name of a configuration index register but not its index.

For a complete description of each of these registers, see “Configuration Index Registers” on page 5-13. For information on how to access these registers, see “Configuration Register Overview” on page 5-2.

Configuration Index Register	Index	Function
Activity Mask 1	75h	PMU
Activity Mask 2	76h	PMU
Activity Status 1	A0h	PMU
Activity Status 2	A1h	PMU
Auto Low-Speed Control	9Fh	PMU
CA24–CA25 Control 1	B5h	PCMCIA
CA24–CA25 Control 2	B6h	PCMCIA
CA24–CA25 Control 3	B7h	PCMCIA
Clock Control	8Fh	PMU
Command Delay	60h	ISA/MCU
Control A	48h	Control
Control B	77h	Control
CPU Status 0	A3h	PMU
CPU Status 1	A4h	PMU
Doze to Sleep Mode Timer	85h	PMU
Drive Timer	47h	PMU
Function Enable 1	B0h	PMU
Function Enable 2	B1h	PMU
General-Purpose I/O 0	89h	PMU
General-Purpose I/O 1	9Ch	PMU
General-Purpose I/O 2	94h	PMU
General-Purpose I/O 3	95h	PMU
General-Purpose I/O Control	91h	PMU
High-Speed to Low-Speed Mode Timer	83h	PMU
I/O Activity Address 0	8Ch	PMU
I/O Activity Address 1	8Dh	PMU
I/O Timeout	40h	PMU
I/O Wait State	61h	ISA/MCU
Low-Speed to Doze Mode Timer	84h	PMU
Memory Configuration 1	66h	MCU
Memory Configuration 2	B9h	MMU

(Continued)

Configuration Index Register	Index	Function
Memory Write Activity Lower Boundary	9Ah	PMU
Memory Write Activity Upper Boundary	9Bh	PMU
Miscellaneous 1	6Fh	ISA/MCU
Miscellaneous 2	6Bh	MCU
Miscellaneous 3	BAh	MMU/PMU
Miscellaneous 4	44h	PMU
Miscellaneous 5	B3h	PMU
Miscellaneous 6	70h	PMU/MCU
MMS Address	6Dh	MCU
MMS Address Extension 1	6Ch	MCU
MMS Address Extension 2	6Eh	MCU
MMS Memory Wait State 1	62h	MCU
MMS Memory Wait State 2	50h	MCU
MMSA Address Extension 1	67h	MCU
MMSA Device 1	71h	MCU
MMSA Device 2	72h	MCU
MMSA Socket	A8h	MCU
MMSB Control	74h	PMU/MCU
MMSB Device	73h	MCU
MMSB Socket	A9h	PMU/MCU
NMI/SMI Control	A5h	PMU
NMI/SMI Enable	82h	PMU
PCMCIA Card Reset	B4h	PCMCIA
PCMCIA Data Width	0Ah	PCMCIA
PCMCIA I/O Card IRQ Redirection Control A	06h	PCMCIA
PCMCIA I/O Card IRQ Redirection Control B	16h	PCMCIA
PCMCIA I/O Window A1 Lower Byte End	01h	PCMCIA
PCMCIA I/O Window A1 Lower Byte Start	00h	PCMCIA
PCMCIA I/O Window A1 Upper Byte	02h	PCMCIA
PCMCIA I/O Window A2 Lower Byte End	04h	PCMCIA
PCMCIA I/O Window A2 Lower Byte Start	03h	PCMCIA
PCMCIA I/O Window A2 Upper Byte	05h	PCMCIA
PCMCIA I/O Window B1 Lower Byte End	11h	PCMCIA
PCMCIA I/O Window B1 Lower Byte Start	10h	PCMCIA
PCMCIA I/O Window B1 Upper Byte	12h	PCMCIA
PCMCIA I/O Window B2 Lower Byte End	14h	PCMCIA
PCMCIA I/O Window B2 Lower Byte Start	13h	PCMCIA
PCMCIA I/O Window B2 Upper Byte	15h	PCMCIA
PCMCIA REGA Address	8Ah	PCMCIA
PCMCIA REGB Address	9Eh	PCMCIA
PCMCIA Socket A Status	A2h	PCMCIA
PCMCIA Socket B Status	0Ch	PCMCIA
PCMCIA Status Change	A6h	PCMCIA
PCMCIA Status Change IRQ Enable	0Dh	PCMCIA
PCMCIA Status Change IRQ Redirection	0Eh	PCMCIA
PCMCIA VPPA Address	07h	PCMCIA
PCMCIA VPPB Address	17h	PCMCIA

*(Continued)*

<b>Configuration Index Register</b>	<b>Index</b>	<b>Function</b>
PIO Address	45h	PMU
PIO Timer	46h	PMU
PIRQ Configuration	B2h	PMU
PMU Control 1	A7h	PMU
PMU Control 2	AFh	PMU
PMU Control 3	ADh	PMU
Power Control 1	80h	PMU
Power Control 2	81h	PMU
Power Control 3	ABh	PMU
Power Control 4	ACH	PMU
Resume Mask	08h	PMU
Resume Status	09h	PMU
ROM Configuration 1	65h	MCU
ROM Configuration 2	51h	MCU
ROM Configuration 3	B8h	MMU
Shadow RAM Enable 1	68h	MCU
Shadow RAM Enable 2	69h	MCU
Sleep to Suspend Mode Timer	86h	PMU
SMI Enable	41h	PMU
SMI I/O Status	42h	PMU
SMI MMS Page	AAh	PMU
SMI Status	43h	PMU
Software Mode Control	88h	PMU
Suspend to Off Mode Timer	87h	PMU
UART Clock Enable	92h	PMU
Version	64h	PMU
Wait State Control	63h	MCU





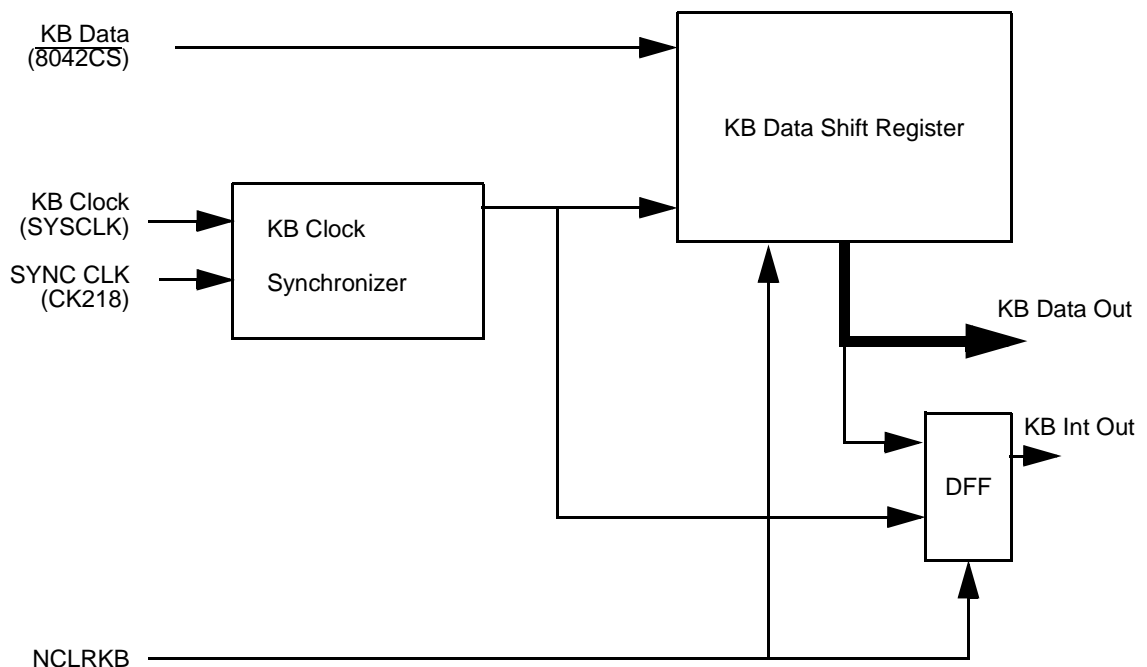


# B XT-KEYBOARD INTERFACE

The XT-keyboard interface consists of clock and data inputs to the ÉlanSC300 microcontroller and is compatible with the IBM PC/XT keyboard. One of the ÉlanSC300 microcontroller's output pins, such as a Programmable General-Purpose (PGP) pin, may be used to drive an actual XT-keyboard reset input.

The KB clock input is synchronized by two serial flip-flops, which are clocked by the CK218 signal. This signal is actually the CPU clock signal divided by 6. The (thus delayed) KB clock signal is then used to clock the data into the KB Data Shift register. The first bit to be clocked in is the start bit, and must be a logic 1 (High). Eight more bits are then shifted in, beginning with the least significant bit. On the ninth clock, the start bit is shifted into the DFF, which drives the KB interrupt output and is connected to IRQ1. At this same time, the KB Data I/O pad changes directions to become an output and is driven Low as a busy indication to the keyboard or other driving device. At this time, the host should respond to the interrupt and read the byte assembled in the KB Data Shift register at port 060h. The host also has the option of driving the clock pin Low as an additional handshake indication. After the host has read the byte from port 060h, the host clears the KB Data Shift register and the interrupt flip-flop by writing a 1 and then a 0 to bit 7 of port 061h. This action not only clears the shift register and interrupt, but also releases the data line to function as an input again.

Figure B-1 XT Keyboard Block Diagram



## B.1 XT KEYBOARD ENABLE

The ÉlanSC300 microcontroller's XT keyboard circuitry is enabled by bit 3 of Index 0ADh. Setting this bit

- Allows the  $\overline{8042CS}$  and SYSCLK pins to become three-stated so that they may be used as inputs (all other requirements being met).
- Qualifies an internal decode so that port 060h is read as an internal port.
- Switches a multiplexer to vector IRQ1 from the external pin to the output of this circuitry.

## B.2 KEYBOARD INTERFACE CONTROL

Two bits are provided for control of the XT keyboard interface. These bits are located at port 061h. Bit 7 is used to clear the keyboard interrupt and shift register, a flip-flop that enables the data line as an output (to act as a busy signal to the keyboard when Low). Two writes are required for the proper operation of this bit—one to set it, and another to clear it. If bit 7 is not cleared, the shift register is held in a *clear* configuration. Bit 6, when High, makes the KBCLK line an output (driving it Low), which can also be used as a busy signal to the keyboard.

## B.3 KEYBOARD DATA PORT

Once a serial keyboard byte has been assembled, it can be read at port 060h.

## B.4 I/O MAP SUMMARY

- Setting bit 3 of Index ADh enables the XT keyboard.
- Port 060h contains the keyboard data.
- Setting bit 7 of Port 061h clears the keyboard shift register and interrupt.
- Setting bit 6 of Port 061h forces the keyboard clock Low.

## B.5 PINS USED

The XT keyboard option uses the following pins:

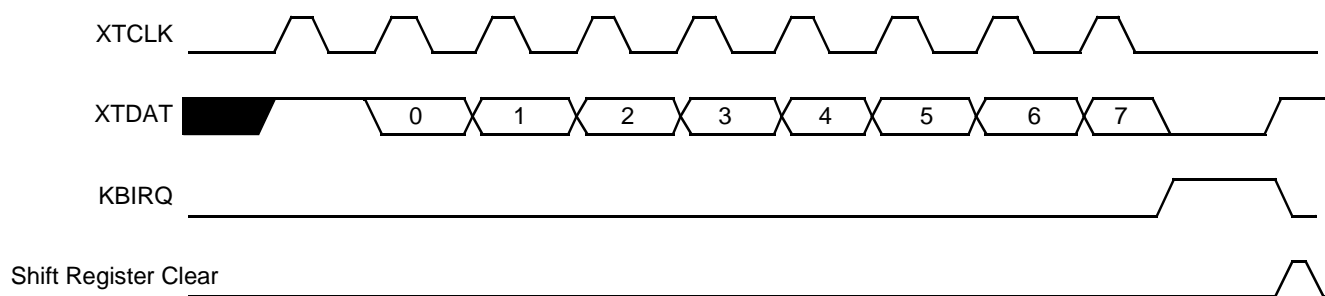
- **SYSCLK** Becomes the keyboard clock (XTCLK).
- **8042CS** Becomes the keyboard data (XTDAT).

Because both pins can also act as outputs, and the data line is driven as an output at the end of each transferred byte, the pins must be driven by open-drain or open-collector drivers with external pull-ups.

## B.6 TIMING

The XT-keyboard clock runs at approximately 100 kHz, or 10  $\mu$ s per bit. The falling edge of the XTCLK input clocks the shift register, but it is delayed by two CPUCLK periods divided by 6. Therefore, XTDAT should be changed on the rising edge of the XTCLK signal. The XT-keyboard interface runs at speeds as high as 250 kHz.

Figure B-2 XT Keyboard Timing Diagram







# INDEX

## Numbers

0CLK\_DOZ bit, 5-55  
0CLK\_SLP bit, 5-56  
0CLK\_SUS bit, 5-56  
8-bit ISA I/O access command delay (table), 5-33  
8-bit ISA I/O memory access command delay (table), 5-33  
8BMWS0 bit  
    8-bit ISA memory-cycle wait states (table), 5-36  
    function, 5-35  
8BMWS1 bit  
    8-bit ISA memory-cycle wait states (table), 5-36  
    function, 5-35  
8MCD bit  
    function, 5-32  
    PCMCIA command delay select logic (table), 5-31  
    ROM DOS command delay select logic (table), 5-31  
16BMWS0 bit  
    16-bit ISA memory-cycle wait states (table), 5-36  
    function, 5-35  
16BMWS1 bit  
    16-bit ISA memory-cycle wait states (table), 5-36  
    function, 5-35  
16IOWAIT bit, 5-37  
24/12 bits, 4-17  
32KHZSTE bit, 5-81  
80-ns DRAM support, 2-26  
6845 video index registers. *See* video index registers, standard.  
8042CS pin, for XT-keyboard interface, B-2  
8254 clock (timer) speeds (table), 1-6  
16450 clock (UART) speeds (table), 1-6

## A

A0 bit, 5-26  
A2 bit, 5-27  
A3 bit, 5-27

A4 bit, 5-27  
A5 bit, 5-27  
A6 bit, 5-27  
A7 bit, 5-27  
A8 bit, 5-27  
A9 bit, 5-27  
A20SMI bit, 5-43  
AC Input Status pin, 1-34  
ACIN bit  
    Activity Mask 1 Register (Index 75h), 5-53  
    Activity Status 1 Register (Index A0h), 5-67  
    CPU Status 1 Register (Index A4h), 5-69  
ACIN pin, 1-34  
 $\overline{\text{ACK}}$  bit  
    Parallel Status Port (Ports 279h & 379h), 4-8  
    Parallel Status Port (Ports 279h, 379h & 3BDh), 4-7  
ACKSTAT bit, 4-8  
activity  
    CPU and non-CPU related, 1-19  
    events constituting activity, 1-10  
    inactivity states and transition intervals (table), 1-10  
Activity Mask 1 Register (Index 75h)  
    activity selection, 1-19  
    bit descriptions, 5-53–5-54  
    generating PMU-activity event with ACIN pin, 1-34  
    initializing for suspend/resume operation, 5-56  
    PMU operating-mode transitions, 1-8  
Activity Mask 2 Register (Index 76h)  
    activity selection, 1-19  
    bit descriptions, 5-54  
    PMU operating-mode transitions, 1-8  
activity monitors  
    definition, 1-2  
    purpose and use, 1-19  
    state transition flowchart (figure), 1-20–1-21  
Activity Status 1 Register (Index A0h)  
    activity monitoring, 1-19  
    bit descriptions, 5-67  
Activity Status 2 Register (Index A1h)  
    activity monitoring, 1-19  
    bit descriptions, 5-67

address registers

- CGA Index Address register (Port 3D4h), 3-23
- Configuration Address Register (Port 022h), 5-12
- Cursor Address Registers (Video Indexes 0E-0Fh), 3-27
- HGA Index Address register (Port 3B4h), 3-21
- I/O Activity Address 0 Register (Index 8Ch), 5-59
- I/O Activity Address 1 Register (Index 8Dh), 1-8, 5-60
- MMS Address Extension 1 Register (Index 6Ch), 5-44
- MMS Address Extension 2 Register (Index 6Eh), 5-46
- MMS Address Register (Index 6Dh), 5-44–5-46
- MMSA Address Extension 1 Register (Index 67h), 5-42
- NMI/RTC Index Address Register (Port 070h), 4-19
- Parallel EPP Address Port (Ports 27Bh & 37Bh), 4-9
- PCMCIA REGA Address Register (Index 8Ah), 5-59
- PCMCIA REGB Address Register (Index 9Eh), 5-66
- PCMCIA VPPA Address Register (Index 07h), 5-15
- PCMCIA VPPB Address Register (Index 17h), 5-24
- PIO Address Register (Index 45h), 5-27
- Start Address Registers (Video Indexes 0C-0Dh), 3-27

AF bit, 4-17

AICBL1CHG bit, 5-70–5-71

AICBL2CHG bit, 5-70–5-71

AIE bit, 4-17

all-points-addressable (APA) mode. *See* graphics mode.

ALTA20 bit, 4-19

ALTBK bit, 3-25

ALTPAL bit, 3-25

Am386SXLV microprocessor, integration in ÉlanSC300 microcontroller, xvii

APA (all-points-addressable) mode. *See* graphics mode.

ARDYCHG bit, 5-70–5-71

ASB. *See* auto screen blanking (ASB).

AT peripheral registers. *See* PC/AT peripheral registers.

attribute byte

- CGA attribute byte, 3-9–3-10
  - background-color bit values (table), 3-10
  - foreground-color bit values (table), 3-10
- HGA attribute byte, 3-11
  - bit values (table), 3-11
  - purpose and use, 3-9

AUTBLNK bit, 3-30

AUTLOW bit, 5-55

Auto Low-Speed Control Register (Index 9Fh)

- bit descriptions, 5-66
- low-speed duration period select logic (table), 5-67
- trigger period select logic (table), 5-66

Auto Low-Speed logic

- overview, 1-3
- purpose and use, 1-40

auto screen blanking (ASB)

- description, 3-32
- timer setting (table), 3-32

AUTOFDX bit, 4-7

## B

BAK2-BAK0 bits

- CGA attribute byte, 3-9
- CGA background-color bit values (table), 3-10
- HGA attribute byte, 3-11
- bit values (table), 3-11

battery-management logic, 1-32–1-34

- AC Input Status (ACIN) pin, 1-34
- Battery Level 1 ( $\overline{BL1}$ ) pin, 1-32
- Battery Level 2 ( $\overline{BL2}$ ) pin, 1-32–1-33
- Battery Level 3 ( $\overline{BL3}$ ) pin, 1-33
- Battery Level 4 ( $\overline{BL4}$ ) pin, 1-33–1-34
- functionality of BL4-BL1 (table), 1-32
- overview, 1-2

BHE bit, 5-26

BI bit, 4-12

BICBL1CHG bit, 5-70–5-71

BICBL2CHG bit, 5-70–5-71

BKMISS bit

- DRAM bank miss wait state select logic (table), 5-37
- function, 5-37

BL1 bit

- NMI/SMI Enable Register (Index 82h), 5-56
- PCMCIA Status Change Register (Index A6h), 5-70

$\overline{BL1}$  pin, 1-32

BL1IN bit, 5-68

BL1LOWSP bit, 5-77

BL2 bit

- NMI/SMI Enable Register (Index 82h), 5-56
- PCMCIA Status Change Register (Index A6h), 5-70

$\overline{BL2}$  pin

- battery-management logic, 1-32–1-33
- SMI generation, 1-33

BL2IN bit, 5-68

- BL3 bit
    - NMI/SMI Enable Register (Index 82h), 5-56
    - PCMCIA Status Change Register (Index A6h), 5-70
  - $\overline{\text{BL3}}$  pin, 1-33
  - BL3IN bit, 5-68
  - $\overline{\text{BL4}}$  pin, 1-33–1-34
  - BLINK bit
    - CGA attribute byte, 3-9
    - HGA attribute byte, 3-11
  - BLNKTIM1-BLNKIM0 bits, 3-31
  - BRDYCHG bit, 5-70–5-71
  - bulletin board support, iii
  - bus
    - bus option status table, 5-41
    - determining bus configuration (table), 5-4
    - general bus I/O wait states (table), 5-35
    - PC/AT bus, 5-4
    - quiet bus feature, 1-41
  - bus configurations, 2-6–2-7
    - ÉlanSC300 microcontroller bus configurations (table), 2-7
    - ISA, local bus, and internal LCD configurations, 2-6–2-7
    - pins for bus configuration (table), 2-6
    - supported by ÉlanSC300 microcontroller, 2-3
  - $\overline{\text{BUSY}}$  bit
    - Parallel Status Port (Ports 279h & 379h), 4-8
    - Parallel Status Port (Ports 279h, 379h & 3BDh), 4-7
  - BVD1\_A signal
    - description (table), 1-23
    - false wake-ups, 1-24
  - BVD1\_B signal
    - description (table), 1-23
    - false wake-ups, 1-24
  - BVDA1 bit, 5-68
  - BVDA2 bit, 5-68
  - BVDB1 bit, 5-19
  - BVDB2 bit, 5-19
- ## C
- CA7-CA0 bits, 3-9
  - CA24-CA25 Control 1 Register (Index B5h), 5-82
  - CA24-CA25 Control 2 Register (Index B6h), 5-83
  - CA24-CA25 Control 3 Register (Index B7h), 5-83
  - CARDCMD bit (table), 5-32
  - CARDCMDL bit, 5-30
  - CARDWS0 bit
    - function, 5-30
    - PCMCIA wait state select logic (table), 5-31
  - CARDWS1 bit
    - function, 5-30
    - PCMCIA wait state select logic (table), 5-31
  - CARDWSEN bit
    - function, 5-33
    - PCMCIA command delay select logic (table), 5-32
    - PCMCIA wait state select logic (table), 5-31
  - CARESET bit, 5-82
  - $\overline{\text{CAS0H}}$  signal
    - selecting DRAM memory banks, 2-3
    - specifying SRAM accesses, 2-4
  - $\overline{\text{CAS0L}}$  signal
    - selecting DRAM memory banks, 2-3
    - specifying SRAM accesses, 2-4
  - $\overline{\text{CAS1H}}$  signal, 2-4
  - $\overline{\text{CAS1L}}$  signal
    - selecting DRAM memory banks, 2-3
    - specifying SRAM accesses, 2-4
  - CBRESET bit, 5-82
  - CD bit, 5-68
  - CDB bit, 5-19
  - central processing unit. *See* CPU.
  - CFG0 bit
    - bus option status table, 5-41
    - function, 5-41
  - CFG1 bit
    - bus option status table, 5-41
    - function, 5-41
  - CGA attribute byte
    - background-color bit values (table), 3-10
    - bit descriptions, 3-9
    - foreground-color bit values (table), 3-10
  - CGA Color Select register (Port 3D9h)
    - bit descriptions, 3-25
    - screen-background color-bit logic (table), 3-25
  - CGA colors, mapping to gray scales (table), 3-30
  - CGA Index Address register (Port 3D4h), 3-23
  - CGA Index Data register (Port 3D5h), 3-23
  - CGA Mode Control register (Port 3D8h), 3-24
  - CGA Status register (Port 3DAh), 3-26

- CGA text mode
  - character fonts 1 area (table), 3-7
  - character fonts 2 area (table), 3-8
  - special character fonts area (table), 3-8
- CGATXTF bit, 3-30
- character byte, text mode, 3-9
- CHGFUSET bit, 5-77
- CHGON bit, 5-77
- Clock Control Register (Index 8Fh)
  - bit descriptions, 5-60
  - PLL restart time select logic (table), 5-61
- clock sources, PMU, 1-9
- clock-switching logic, 1-16–1-18
  - clock startup and shutdown logic, 1-17
  - logic flowchart (figure), 1-18
  - CPU/memory clock switching, 1-16–1-17
  - definition, 1-2
- clocks. *See also* specific clocks.
  - clock-switching permutations available with PMU modes, 1-8
  - keyboard clock, B-2
  - PMU clock speeds (table), 1-6
  - registers for controlling (table), 5-9
- CLRMPEN bit, 3-31
- COLBUR bit, 3-24
- COLEM bit, 3-30
- Color Mapping Registers (Video Index 14-17h & 1C-1Fh)
  - description, 3-29
  - gray-scale mapping (table), 3-29
  - mapping CGA colors to gray scales (table), 3-30
- COM bit
  - Activity Mask 2 Register (Index 76h), 5-54
  - Activity Status 2 Register (Index A1h), 5-67
- Command Delay Register (Index 60h)
  - 8-bit ISA I/O access command delay (table), 5-33
  - 8-bit ISA memory access command delay (table), 5-33
  - bit descriptions, 5-32–5-33
  - refresh cycle wait states (table), 5-33
  - ROM BIO wait states (table), 5-33
- command delays
  - duration for various cycles (table), 2-24
  - purpose and use, 2-23
- Configuration Address Register (Port 022h), 5-12
- Configuration Data Register (Port 023h), 5-12
- configuration index registers
  - Activity Mask 1 Register (Index 75h), 1-8, 1-19, 1-34, 5-53–5-54
  - Activity Mask 2 Register (Index 76h), 1-8, 1-19, 5-54
  - Activity Status 1 Register (Index A0h), 1-19, 5-67
  - Activity Status 2 Register (Index A1h), 1-19, 5-67
  - Auto Low-Speed Control Register (Index 9Fh), 5-66
  - CA24-CA25 Control 1 Register (Index B5h), 5-82
  - CA24-CA25 Control 2 Register (Index B6h), 5-83
  - CA24-CA25 Control 3 Register (Index B7h), 5-83
  - Clock Control Register (Index 8Fh), 5-60–5-61
  - Command Delay Register (Index 60h), 5-32–5-33
  - Control A Register (Index 48h), 5-30
  - Control B Register (Index 77h), 5-55
  - controlling low-speed CPU clock, 1-6
  - CPU Status 0 Register (Index A3h), 1-32, 5-68–5-69
  - CPU Status 1 Register (Index A4h), 1-9, 5-69
  - Doze to Sleep Timer Register (Index 85h), 1-22, 5-57
  - Drive Timer Register (Index 47h), 5-29
  - Function Enable 1 Register (Index B0h), 1-31, 5-77–5-78
  - Function Enable 2 Register (Index B1h), 2-3, 5-78–5-79
  - General-Purpose I/O 1 Register (Index 9Ch), 5-65
  - General-Purpose I/O 2 Register (Index 94h), 1-7, 5-63
  - General-Purpose I/O 3 Register (Index 95h), 1-7, 5-63
  - General-Purpose I/O Control Register (Index 91h), 1-7, 5-61–5-62
  - General-Purpose I/O Register (Index 89h), 5-59
  - High-Speed to Low-Speed Timer Register (Index 83h), 1-22, 5-57
  - I/O Activity Address 0 Register (Index 8Ch), 1-8, 5-59
  - I/O Activity Address 1 Register (Index 8Dh), 1-8, 5-60
  - I/O Timeout Register (Index 40h), 1-31, 5-24–5-25
  - I/O Wait State Register (Index 61h), 5-34–5-35
  - Low-Speed to Doze Timer Register (Index 84h), 1-22, 5-57
  - mandatory settings, 5-2–5-3
  - Memory Configuration 1 Register (Index 66h), 2-3, 2-5, 5-41
  - Memory Configuration 2 Register (Index B9h), 1-7, 5-85–5-86
  - Memory Write Activity Lower Boundary Register (Index 9Ah), 1-8, 5-64
  - Memory Write Activity Upper Boundary Register (Index 9Bh), 1-8, 5-64–5-65
  - Miscellaneous 1 Register (Index 6Fh), 2-10, 5-47–5-48
  - Miscellaneous 2 Register (Index 6Bh), 1-27, 5-43
  - Miscellaneous 3 Register (Index BAh), 5-86–5-87
  - Miscellaneous 4 Register (Index 44h), 1-29, 1-41, 5-27



- Miscellaneous 5 Register (Index B3h), 1-28, 2-25, 5-81
- Miscellaneous 6 Register (Index 70h), 1-34, 5-48
- MMS Address Extension 1 Register (Index 6Ch), 5-44
- MMS Address Extension 2 Register (Index 6Eh), 5-46
- MMS Address Register (Index 6Dh), 5-44–5-46
- MMS Memory Wait State 1 Register (Index 62h), 2-5, 5-35–5-36
- MMS Memory Wait State 2 Register (Index 50h), 5-30–5-32
- MMSA Address Extension 1 Register (Index 67h), 5-42
- MMSA Device 1 Register (Index 71h), 5-49
- MMSA Device 2 Register (Index 72h), 5-50
- MMSA Socket Register (Index A8h), 5-72
- MMSB Control Register (Index 74h), 1-6, 1-33, 5-52
- MMSB Device Register (Index 73h), 5-51
- MMSB Socket Register (Index A9h), 1-27, 5-56, 5-73
- NMI/SMI Control Register (Index A5h), 1-26–1-28, 1-32, 1-33, 1-36, 1-38, 1-39, 5-70
- NMI/SMI Enable Register (Index 82h), 1-32, 1-33, 1-37, 5-56
- overview, 5-2
- PCMCIA Card Reset Register (Index B4h), 2-3, 5-82
- PCMCIA Data Width Register (Index 0Ah), 5-17–5-18
- PCMCIA I/O Card IRQ Redirection Control A Register (Index 06h), 5-14–5-15
- PCMCIA I/O Card IRQ Redirection Control B Register (Index 16h), 5-23–5-24
- PCMCIA I/O Window A1 Lower Byte End Register (Index 01h), 5-13
- PCMCIA I/O Window A1 Lower Byte Start Register (Index 00h), 5-13
- PCMCIA I/O Window A1 Upper Byte End Register (Index 02h), 5-13
- PCMCIA I/O Window A2 Lower Byte End Register (Index 04h), 5-14
- PCMCIA I/O Window A2 Lower Byte Start Register (Index 03h), 5-13
- PCMCIA I/O Window A2 Upper Byte Register (Index 05h), 5-14
- PCMCIA I/O Window B1 Lower Byte End Register (Index 11h), 5-22
- PCMCIA I/O Window B1 Lower Byte Start Register (Index 10h), 5-21
- PCMCIA I/O Window B1 Upper Byte End Register (Index 12h), 5-22
- PCMCIA I/O Window B2 Lower Byte End Register (Index 14h), 5-22
- PCMCIA I/O Window B2 Lower Byte Start Register (Index 13h), 5-22
- PCMCIA I/O Window B2 Upper Byte Register (Index 15h), 5-23
- PCMCIA REGA Address Register (Index 8Ah), 5-59
- PCMCIA REGB Address Register (Index 9Eh), 5-66
- PCMCIA Socket A Status Register (Index A2h), 1-28, 1-32, 1-33, 1-36, 1-38, 1-39, 2-21, 5-68
- PCMCIA Socket B Status Register (Index 0Ch), 5-19
- PCMCIA Status Change IRQ Enable Register (Index 0Dh), 2-21, 5-19–5-20
- PCMCIA Status Change IRQ Redirection Register (Index 0Eh), 5-20–5-21
- PCMCIA Status Change Register (Index A6h), 1-28, 2-21, 5-70–5-71
- PCMCIA VPPA Address Register (Index 07h), 5-15
- PCMCIA VPPB Address Register (Index 17h), 5-24
- PIO Address Register (Index 45h), 1-29, 5-27
- PIO Timer Register (Index 46h), 1-29, 5-27–5-28
- PIRQ Configuration Register (Index B2h), 5-79–5-80
- PMU Control 1 Register (Index A7h), 1-8, 1-41, 5-71
- PMU Control 2 Register (Index AFh), 1-22, 1-32, 5-77–5-78
- PMU Control 3 Register (Index ADh), 1-8, 1-34, 5-76
- PMU operating-mode transitions, 1-8
- Power Control 1 Register (Index 80h), 1-6, 1-9, 5-55
- Power Control 2 Register (Index 81h), 1-7, 1-9, 5-56
- Power Control 3 Register (Index ABh), 5-74
- Power Control 4 Register (Index ACh), 5-75
- recommended settings, 5-3
- reference list in alphabetical order (table), A-1–A-3
- Reserved Register (Index 0Bh), 5-16
- Reserved Register (Index 0Fh), 5-21
- Reserved Register (Index 6Ah), 5-43
- Reserved Register (Index 8Bh), 5-59
- Reserved Register (Index 8Eh), 5-60
- Reserved Register (Index 9Dh), 5-66
- Reserved Register (Index 90h), 5-61
- Reserved Register (Index 93h), 5-63
- Reserved Register (Index AEh), 5-76
- Reserved Registers (Index 52h-5Fh), 5-32
- Reserved Registers (Index 78-7Fh), 5-55
- Reserved Registers (Index 96-99h), 5-64
- Reserved Registers (Indexes 28-39h), 5-24
- Reserved Registers (Indexes 49-4Fh), 5-30
- Resume Mask Register (Index 08h), 1-19, 5-16
- Resume Status Register (Index 09h), 1-19, 1-24–1-25, 5-17
- ROM Configuration 1 Register (Index 65h), 2-5, 2-9, 5-40
- ROM Configuration 2 Register (Index 51h), 2-9, 5-32
- ROM Configuration 3 Register (Index B8h), 2-10, 2-25, 5-84–5-85
- Shadow RAM Enable 1 Register (Index 68h), 5-42
- Shadow RAM Enable 2 Register (Index 69h), 5-43
- Sleep to Suspend Timer Register (Index 86h), 1-22, 1-37, 5-57
- SMI Enable Register (Index 41h), 1-30, 1-31, 5-25
- SMI I/O Status Register (Index 42h), 1-30, 5-26
- SMI MMS Page Register (Index AAh), 1-27, 5-73
- SMI Status Register (Index 43h), 1-28, 5-26

- Software Mode Control Register (Index 88h), 1-29, 5-58
  - Suspend to Off Timer Register (Index 87h), 1-22, 5-58
  - UART Clock Enable Register (Index 92h), 5-63
  - Version Register (Index 64h), 5-38–5-39
  - Wait State Control Register (Index 63h), 2-5, 5-37–5-38
  - configuration port registers
    - Configuration Address Register (Port 022h), 5-12
    - Configuration Data Register (Port 023h), 5-12
    - overview, 5-2
  - configuring ÉlanSC300 microcontroller
    - clocks (phase-locked loops), 5-9
    - CPU and PC/AT compatibility (table), 5-5
    - determining bus configuration (table), 5-4
    - direct memory accesses (table), 5-5
    - general-purpose and PMC pins, 5-9
    - interrupts and their mapping (table), 5-5
    - memory mapping (MMS windows) (table), 5-6
    - parallel port (table), 5-7
    - PC/AT bus and its timing (table), 5-4
    - PCMCIA card interface (table), 5-7–5-8
    - power management activities and events (table), 5-10
    - power management state timers (table), 5-11
    - power management status (table), 5-10
    - ROM accesses and ROM cycles (table), 5-11
    - SMI status (table), 5-11
    - speed of CPU (table), 5-5
    - system DRAM (table), 5-6
    - system SRAM (table), 5-7
    - UART (table), 5-7
    - video display (table), 5-12
  - Control 1 Register (Video Index 20h)
    - bit descriptions, 3-34
    - printer I/O port select logic (table), 3-34
  - Control A Register (Index 48h), 5-30
  - Control B Register (Index 77h), 5-55
  - control logic for PCMCIA controller, 2-20
  - CP1/HDRV signal (table), 3-2
  - CP2/VDO signal (table), 3-2
  - CPU and PC/AT compatibility, controlling (table), 5-5
  - CPU clock, low-speed, controlling, 1-6
  - CPU speed, controlling (table), 5-5
  - CPU Status 0 Register (Index A3h)
    - bit descriptions, 5-68
    - last PMU mode indicator bits (table), 5-73
    - reading status of  $\overline{BL2}$  pin, 1-32
  - CPU Status 1 Register (Index A4h)
    - bit descriptions, 5-69
    - present PMU mode indicator bits (table), 5-69
    - reading PMU mode, 1-9
  - CPUCLK
    - High-Speed PLL mode, 1-4
    - Low-Speed PLL mode, 1-5
    - PMU clock speeds (table), 1-6
  - CPU\_IDLE bit, 5-43
  - CTS bit, 4-13
  - Cursor Address Registers (Video Indexes 0E-0Fh), 3-27
  - Cursor Start and End Registers (Video Indexes 0A-0Bh), 3-27
  - customer service, iii
- ## D
- data-path disabling logic, 1-41
  - $\overline{DBUFOE}$  signal, 5-77
  - DCTS bit, 4-13
  - DDSR bit, 4-13
  - device-power-down logic
    - description, 1-29–1-31
    - SMI device-powerdown flowchart (figure), 1-30
  - DIR bit
    - Parallel Control Port (Ports 27Ah & 37Ah), 4-8
    - Parallel Control Port (Ports 27Ah, 37Ah & 37Eh), 4-7
  - direct memory access, controlling (table), 5-5
  - Disable Software Switch Register (Video Index 13h)
    - description, 3-28
    - disabling extended video-controller registers (table), 3-28
  - DISCMD bit, 5-30
  - DISDEN bit, 5-27
  - disk drive management registers
    - Drive Timer Register (Index 47h), 5-29
    - I/O Timeout Register (Index 40h), 5-24–5-25
    - I/O Wait State Register (Index 61h), 5-34–5-35
      - floppy disk drive wait states (table), 5-34
      - hard disk drive wait states (table), 5-35
    - SMI Enable Register (Index 41h), 5-25
    - SMI Status Register (Index 43h), 5-26
  - disk drives, powered-down, 1-29–1-31
  - display configurations for LCD panels. See LCD panels.

- display interface for video controller
  - display-data bits, 3-2
  - signals (table), 3-2
- display memory for video controller
  - description, 3-3
  - SRAM address mapping during CPU access (table), 3-4
- DISREFOFF bit, 5-85
- DISW bit, 5-40
- divider-chain bits, time-base (table), 4-16
- Divisor Latch Lower Byte (Ports 2F8h & 3F8h), 4-10
- Divisor Latch Upper Byte (Ports 2F8h & 3F9h), 4-10
- DLAB bit, 4-11
- DM bit, 4-17
- DMA access, controlling (table), 5-5
- DMA clock
  - clock stop feature, 1-41
  - PMU clock speeds (table), 1-6
- DMA Controller Registers
  - DMA Controller 1 addresses (table), 4-4
  - DMA Controller 2 addresses (table), 4-5
- DMA Page Registers, 4-5–4-6
- DMAMMS bit, 5-34
- DMASTCLK bit, 5-77
- DMSTAT bit, 3-22–3-23
- DMWS bit, 5-47
- documentation
  - contents of manual, xviii
  - literature fulfillment service, iii
  - purpose of manual, xvii
  - related AMD publications, xviii
- DOS chip-select signal
  - $\overline{\text{DOSCS}}$  wait-state control-bit logic (table), 2-26
  - high-speed clock ROM cycles, 2-25
  - purpose and use, 2-25–2-26
- $\overline{\text{DOSCS}}$  signal
  - accessing ROM DOS memory, 2-9
  - address decode (table), 5-85
  - controlling with ROM Configuration 3 Register (Index B8h), 5-84
  - DOS chip-select signal, 2-9–2-10
  - high-speed clock ROM cycles, 2-25
  - number of wait states of  $\overline{\text{DOSCS}}$  cycle (table), 5-84
  - ROM chip-select command gating, 2-23
  - ROM chip-select signal, 2-25
  - wait-state control-bit logic (table), 2-26
- DOTD bit, 3-34
- Doze mode, 1-5–1-6. *See also* Low-Speed to Doze Timer Register (Index 84h).
- Doze to Sleep Timer Register (Index 85h)
  - description, 5-57
  - state-transition timing, 1-22
- DR bit, 4-12
- DRAM
  - 80-ns DRAM support, 2-26
  - 8254-based DRAM refresh, 2-4
  - choosing between DRAM and SRAM, 2-2
  - configurations, 2-3
  - DRAM bank miss wait state select logic (table), 5-37
  - DRAM first cycle wait state select logic (table), 5-37
  - Enhanced Page mode, 2-5
  - memory configuration (DRAM and SRAM) (table), 5-41
  - memory initialization example (table), 2-3
  - Page-mode DRAMs, 2-5
  - registers for setting up (table), 5-6
  - self-refresh DRAMs, 2-26
  - slow refresh, 2-4
- Drive Timer Register (Index 47h)
  - bit descriptions, 5-29
  - hard drive and floppy disk drive timer setting bit logic (table), 5-29
- drives. *See* disk drive management registers.
- DRLSD bit, 4-13
- DRQ pins as wake-up signals, 1-24
- DRQ0 bit, 5-67
- DRQ1 bit
  - Activity Mask 1 Register (Index 75h), 5-53
  - Activity Status 1 Register (Index A0h), 5-67
- DRQ1 signal (table), 1-23
- DRQ2 bit
  - Activity Mask 1 Register (Index 75h), 5-53
  - Activity Status 1 Register (Index A0h), 5-67
- DRQ2 signal (table), 1-23
- DRQ3 bit
  - Activity Mask 1 Register (Index 75h), 5-53
  - Activity Status 1 Register (Index A0h), 5-67
- DRQ3 signal (table), 1-23
- DRQ5 signal (table), 1-23
- DRQ6 signal (table), 1-23
- DRQ7-DRQ5 bits, 5-53
- DRQ7 signal (table), 1-23
- $\overline{\text{DSCE}}$  signal (table), 3-2

- DSE bit, 4-17
  - DSM14-DSMA0 signals (table), 3-2
  - DSMD7-DSMD0 signals (table), 3-2
  - $\overline{\text{DSOE}}$  signal (table), 3-2
  - DSPCON bit
    - description, 3-31
    - disabling internal video controller, 3-32
  - DSPMOD bit, 3-30
  - DSPTYPE bit, 3-30
  - DSR bit, 4-13
  - $\overline{\text{DSWE}}$  signal (table), 3-2
  - DTR bit, 4-12
  - DV2-DV0 bits, 4-15
  - DZ bit
    - NMI/SMI Control Register (Index A5h), 5-70
    - NMI/SMI Enable Register (Index 82h), 5-56
  - DZ0 bit, 5-75
  - DZ1 bit, 5-75
  - DZ2 bit, 5-55
  - DZ3 bit, 5-74
  - DZ4 bit, 5-74
- ## E
- e-mail support, iii
  - E0A21 bit, 5-46
  - E0A22 bit, 5-46
  - E0A23 bit, 5-44
  - E1A21 bit, 5-46
  - E1A22 bit, 5-46
  - E1A23 bit, 5-44
  - E2A21 bit, 5-46
  - E2A22 bit, 5-46
  - E2A23 bit, 5-46
  - E3A21 bit, 5-46
  - E3A22 bit, 5-46
  - E3A23 bit, 5-46
  - E4A21 bit, 5-42
  - E4A22 bit, 5-42
  - E4A23 bit, 5-44
  - E5A21 bit, 5-42
  - E5A22 bit, 5-42
  - E5A23 bit, 5-44
  - E6A21 bit, 5-42
  - E6A22 bit, 5-42
  - E6A23 bit, 5-44
  - E7A21 bit, 5-42
  - E7A22 bit, 5-42
  - E7A23 bit, 5-44
  - EB\_RMMD0 bit, 5-78
  - EB\_RMMD1 bit, 5-78
  - EIC bit, 4-18
  - ENSELFREF bit, 5-81
  - ÉlanSC300 microcontroller
    - architecture, xvii
    - configuring. *See* configuring ÉlanSC300 microcontroller.
    - overview, xvii
  - ELSI bit, 4-10
  - EMBA0 bit, 5-44
  - EMBA1 bit, 5-44
  - EMBA2 bit, 5-44
  - EMBA3 bit, 5-44
  - EMDP00 bit
    - MMSA Device 1 Register (Index 71h), 5-49
    - MMSB Device Register (Index 73h), 5-51
  - EMDP01 bit
    - MMSA Device 1 Register (Index 71h), 5-49
    - MMSB Device Register (Index 73h), 5-51
  - EMDP10 bit
    - MMSA Device 1 Register (Index 71h), 5-49
    - MMSB Device Register (Index 73h), 5-51
  - EMDP11 bit
    - MMSA Device 1 Register (Index 71h), 5-49
    - MMSB Device Register (Index 73h), 5-51
  - EMDP20 bit
    - MMSA Device 1 Register (Index 71h), 5-49
    - MMSB Device Register (Index 73h), 5-51
  - EMDP21 bit
    - MMSA Device 1 Register (Index 71h), 5-49
    - MMSB Device Register (Index 73h), 5-51
  - EMDP30 bit
    - MMSA Device 1 Register (Index 71h), 5-49
    - MMSB Device Register (Index 73h), 5-51

- EMDP31 bit
  - MMSA Device 1 Register (Index 71h), 5-49
  - MMSB Device Register (Index 73h), 5-51
- EMDP40 bit, 5-50
- EMDP41 bit, 5-50
- EMDP50 bit, 5-50
- EMDP51 bit, 5-50
- EMDP61 bit, 5-50
- EMDP70 bit, 5-50
- EMDP71 bit, 5-50
- EMIO0 bit, 5-44–5-45
- EMIO1 bit, 5-44–5-45
- EMIO2 bit, 5-44–5-45
- EMIO3 bit, 5-44–5-45
- EMSI bit, 4-10
- EN\_14M\_PP bit, 5-86
- Enable Software Switch Register (Video Index 12h)
  - description, 3-28
  - enabling extended video-controller registers (table), 3-28
- ENACDCHG bit, 5-19–5-20
- ENACIN bit, 5-76
- ENADIN1 bit, 5-71
- ENADIN2 bit, 5-71
- ENAICBL1L bit, 5-19–5-20
- ENAICBL2L bit, 5-19–5-20
- ENARDY bit, 5-19–5-20
- ENBCDCHG bit, 5-19–5-20
- ENBICBL1L bit, 5-19–5-20
- ENBICBL2L bit, 5-19–5-20
- ENBRDY bit, 5-19–5-20
- ENCLK bit, 5-63
- ENFD bit, 5-25
- ENFSTRDOS bit
  - function, 5-84
  - ROM DOS enable and wait-state select logic (table), 5-84
  - ROM DOS wait state select logic (table), 5-31
- ENFSTROMCS bit, 5-81
- Enhanced Page mode, 2-5
- ENHD bit, 5-25
- ENHIT bit, 5-64
- EN\_MCE\_PP bit, 5-86
- ENMI bit, 4-19
- ENMMSA bit, 5-40
- ENMMSB bit, 5-52
- EN\_MPOM bit, 5-86
- ENPIO bit, 5-25
- ENPMCIRQ0 bit, 5-52
- ENRAME2 bit, 5-82
- ENRDOSCS bit, 5-84
- ENROMA bit, 5-32
- ENROMC bit, 5-40
- ENROMD bit, 5-40
- ENROME bit, 5-40
- ENROMF bit, 5-40
- EPMODE bit, 5-39
- EPP-compliant registers. *See* Parallel Port Interface Registers.
- EPPMODE bit, 5-77
- EPPTO bit, 4-8
- EPS bit, 4-11
- ERDI bit, 4-10
- ERR bit
  - Parallel Status Port (Ports 279h & 379h), 4-8
  - Parallel Status Port (Ports 279h, 379h & 3BDh), 4-7
- ETDEI bit, 4-10
- extended video registers. *See* video index registers, extended.
- external-device control interface, 1-14–1-16
  - definition, 1-2
  - Latched Power pin, 1-16
  - LCD-Panel Voltage-Control pins, 1-15–1-16
  - Power-Management Control (PMC4-PMC0) pins, 1-14–1-15
  - Programmable General-Purpose Pins 2 and 3, 1-15
- EXTIR0ACT bit, 5-77
- EXTSMI pin
  - external SMI with multiple devices, 1-31
  - generating SMIs, 1-31
- EXTSMIEDG bit, 5-77
- EXTSMIEN bit, 5-77
- EXTSMISTE bit, 5-81

## F

- FCYCWAIT0 bit
  - DRAM first cycle wait state select logic (table), 5-37
  - function, 5-37
- FCYCWAIT1 bit
  - DRAM first cycle wait state select logic (table), 5-37
  - function, 5-37
- FD bit
  - Activity Status 2 Register (Index A1h), 5-67
  - bit descriptions, 5-54
- FDSMI bit, 5-26
- FDT0 bit, 5-29
- FDT1 bit, 5-29
- FDT2 bit, 5-29
- FDT3 bit, 5-29
- FDTOLTCH bit, 5-25
- FDWS0 bit
  - floppy drive wait states (table), 5-34
  - function, 5-34
- FDWS1 bit
  - floppy drive wait states (table), 5-34
  - function, 5-34
- FE bit, 4-12
- floppy disk drive interrupts. *See* disk drive management registers.
- FO0 bit, 5-75
- FO1 bit, 5-75
- FO2 bit, 5-55
- FO3 bit, 5-74
- FO4 bit, 5-74
- fonts, 3-13–3-19
  - font areas, 3-13–3-14
  - font example 1: the letter A, 3-16–3-17
    - address range determination, 3-16
    - bitmap, 3-16
    - selecting font area (table), 3-17
    - storing font (table), 3-17
  - font example 2: the letter M, 3-18–3-19
    - bitmap, 3-18
    - determining address ranges, 3-18
    - selecting font area (table), 3-19
    - storing the font (table), 3-19
  - storing fonts in video SRAM, 3-15–3-16
  - video controller font fetches to SRAM, 3-14–3-15
    - CA7-CA0, 3-15
    - DSPMOD, 3-15
    - font areas (table), 3-14
    - RA3-RA0, 3-15
    - SPCHRFMT, 3-15
    - SRAM address mapping during font fetches (table), 3-15
- FOR2-FOR0 bits
  - CGA attribute byte, 3-9
  - CGA foreground-color bit values (table), 3-10
  - HGA attribute byte, 3-11
  - bit values (table), 3-11
- FRDOSWS0 bit
  - function, 5-84
  - ROM DOS enable and wait-state select logic (table), 5-84
- FRDOSWS1 bit
  - function, 5-84
  - ROM DOS enable and wait-state select logic (table), 5-84
- FRM/VDRV signal (table), 3-2
- FSTROMWS0 bit, 5-81
- FSTROMWS1 bit, 5-81
- Function Enable 1 Register (Index B0h)
  - bit descriptions, 5-77–5-78
  - EXTSMI pin management, 1-31
  - latch and buffer logic (table), 5-78
  - powered-down device SMIs, 1-31
- Function Enable 2 Register (Index B1h)
  - bit descriptions, 5-78–5-79
  - DRAM memory configuration, 2-3
  - high-speed PLL frequency select (table), 5-79
  - RAM mode decode logic (table), 5-79

## G

- GATEA20 bit, 5-47
- General-Purpose I/O 1 Register (Index 9Ch), 5-65
- General-Purpose I/O 2 Register (Index 94h)
  - bit descriptions, 5-63
  - determining state of PGP pins in Off mode, 1-7
- General-Purpose I/O 3 Register (Index 95h)
  - bit descriptions, 5-63
  - determining state of PGP pins in Off mode, 1-7
- General-Purpose I/O Control Register (Index 91h)
  - bit descriptions, 5-61–5-62
  - determining state of PGP pins in Off mode, 1-7
- General-Purpose I/O Register (Index 89h), 5-59
- general-purpose pins. *See* PGP pins, controlling.
- GENSMI bit, 5-73

graphics mode, 3-4–3-7  
 640x200 and 320x200 video-memory data (table), 3-5  
 640x200 video-memory data for 480x320 LCD panel (table), 3-6  
 description, 3-4  
 memory requirements  
 480x320 LCD panel, 3-5  
 640x200 LCD panel, 3-5

Graphics Truncation Start Register (Video Index 23h), 3-36

Graphics Truncation Stop Register (Video Index 24h), 3-36–3-37

gray-scale mapping using color-mapping registers (table), 3-29

GRAYSC bit, 3-31

GRPCON bit, 3-24

## H

hard disk drive interrupts. *See* disk drive management registers.

HD bit

Activity Status 2 Register (Index A1h), 5-67  
 bit descriptions, 5-54

HDSMI bit, 5-26

HDT0 bit, 5-29

HDT1 bit, 5-29

HDT2 bit, 5-29

HDT3 bit, 5-29

HDTOLTCH bit, 5-25

HDWS0 bit

function, 5-34  
 hard drive wait states (table), 5-35

HDWS1 bit

function, 5-34  
 hard drive wait states (table), 5-35

HGA attribute byte

bit descriptions, 3-11  
 bit values (table), 3-11

HGA Configuration register (Port 3BFh), 3-23

HGA Index Address register (Port 3B4h), 3-21

HGA Index Data register (Port 3B5h), 3-22

HGA Mode Control register (Port 3B8h), 3-22

HGA Status register (Port 3BAh), 3-22–3-23

HGA text mode, 3-8

high-speed clock ROM cycles, 2-25

High-Speed PLL mode

clock-switching logic, 1-16–1-17  
 logic flowchart (figure), 1-17  
 high-speed PLL frequency select (table), 5-79  
 PMU operating-mode transitions, 1-8  
 purpose and use, 1-4–1-5  
 shutting off with software, 1-5

High-Speed to Low-Speed Timer Register (Index 83h)

description, 5-57  
 state-transition timing, 1-22

hit-count limit bit logic (table), 5-65

Horizontal Registers (Video Indexes 00-03h), 3-26–3-27

hotline support, iii

HOTRST bit, 4-19

HSA16 bit, 5-64–5-65

HSA17 bit, 5-64–5-65

HSA18 bit, 5-64–5-65

HSA19 bit, 5-64–5-65

HSPLLQ0 bit, 5-78

HSPLLQ1 bit, 5-78

## I

ICAINTIR0 bit, 5-14

ICAINTIR1 bit, 5-14

ICAINTIR2 bit, 5-14

ICAINTIR3 bit, 5-14

ICAIOEN bit, 5-14–5-15

ICAIOWIN1 bit, 5-14–5-15

ICAIOWIN2 bit, 5-14–5-15

ICA\_RI bit, 5-17

ICARIEN bit, 5-16

ICBINTIR0 bit, 5-23

ICBINTIR1 bit, 5-23

ICBINTIR2 bit, 5-23

ICBINTIR3 bit, 5-23

ICBIOEN bit, 5-23

ICBIOWIN1 bit, 5-23–5-24

ICBIOWIN2 bit, 5-23

- ICBL1AMSK bit, 5-16
- ICBL1BMSK bit, 5-16
- ICB\_RI bit, 5-17
- ICBRIEN bit, 5-16
- ICCDACHG bit, 5-70–5-71
- ICCDBCHG bit, 5-70–5-71
- ICCHGIR0 bit, 5-20–5-21
- ICCHGIR1 bit, 5-20–5-21
- ICCHGIR2 bit, 5-20–5-21
- ICCHGIR3 bit, 5-20–5-21
- ICIOA16S0 bit, 5-17–5-18
- ICIOA16S1 bit, 5-17–5-18
- ICIOB16S0 bit, 5-17–5-18
- ICIOB16S1 bit, 5-17–5-18
- ID0 bit, 4-10
- ID1 bit, 4-10
- inactivity states and transition intervals (table), 1-10
- $\overline{INIT}$  bit
  - Parallel Control Port (Ports 27Ah & 37Ah), 4-8
  - Parallel Control Port (Ports 27Ah, 37Ah & 37Eh), 4-7
- INT bit
  - Activity Mask 1 Register (Index 75h), 5-53
  - Activity Status 1 Register (Index A0h), 5-67
- INTEN bit
  - CGA attribute byte, 3-9
  - HGA attribute byte, 3-11
- Interlace Mode Register (Video Index 08h), 3-27
- internal video controller, disabling, 3-32
- interrupt registers
  - Interrupt Controller 1 I/O addresses (table), 4-2
  - Interrupt Controller 2 I/O addresses (table), 4-3
  - Interrupt Enable Register (Ports 2F9h & 3F9h), 4-10
  - Interrupt Identification Register (Ports 2FAh & 3FAh)
    - bit descriptions, 4-10
    - interrupt ID bit logic (table), 4-11
  - interrupt redirect logic (table), 5-80
  - PCMCIA I/O Card IRQ Redirection Control A Register (Index 06h), 5-14–5-15
  - PCMCIA I/O Card IRQ Redirection Control B Register (Index 16h), 5-23–5-24
  - PCMCIA Status Change IRQ Redirection Register (Index 0Eh), 5-20–5-21
  - PIRQ Configuration Register (Index B2h), 5-79–5-80
- interrupts
  - enabling and specifying mapping (table), 5-5
  - PCMCIA interrupt handler, 2-22
- INTIOWAIT bit, 5-37
- INVICAIRQ bit, 5-14
- I/O Activity Address 0 Register (Index 8Ch)
  - description, 5-59
  - PMU operating-mode transitions, 1-8
- I/O Activity Address 1 Register (Index 8Dh)
  - description, 5-60
  - PMU operating-mode transitions, 1-8
- I/O drive type description (table), 5-86
- I/O Timeout Register (Index 40h)
  - bit descriptions, 5-25
  - powered-down device SMIs, 1-31
- I/O Wait State Register (Index 61h)
  - bit descriptions, 5-34
  - floppy disk drive wait states (table), 5-34
  - general bus I/O wait states (table), 5-35
  - hard disk drive wait states (table), 5-35
- I/O window map for PCMCIA cards, 2-20
- IOCD0 bit, 5-32–5-33
- IOCD1 bit, 5-32–5-33
- IOCHCK bit, 4-18
- $\overline{IOCS16}/LCDDL0$  signal (table), 3-2
- IOR bit, 5-26
- $\overline{IOR}$  signal, 2-20
- $\overline{TORESET}$  pin, 1-40
- IOW bit, 5-26
- $\overline{IOW}$  signal, 2-20
- IOWS0 bit
  - function, 5-34
  - general bus I/O wait states (table), 5-35
- IOWS1 bit
  - function, 5-34
  - general bus I/O wait states (table), 5-35
- IP bit, 4-10
- IRQ pins as wake-up signals, 1-24
- IRQ0SMI bit, 5-26
- IRQ0SMIEN bit, 5-76
- IRQ1 signal (table), 1-23
- IRQ3 bit, 5-17
- IRQ3 signal (table), 1-23
- IRQ3MSK bit, 5-16



- IRQ4 bit, 5-17
- IRQ4 signal (table), 1-23
- IRQ4MSK bit, 5-16
- IRQ6 bit, 1-29
- IRQ8 bit, 5-17
- IRQ8 signal (table), 1-23
- IRQ8MSK bit, 5-16
- IRQ14 bit, 1-29
- IRQ14/LCDDL2 signal (table), 3-2
- IRQEN bit
- enabling hard-drive interrupt, 1-29
  - Miscellaneous 4 Register (Index 44h), 5-27
  - Parallel Control Port (Ports 27Ah & 37Ah), 4-8
  - Parallel Control Port (Ports 27Ah, 37Ah & 37Eh), 4-7
- IRQF bit, 4-17
- ISA bus
- configurations, 2-6–2-7
    - bus configuration (table), 2-6
    - ÉlanSC300 microcontroller bus configurations (table), 2-7
  - I/O access command delay, 8-bit (table), 5-33
  - memory access command delay, 8-bit (table), 5-33
  - memory-cycle wait states (table)
    - 8-bit, 5-36
    - 16-bit, 5-36
- ## K
- KB bit
- Activity Mask 1 Register (Index 75h), 5-53–5-54
  - Activity Status 1 Register (Index A0h), 5-67
- KBCLKEN bit, 4-18
- KBSMIEN bit
- SMI Enable Register (Index 41h), 5-25
  - SMI I/O Status Register (Index 42h), 5-26
- keyboard. *See* XT-keyboard interface.
- keyboard clock, disabled in Sleep mode, 1-7
- ## L
- latch and buffer logic (table), 5-78
- Latched Power pin, 1-16
- LCD controller. *See also* video controller.
- bus and internal LCD configurations, 2-6–2-7
    - ÉlanSC300 microcontroller bus configurations (figure), 2-7
  - LCD-Panel Voltage-Control pins, 1-15–1-16
  - LCD panels
    - 640x200 and 320x200 video-memory data (table), 3-5
    - 640x200 video-memory data for 480x320 LCD panel (table), 3-6
    - display configuration programming examples, 3-37–3-42
      - 320x240 LCD panel configuration (table)
        - CGA (Video Index 18h, Bit 0 = 0), 3-40
        - HGA (Video Index 18h, Bit 0 = 1), 3-41
      - 480x320 LCD panel configuration (table), 3-39
      - 640x200 LCD panel configuration (table), 3-38
      - 720x348 LCD panel configuration (table), 3-42
    - memory requirements
      - 480x320 LCD panel, 3-5
      - 640x200 LCD panel, 3-5
  - LCD Special Register (Video Index 25h), 3-37
  - LCDD0/R signal (table), 3-2
  - LCDD1/G signal (table), 3-2
  - LCDD2/B signal (table), 3-2
  - LCDD3/I signal (table), 3-2
  - LCDDUEN bit, 5-78–5-79
  - LCDMOD bit, 3-37
  - LCDPAN1-LCDPAN0 bits, 3-30
  - LIND0 bit, 5-68–5-69
  - LIND1 bit, 5-68–5-69
  - LIND2 bit, 5-68–5-69
  - Line Control Register (Ports 2FBh & 3FBh)
    - bit descriptions, 4-11
    - word length bit logic (table), 4-11
  - Line Status Register (Ports 2FDh & 3FDh), 4-12
  - LOOP bit, 4-12
  - low-speed CPU clock, controlling, 1-6
  - low-speed duration period select logic (table), 5-67
  - Low-Speed PLL mode
    - clock-switching logic, 1-17
    - logic flowchart (figure), 1-17
    - description, 1-5
    - low-speed PLL mode CPU clock speed select (table), 5-76
    - PMU operating-mode transitions, 1-8
  - Low-Speed to Doze Timer Register (Index 84h)
    - description, 5-57
    - state-transition timing, 1-22
  - LOW0 bit, 5-66–5-67

- LOW1 bit, 5-66–5-67
- $\overline{\text{LPH}}$  pin, 1-16
- LPH1 bit, 5-68
- LPT bit
- Activity Mask 2 Register (Index 76h), 5-54
  - Activity Status 2 Register (Index A1h), 5-67
- LSA14 bit, 5-64
- LSA15 bit, 5-64
- LSA16 bit, 5-64
- LSA17 bit, 5-64
- LSA18 bit, 5-64
- LSA19 bit, 5-64
- LSRWCNTL bit, 5-30
- $\overline{\text{LVDD}}$  pin
- controlling voltage on LCD panel (note), 1-16
  - description (table), 3-3
  - miscellaneous signals interface, 3-3
  - purpose and use, 1-15–1-16
- $\overline{\text{LVEE}}$  pin
- controlling voltage on LCD panel (note), 1-16
  - description (table), 3-3
  - miscellaneous signals interface, 3-3
  - purpose and use, 1-15–1-16
- ## M
- M signal (table), 3-2
- MAINOFF bit, 5-76
- manual. *See* documentation.
- MASTG bit, 3-23
- Max Scan Line Register (Video Index 09h)
- description, 3-27
  - determining size of stored font, 3-14
- $\overline{\text{MCEH}}$  signal, 2-20
- $\overline{\text{MCEH}}_A$  signal, 5-14–5-15
- $\overline{\text{MCEL}}$  signal, 2-20
- $\overline{\text{MCEL}}_A$  signal, 5-14–5-15
- $\overline{\text{MCS16}}$ /LCDDL1 signal (table), 3-2
- $\overline{\text{MCS16}}$  signal, 5-81, 5-84
- MEMADRS0 bit, 5-85
- MEMADRS1 bit, 5-85
- MEMCTLS0 bit, 5-85
- MEMCTLS1 bit, 5-85
- MEMDATS0 bit, 5-85
- MEMDATS1 bit, 5-85
- Memory Configuration 1 Register (Index 66h)
- bit descriptions, 5-41
  - bus option status table, 5-41
  - DRAM memory configuration, 2-3
  - memory configuration (DRAM and SRAM) (table), 5-41
  - setting up Page-mode DRAM accesses, 2-5
  - SRAM wait states, 2-5
- Memory Configuration 2 Register (Index B9h)
- bit descriptions, 5-85
  - I/O drive type description (table), 5-85
  - Off mode function, 1-7
  - output drive strength select logic (table), 5-85
- memory interface for video controller
- display memory, 3-3–3-4
  - SRAM address mapping during CPU access (table), 3-4
  - memory interface signals (table), 3-2
- memory management. *See also* Memory Mapping System (MMS).
- 16-Mbyte address spaces, 2-1
  - 64-Mbyte address spaces, 2-1
  - 80-ns DRAM support, 2-26
  - DOS chip-select signal, 2-25–2-26
  - $\overline{\text{DOSCS}}$  wait-state control-bit logic (table), 2-26
  - high-speed clock ROM cycles, 2-25
  - overview, 2-1
  - PCMCIA controller, 2-16–2-22
    - block diagram (figure), 2-18
    - control logic, 2-20
    - ÉlanSC300 microcontroller PCMCIA signal compatibility (table), 2-16–2-17
    - functional blocks, 2-17
    - interrupt handler, 2-22
    - I/O window map, 2-20
    - memory window map, 2-19
    - programming logic, 2-21
    - status logic, 2-21
- ROM bios memory, 2-7–2-9
- address initialization (table), 2-8
  - copying ROM contents (figure), 2-9
  - high memory (figure), 2-8
- ROM chip-select command gating, 2-23
- ROM chip-select signal, 2-25
- $\overline{\text{ROMCS}}$  wait-state control-bit logic (table), 2-25
- ROM DOS memory, 2-9–2-10
- self-refresh DRAMs, 2-26
- system memory, 2-2–2-7. *See also* DRAM; SRAM.
- DRAM configurations, 2-3
  - ISA, local bus, and internal LCD configurations, 2-6–2-7

- overview, 2-2–2-3
- refresh and wait states, 2-4–2-6
- SRAM configurations, 2-3–2-4
- typical AT address space (figure), 2-2
- wait states and command delays, 2-23
  - command delay duration for various cycles (table), 2-24
  - wait states for various cycles (table), 2-24
- Memory Mapping System (MMS), 2-10–2-15. *See also*
  - specific MMS bits and registers.
  - memory mapping system (figure), 2-11
  - memory-window map for PCMCIA cards, 2-19
  - registers (table), 2-19
  - MMS mapping example (figure), 2-14
  - MMS mapping example settings (table), 2-14–2-15
  - MMSA and MMSB (figure), 2-11
  - registers for setting up, 2-12, 5-6
- Memory Write Activity Lower Boundary Register (Index 9Ah)
  - bit descriptions, 5-64
  - hit-count limit bit logic (table), 5-65
  - PMU operating-mode transitions, 1-8
- Memory Write Activity Upper Boundary Register (Index 9Bh)
  - bit descriptions, 5-64–5-65
  - PMU operating-mode transitions, 1-8
- MEMR signal
  - disabling, 5-30
  - generated by control logic, 2-20
  - quiet bus feature, 1-41
- MEMW signal
  - disabling, 5-30
  - generated by control logic, 2-20
  - quiet bus feature, 1-41
- merging PMU modes, 1-9
- Micro Power Off mode, 1-40
- mirrored I/O register conflicts (note), 4-1
- Miscellaneous 1 Register (Index 6Fh)
  - bit descriptions, 5-47
  - MMS memory range select logic (table), 5-47–5-48
  - preventing overlap of linear-decode address range, 2-10
- Miscellaneous 2 Register (Index 6Bh)
  - bit descriptions, 5-43
  - enabling SMIs, 1-27
- Miscellaneous 3 Register (Index BAh)
  - bit descriptions, 5-86
  - parallel port pin redefinition (table), 5-87
- Miscellaneous 4 Register (Index 44h)
  - bit descriptions, 5-27
  - data-path disabling logic, 1-41
  - powered-down device SMIs, 1-29
- Miscellaneous 5 Register (Index B3h)
  - bit descriptions, 5-81
  - controlling number of wait states, 2-25
  - determining source of SMIs, 1-28
  - enabling high-speed clock rate, 2-25
  - ROM BIOS enable and wait-state select logic (table), 5-81
- Miscellaneous 6 Register (Index 70h)
  - bit descriptions, 5-48
  - generating PMU-activity event with ACIN pin, 1-34
- miscellaneous signals interface, video controller
  - description, 3-3
  - signals (table), 3-3
- MISOUT bit
  - DRAM bank miss wait state select logic (table), 5-37
  - MMS Memory Wait State 1 Register (Index 62h), 5-35
- MMS. *See* Memory Mapping System (MMS).
- MMS Address Extension 1 Register (Index 6Ch), 5-44
- MMS Address Extension 2 Register (Index 6Eh), 5-46
- MMS Address Register (Index 6Dh)
  - bit descriptions, 5-44–5-45
  - MMSA/B Page Register I/O addresses (table), 5-45
  - MMSA base addresses (table), 5-46
  - page register contents description (table), 5-45
- MMS bit, 5-67
- MMS memory range select logic (table), 5-47–5-48
- MMS Memory Wait State 1 Register (Index 62h)
  - 8-bit ISA memory-cycle wait states (table), 5-36
  - 16-bit ISA memory-cycle wait states (table), 5-36
  - bit descriptions, 5-35
  - setting up Page-mode DRAM accesses, 2-5
- MMS Memory Wait State 2 Register (Index 50h)
  - bit descriptions, 5-30–5-31
  - PCMCIA command state select logic (table), 5-32
  - PCMCIA wait state select logic (table), 5-31
  - ROM DOS command delay select logic (table), 5-31
  - ROM DOS wait state select logic (table), 5-31
- MMS1P0 bit, 5-73
- MMS1P1 bit, 5-73
- MMS1P2 bit, 5-73
- MMS1P3 bit, 5-73
- MMSA Address Extension 1 Register (Index 67h), 5-42
- MMSA Device 1 Register (Index 71h)
  - bit descriptions, 5-49
  - page 0-3 device select (table), 5-49

- MMSA Device 2 Register (Index 72h)  
 bit descriptions, 5-50  
 page 4-7 device select (table), 5-50
- MMSA Socket Register (Index A8h), 5-72
- MMSABSEL bit, 5-52
- MMSAPG00 bit, 5-82
- MMSAPG01 bit, 5-82
- MMSAPG10 bit, 5-82
- MMSAPG11 bit, 5-82
- MMSAPG20 bit, 5-82
- MMSAPG21 bit, 5-82
- MMSAPG30 bit, 5-82
- MMSAPG31 bit, 5-82
- MMSAPG40 bit, 5-83
- MMSAPG41 bit, 5-83
- MMSAPG50 bit, 5-83
- MMSAPG51 bit, 5-83
- MMSAPG60 bit, 5-83
- MMSAPG61 bit, 5-83
- MMSAPG70 bit, 5-83
- MMSAPG71 bit, 5-83
- MMSB Control Register (Index 74h)  
 bit descriptions, 5-52  
 enabling Sleep mode on Battery Level 2, 1-33  
 starting low-speed CPU clock, 1-6
- MMSB Device Register (Index 73h)  
 bit descriptions, 5-51  
 page 0-3 device select (table), 5-51
- MMSB Socket Register (Index A9h)  
 bit descriptions, 5-73  
 enabling SMIs, 1-27  
 selecting SMI or NMI, 5-56
- MMSBPG00 bit, 5-83
- MMSBPG01 bit, 5-83
- MMSBPG10 bit, 5-83
- MMSBPG11 bit, 5-83
- MMSBPG20 bit, 5-83
- MMSBPG21 bit, 5-83
- MMSBPG30 bit, 5-83
- MMSBPG31 bit, 5-83
- MMSP0 bit, 5-72
- MMSP1 bit, 5-72
- MMSP2 bit, 5-72
- MMSP3 bit, 5-72
- MMSP4 bit, 5-72
- MMSP5 bit, 5-72
- MMSP6 bit, 5-72
- MMSP7 bit, 5-72
- MMSZ0 bit, 5-47
- MMSZ1 bit, 5-47
- MMSZ2 bit, 5-47
- MMSZ3 bit, 5-47
- MOD0 bit, 5-41
- MOD1 bit, 5-41
- Modem Control Register (Ports 2FCh & 3FCh), 4-12
- Modem Status Register (Ports 2FEh & 3FEh), 4-13
- MODFRQ bit, 3-37
- MRDLY bit, 5-47
- MS0 bit  
 function, 5-41  
 memory configuration (DRAM and SRAM) (table), 5-41
- MS1 bit  
 function, 5-41  
 memory configuration (DRAM and SRAM) (table), 5-41
- MS2 bit  
 function, 5-41  
 memory configuration (DRAM and SRAM) (table), 5-41
- MTS bit, 5-48

## N

- NENLB2 bit, 5-52
- NENLB4 bit, 5-52
- NFRDOSEN bit, 5-35
- NFROMEN bit, 5-35
- NMI/RTC Index Address Register (Port 070h), 4-19
- NMI/SMI Control Register (Index A5h)  
 description, 5-70  
 enabling  $\overline{BL1}$  pin, 1-32  
 processing NMI or SMI sources, 1-28  
 resume pseudocode, 1-39  
 SMI generation by  $\overline{BL2}$  pin, 1-33  
 suspend pseudocode, 1-38

suspend/resume operation, 1-36  
 Temporary-On mode, 1-26–1-27

NMI/SMI Enable Register (Index 82h)  
 bit descriptions, 5-56  
 enabling  $\overline{BL1}$  pin, 1-32  
 selecting NMI or SMI, 5-56  
 SMI generation by  $\overline{BL2}$  pin, 1-33  
 start of SMI handler, 1-37

NMIs. *See also* SMI and NMI control.  
 overview, 1-2  
 suspend/resume operation, 1-35

Nonmaskable Interrupt (NMI). *See* NMIs.

## O

OE bit, 4-12

Off mode, 1-7–1-8

ON bit

NMI/SMI Enable Register (Index 82h), 5-56  
 NMI/SMI Control Register (Index A5h), 5-70

ONCLK0 bit, 5-76

ONCLK1 bit, 5-76

operating-mode transitions, PMU

High-Speed PLL, 1-8  
 illustration, 1-4  
 Low-Speed PLL, 1-8  
 registers for setting activities, 1-8  
 Video PLL, 1-8

$\overline{OUT1}$  bit, 4-12

$\overline{OUT2}$  bit, 4-12

output drive strength select logic (table), 5-86

## P

page 0-3 device select (table)

MMSA Device 1 Register (Index 71h), 5-49  
 MMSB Device Register (Index 73h), 5-51

page 4-7 device select (table), 5-50

page boundaries in 80x25 text mode (table), 3-33

page mode

Enhanced Page mode, 2-5  
 Page-mode DRAMs, 2-5

page register

contents description (table), 5-45  
 MMSA/B page register I/O addresses (table), 5-45

pages

definition, 2-5  
 page hits, 2-5

parallel port

parallel port pin redefinition (table), 5-87  
 registers for setting up (table), 5-7

Parallel Port Interface Registers, 4-6–4-9

AT-compatible mode

Parallel Control Port (Ports 27Ah, 37Ah & 37Eh), 4-7

Parallel Data Port (Ports 278h, 278h, & 3BCh), 4-7

Parallel Status Port (Ports 279h, 379h & 3BDh), 4-7

EPP-compliant mode

Parallel Control Port (Ports 27Ah & 37Ah), 4-8

Parallel Data Port (Ports 278h & 378h), 4-8

Parallel EPP 32-Bit Data Register (Ports 27C-27Fh & 37C-37Fh), 4-9

Parallel EPP Address Port (Ports 27Bh & 37Bh), 4-9

Parallel Status Port (Ports 279h & 379h), 4-8

overview, 4-6–4-9

PC/AT bus. *See also* bus.

controlling PC/AT bus and its timing (table), 5-4  
 determining bus configuration (table), 5-4

PC/AT compatibility with CPU, controlling (table), 5-5

PC/AT peripheral registers

conflicts with mirrored I/O registers (note), 4-1

DMA Controller Registers

DMA Controller 1 addresses (table), 4-4

DMA Controller 2 addresses (table), 4-5

DMA Page Registers, 4-5–4-6

Interrupt Controller Registers

Interrupt Controller 1 I/O addresses (table), 4-2

Interrupt Controller 2 I/O addresses (table), 4-3

NMI/RTC Index Address Register (Port 070h), 4-19

overview, 4-1

Parallel Port Interface Registers, 4-6–4-9

AT-compatible mode

Parallel Control Port (Ports 27Ah, 37Ah & 37Eh), 4-7

Parallel Data Port (Ports 278h, 278h, & 3BCh), 4-7

Parallel Status Port (Ports 279h, 379h & 3BDh), 4-7

EPP-compliant mode

Parallel Control Port (Ports 27Ah & 37Ah), 4-8

Parallel Data Port (Ports 278h & 378h), 4-8

Parallel EPP 32-Bit Data Register (Ports 27C-27Fh & 37C-37Fh), 4-9

Parallel EPP Address Port (Ports 27Bh & 37Bh), 4-9

Parallel Status Port (Ports 279h & 379h), 4-8

overview, 4-6–4-9

- Port 92 (Port 092h), 4-19
- Port B Register (Port 061h), 4-18
- probing address data with logic analyzer (note), 4-1
- Programmable Interval Timer Registers, 4-3
- real-time clock
  - addressing, 4-14
  - overview, 4-14
  - Register A (RTC Index 0Ah), 4-15–4-16
  - Register B (RTC Index 0Bh), 4-17
  - Register C (RTC Index 0Ch), 4-17
  - Register D (RTC Index 0Dh), 4-17
  - RTC register summary (table), 4-15
- RTC Index Data Register (Port 071h), 4-19
- System Timer Registers (table), 4-3
- UART registers
  - Divisor Latch Lower Byte (Ports 2F8h & 3F8h), 4-10
  - Divisor Latch Upper Byte (Ports 2F8h & 3F9h), 4-10
  - Interrupt Enable Register (Ports 2F9h & 3F9h), 4-10
  - Interrupt Identification Register (Ports 2FAh & 3FAh), 4-10–4-11
  - Line Control Register (Ports 2FBh & 3FBh), 4-11
  - Line Status Register (Ports 2FDh & 3FDh), 4-12
  - Modem Control Register (Ports 2FCh & 3FCh), 4-12
  - Modem Status Register (Ports 2FEh & 3FEh), 4-13
  - Receiver Buffer Register (Ports 2F8h & 3F8h), 4-9
  - Scratch Pad Register (Ports 2FFh & 3FFh), 4-13
  - Transmitter Holding Register (Ports 2F8h & 3F8h), 4-9
- XT Keyboard Data Register (Port 060h), 4-18
- PCM pins
  - PIO timeout settings example (table), 1-12
  - power-management control pin settings (table), 1-12
- PCMCIA Card Reset Register (Index B4h)
  - bit descriptions, 5-82
  - DRAM memory configuration, 2-3
- PCMCIA controller, 2-16–2-22
  - block diagram (figure), 2-18
  - control logic, 2-20
  - ÉlanSC300 microcontroller PCMCIA signal compatibility (table), 2-16–2-17
  - functional blocks, 2-17
  - interrupt handler, 2-22
  - I/O window map, 2-20
  - memory window map
    - description, 2-19
    - registers (table), 2-19
  - programming logic, 2-21
  - registers for setting up PCMCIA card interface (table), 5-7–5-8
  - status logic, 2-21
- PCMCIA Data Width Register (Index 0Ah)
  - bit descriptions, 5-17
  - ICIOA16SO-ICIOA16S1 bit logic (table), 5-18
  - ICIOB16SO-ICIOB16S1 bit logic (table), 5-18
- PCMCIA I/O Card IRQ Redirection Control A Register (Index 06h)
  - bit descriptions, 5-14–5-15
  - interrupt redirect bit logic (table), 5-15
- PCMCIA I/O Card IRQ Redirection Control B Register (Index 16h)
  - bit descriptions, 5-23–5-24
  - interrupt redirection bit logic (table), 5-24
- PCMCIA I/O Window A1 Lower Byte End Register (Index 01h), 5-13
- PCMCIA I/O Window A1 Lower Byte Start Register (Index 00h), 5-13
- PCMCIA I/O Window A1 Upper Byte End Register (Index 02h), 5-13
- PCMCIA I/O Window A2 Lower Byte End Register (Index 04h), 5-14
- PCMCIA I/O Window A2 Lower Byte Start Register (Index 03h), 5-13
- PCMCIA I/O Window A2 Upper Byte Register (Index 05h), 5-14
- PCMCIA I/O Window B1 Lower Byte End Register (Index 11h), 5-22
- PCMCIA I/O Window B1 Lower Byte Start Register (Index 10h), 5-21
- PCMCIA I/O Window B1 Upper Byte End Register (Index 12h), 5-22
- PCMCIA I/O Window B2 Lower Byte End Register (Index 14h), 5-22
- PCMCIA I/O Window B2 Lower Byte Start Register (Index 13h), 5-22
- PCMCIA I/O Window B2 Upper Byte Register (Index 15h), 5-23
- PCMCIA REGA Address Register (Index 8Ah), 5-59
- PCMCIA REGB Address Register (Index 9Eh), 5-66
- PCMCIA Socket A Status Register (Index A2h)
  - bit descriptions, 5-68
  - enabling  $\overline{BLT}$  pin, 1-32
  - processing NMI or SMI sources, 1-28
  - resume pseudocode, 1-39
  - SMI generation by  $\overline{BLT}$  pin, 1-33
  - status logic information, 2-21
  - suspend pseudocode, 1-38
  - suspend/resume operation, 1-36

- PCMCIA Socket B Status Register (Index 0Ch)  
 bit descriptions, 5-19  
 status logic information, 2-21
- PCMCIA Status Change IRQ Enable Register (Index 0Dh), 5-19–5-20
- PCMCIA Status Change IRQ Redirection Register (Index 0Eh)  
 bit descriptions, 5-20–5-21  
 IRQ select logic, 5-21
- PCMCIA Status Change Register (Index A6h)  
 bit descriptions, 5-70–5-71  
 processing NMI or SMI sources, 1-28  
 status logic information, 2-21
- PCMCIA VPPA Address Register (Index 07h), 5-15
- PCMCIA VPPB Address Register (Index 17h), 5-24
- PCMCIA wait state select logic (table), 5-31
- PCMCIASMI bit, 5-26
- PE bit  
 Line Control Register (Ports 2FBh & 3FBh), 4-11  
 Line Status Register (Ports 2FDh & 3FDh), 4-12  
 Parallel Status Port (Ports 279h & 379h), 4-8  
 Parallel Status Port (Ports 279h, 379h & 3BDh), 4-7
- periodic-interrupt rate-selection bits (table), 4-16
- peripheral-device power programming example, 1-11–1-13  
 PIO timeout settings (table), 1-12  
 power-management control pin settings (table), 1-12  
 SMI-generation settings for PIO access (table), 1-13
- PF bit, 4-17
- PFWS bit  
 DRAM first cycle wait state select logic (table), 5-37  
 ROM Configuration 1 Register (Index 65h), 5-40
- PG0IN bit, 5-68
- PG0IO0 bit, 5-61–5-62
- PG0IO1 bit, 5-61–5-62
- PG1IN bit, 5-69
- PG1IO0 bit, 5-61–5-62
- PG1IO1 bit, 5-61–5-62
- PG2IO0 bit, 5-61–5-62
- PG2IO1 bit, 5-61–5-62
- PG3IO0 bit, 5-61–5-62
- PG3IO1 bit, 5-61–5-62
- PGP pins, controlling  
 General-Purpose I/O 1 Register (Index 9Ch), 5-65
- General-Purpose I/O 2 Register (Index 94h), 1-7, 5-63
- General-Purpose I/O 3 Register (Index 95h), 1-7, 5-63
- General-Purpose I/O Control Register (Index 91h), 1-7, 5-61–5-62
- General-Purpose I/O Register (Index 89h), 5-59  
 registers for setting up (table), 5-9
- PGP0DIR bit, 5-48
- PGP1DIR bit, 5-52
- PGP2 pin, 1-15
- PGP3 pin, 1-15
- PIE bit, 4-17
- PIND0 bit, 5-69
- PIND1 bit, 5-69
- PIND2 bit, 5-69
- PIO Address Register (Index 45h)  
 bit descriptions, 5-27  
 powered-down device SMIs, 1-29
- PIO timeout settings example (table), 1-12
- PIO Timer Register (Index 46h)  
 address range decode logic (table), 5-28  
 bit descriptions, 5-28  
 powered-down device SMIs, 1-29  
 setting logic (table), 5-28
- PIO0 bit  
 Activity Mask 2 Register (Index 76h), 5-54  
 Activity Status 2 Register (Index A1h), 5-67
- PIO1 bit  
 Activity Mask 2 Register (Index 76h), 5-54  
 Activity Status 2 Register (Index A1h), 5-67
- PIOSMI bit, 5-26
- PIOT0 bit, 5-28
- PIOT1 bit, 5-28
- PIOT2 bit, 5-28
- PIOT3 bit, 5-28
- PIOTOLTC bit, 5-25
- PIRQ Configuration Register (Index B2h)  
 bit descriptions, 5-79–5-80  
 interrupt redirect logic (table), 5-80
- PMC pins. *See* Power-Management Control (PMC4-PMC0) pins.
- PMCSMI bit, 5-26
- PMU. *See* Power Management Unit (PMU).
- PMU clock speeds (table), 1-6

- PMU Control 1 Register (Index A7h)
  - bit descriptions, 5-71
  - data-path disabling logic, 1-41
  - PMU operating-mode transitions, 1-8
- PMU Control 2 Register (Index AFh)
  - bit descriptions, 5-77
  - enabling  $\overline{BL1}$  pin, 1-32
  - state-transition timing, 1-22
- PMU Control 3 Register (Index ADh)
  - bit descriptions, 5-76
  - generating PMU-activity event with ACIN pin, 1-34
  - low-speed PLL mode CPU clock speed select (table), 5-76
  - PMU operating-mode transitions, 1-8
- PMU mode
  - last PMU mode indicator bits (table), 5-69
  - PMU mode select logic (table), 5-58
  - present PMU mode indicator bits (table), 5-69
- PMU operating-mode transitions, 1-8
- PMU state machine
  - general-purpose control using PMC pins, 1-14
  - purpose and use, 1-2
- PMU Timer. *See* State-Transition Timer.
- PMW bit
  - Activity Mask 2 Register (Index 76h), 5-54
  - Activity Status 2 Register (Index A1h), 5-67
- Port 92 (Port 092h), 4-19
- Port B Register (Port 061h), 4-18
- port registers. *See* video port registers.
- POWDWN bit, 3-31
- Power Control 1 Register (Index 80h)
  - bit descriptions, 5-55
  - merging PMU modes, 1-9
  - shutting down low-speed and video PLLs in Doze mode, 1-6
- Power Control 2 Register (Index 81h)
  - bit descriptions, 5-56
  - merging PMU modes, 1-9
  - shutting down low-speed and video PLLs
    - Sleep mode, 1-7
    - Suspend mode, 1-7
- Power Control 3 Register (Index ABh), 5-74
- Power Control 4 Register (Index ACh), 5-75
- power management. *See also* Power Management Unit (PMU).
  - activity monitors
    - description, 1-19
    - state transition flowchart (figure), 1-20–1-21
  - auto low-speed logic, 1-40
  - battery-management logic, 1-32–1-34
    - AC Input Status (ACIN) pin, 1-34
    - Battery Level 1 ( $\overline{BL1}$ ) pin, 1-32
    - Battery Level 2 ( $\overline{BL2}$ ) pin, 1-32–1-33
    - Battery Level 3 ( $\overline{BL3}$ ) pin, 1-33
    - Battery Level 4 ( $\overline{BL4}$ ) pin, 1-33–1-34
    - functionality of BL4-BI1 (table), 1-32
  - clock-switching logic, 1-16–1-18
    - clock startup and shutdown logic, 1-17
    - logic flowchart (figure), 1-18
    - CPU/memory clock switching, 1-16–1-17
  - data-path disabling logic, 1-41
  - DMA clock stop, 1-41
  - external-device control interface, 1-14–1-16
    - Latched Power pin, 1-16
    - LCD-Panel Voltage-Control pins, 1-15–1-16
    - Power-Management Control (PMC4-PMC0) pins, 1-14–1-15
    - Programmable General-Purpose Pins 2 and 3, 1-15
  - Micro Power-Off mode, 1-40
  - overview, 1-1
  - quiet bus, 1-41
  - registers for managing (table)
    - controlling activities and events, 5-10
    - controlling power management state timers, 5-11
    - determining power management status, 5-10
  - slow refresh, 1-41
  - SMI and NMI control, 1-25–1-31
    - accesses to powered-down device SMI, 1-29–1-31
    - device-powerdown flowchart (figure), 1-30
    - enabling SMIs, 1-27
    - external SMI
      - with multiple devices, 1-31
      - with single device, 1-31
    - external SMI pin, 1-31
    - processing flowchart (figure), 1-26
    - processing NMI or SMI source, 1-28–1-29
    - sources for generating SMIs, 1-25
    - temporary-on mode, 1-26–1-27
    - treatment of pending SMIs, 1-31
  - state-transition timer, 1-22–1-23
  - suspend/resume pin logic, 1-34–1-39
    - avoiding problems, 1-39
    - capabilities required, 1-35
    - overview, 1-34–1-35
    - programming considerations, 1-36
    - required initialization, 1-37
    - resume input causing SMI, 1-39
    - resume inputs, 1-35
    - resume pseudocode, 1-39
    - start of SMI handler, 1-37
    - suspend input causing SMI, 1-37



- suspend inputs, 1-35
- suspend pseudocode, 1-38
- techniques, 1-1
- wake-up logic, 1-23–1-25

Power-Management Control (PMC4-PMC0) pins, 1-14–1-15

- general-purpose control using PMU state machine, 1-14
- peripheral-device power settings (table), 1-12
- PMC pin functionality (table), 1-14
- purpose and use, 1-14
- registers for setting up (table), 5-9
- timer-controlled shutdown using SMI interface, 1-14–1-15

Power Management Unit (PMU), 1-2–1-13

- clock sources, 1-9
- components, 1-2–1-3
- merging PMU modes, 1-9
- operating-mode transitions
  - description, 1-8
  - illustration, 1-4
- power conservation techniques, 1-2
- power management modes, 1-3–1-8
  - Doze mode, 1-5–1-6
  - High-Speed PLL mode, 1-4–1-5
  - Low-Speed PLL mode, 1-5
  - Off mode, 1-7–1-8
- overview, 1-3
- PMU clock speeds (table), 1-6
- Sleep mode, 1-7
- Suspend mode, 1-7
- programming examples
  - inactivity states and transition intervals (table), 1-10
  - peripheral-device power, 1-11–1-13
  - PIO timeout settings (table), 1-12
  - power management control pin settings (table), 1-12
  - power-management setup, 1-10–1-11
    - table, 1-11
  - SMI-generation settings for PIO accesses (table), 1-13
- reading PMU mode, 1-9

powered-down device SMIs

- description, 1-29–1-31
- SMI device-powerdown flowchart (figure), 1-30

PPBIENB bit, 5-77–5-78

PPISBI bit, 5-77

PPOEN signal, 5-78

PRIM\_RI bit, 5-17

printer I/O port select logic (table), 3-34

PRNPRT1-PRNPRT0 bits, 3-34

Programmable General-Purpose Pins 2 and 3, 1-15.  
*See also* PGP pins, controlling.

Programmable Interval Timer Registers

- description, 4-3
- System Timer Registers (table), 4-3

programmable logic for PCMCIA controller, 2-21

PROTDM bit, 3-34

pseudo-Suspend mode, 1-9

## Q

quiet bus, 1-41

## R

RA0 bit, 5-28

RA1 bit, 5-28

RAM mode decode logic (table), 5-79

$\overline{RAS0}$  signal, 2-3

$\overline{RAS1}$  signal, 2-3

RDOSCMDL bit

- function, 5-30
- ROM DOS command delay select logic (table), 5-31

RDOSSIZ3-RDOSSIZ0 bits

- functions, 5-84
- ROM DOS linear address decode size select logic (table), 5-85

RDOSWS0 bit

- function, 5-30
- ROM DOS wait state select logic (table), 5-31

RDOSWS1 bit

- function, 5-33
- ROM DOS wait state select logic (table), 5-33

RDOSWSEN bit

- function, 5-30
- ROM DOS command delay select logic (table), 5-31
- ROM DOS wait state select logic (table), 5-31

RDY bit, 5-68

RDYB bit, 5-19

read only memory. *See* ROM BIOS memory.

reading PMU mode, 1-9

real-time clock

- addressing, 4-14
- overview, 4-14
- Register A (RTC Index 0Ah)
  - bit descriptions, 4-15

- periodic-interrupt rate-selection bits (table), 4-16
- time-base divider-chain bits (table), 4-16
- Register B (RTC Index 0Bh), 4-17
- Register C (RTC Index 0Ch), 4-17
- Register D (RTC Index 0Dh), 4-17
- RTC register summary (table), 4-15
- Receiver Buffer Register (Ports 2F8h & 3F8h), 4-9
- refresh states, 2-4–2-5
  - 8254-based DRAM refresh, 2-4
  - refresh cycle wait states (table), 5-33
  - refresh initialization example (table), 2-4
  - refresh interval select logic (table), 5-39
  - self-refresh DRAMs, 2-26
  - slow refresh, 1-41, 2-4
- REFSEL bit, 5-71
- REFSEL0 bit
  - function, 5-39
  - refresh interval select logic (table), 5-39
- REFSEL1 bit
  - function, 5-39
  - refresh interval select logic (table), 5-39
- REFWS bit, 5-33
- REG signal
  - generated by control logic, 2-20
  - programmable logic for PCMCIA controller, 2-21
- REG\_A signal
  - controlling, 5-59
  - generation by PCMCIA I/O Card IRQ Redirection Control A Register (Index 06h), 5-14–5-15
- Register A (RTC Index 0Ah)
  - bit descriptions, 4-15
  - periodic-interrupt rate-selection bits (table), 4-16
  - time-base divider-chain bits (table), 4-16
- Register B (RTC Index 0Bh), 4-17
- Register C (RTC Index 0Ch), 4-17
- Register D (RTC Index 0Dh), 4-17
- RESCPU bit, 5-47
- Reserved Register (Index 0Bh), 5-18
- Reserved Register (Index 0Fh), 5-21
- Reserved Register (Index 6Ah), 5-43
- Reserved Register (Index 8Bh), 5-59
- Reserved Register (Index 8Eh), 5-60
- Reserved Register (Index 9Dh), 5-66
- Reserved Register (Index 90h), 5-61
- Reserved Register (Index 93h), 5-63
- Reserved Register (Index AEh), 5-76
- Reserved Registers (Indexes 52h-5Fh), 5-32
- Reserved Registers (Indexes 78-7Fh), 5-55
- Reserved Registers (Indexes 96-99h), 5-64
- Reserved Registers (Indexes 28-39h), 5-24
- Reserved Registers (Indexes 49-4Fh), 5-30
- Reserved Registers (Video Indexes 10-11h), 3-27
- RESIN pin, 1-40
- RESU bit, 5-56
- RESUME bit, 5-70
- Resume Mask Register (Index 08h)
  - bit descriptions, 5-16
  - non-CPU activity selection, 1-19
- Resume Status Register (Index 09h)
  - activity monitoring, 1-19
  - bit descriptions, 5-17
  - false wake-ups, 1-24–1-25
- RFD bit, 4-18
- RI bit, 4-13
- RI signal (table), 1-23
- RIMSK bit, 5-16
- ring-in wake-ups, 1-25
- Ring-Indicate signals, 1-23
- RLSD bit, 4-13
- RMSI bit, 5-38
- ROM BIOS memory, 2-7–2-9
  - address initialization (table), 2-8
  - copying ROM contents to DRAM, 2-8–2-9
    - illustration, 2-9
  - enable and wait-state select logic (table), 5-81
  - high memory (figure), 2-8
  - registers for mapping and controlling (table), 5-11
  - wait states (table), 5-33
- ROM chip-select command gating, 2-23
- ROM chip-select signal
  - high-speed clock ROM cycles, 2-25
  - purpose and use, 2-25
  - ROMCS wait-state control-bit logic (table), 2-25
- ROM Configuration 1 Register (Index 65h)
  - bit descriptions, 5-40
  - caution when performing read-modify-write sequence (note), 2-9
  - setting up Page-mode DRAM accesses, 2-5
- ROM Configuration 2 Register (Index 51h)
  - bit descriptions, 5-32
  - controlling ROM-DOS address space, 2-9

- ROM Configuration 3 Register (Index B8h)
    - accessing ROM-DOS address space, 2-10
    - bit descriptions, 5-84
    - controlling number of wait states, 2-25
    - enabling high-speed clock rate, 2-25
    - ROM DOS enable and wait-state logic (table), 5-84
    - ROM DOS linear address decode size select logic (table), 5-85
  - ROM DOS memory, 2-9–2-10
    - command delay select logic (table), 5-31
    - description, 2-9–2-10
    - enable and wait-state logic (table), 5-84
    - linear address decode size select logic (table), 5-85
    - registers for mapping and controlling (table), 5-11
    - wait state select logic (table), 5-31
  - ROM shadowing, 2-8
  - ROMCS signal
    - enabling and disabling, 2-7–2-9, 5-81
    - high-speed clock ROM cycles, 2-25
    - ROM chip-select command gating, 2-23
    - ROM chip-select signal, 2-25
    - wait-state control-bit logic (table), 2-25
  - ROMDOS16 bit, 5-32
  - ROMWS0 bit, 5-32
  - ROMWS1 bit, 5-32
  - RS3-RS0 bits, 4-15
  - RTC Index Data Register (Port 071h), 4-19
  - RTC registers. *See* real-time clock.
  - RTCSMIEN bit
    - SMI Enable Register (Index 41h), 5-25
    - SMI I/O Status Register (Index 42h), 5-26
  - RTS bit, 4-12
- S
- SACIN bit, 5-48
  - SB bit, 4-11
  - SBBLUE bit, 3-25
  - SBGREEN bit, 3-25
  - SBHE/LCDDL3 signal (table), 3-2
  - SBINT bit, 3-25
  - SBRED bit, 3-25
  - SC2-SC0 bits, 5-64
  - SC03 bit, 5-42
  - SC8B bit, 5-42
  - SC47 bit, 5-42
  - SCCF bit, 5-42
  - SCKTA16I bit, 5-17
  - SCKTA16M bit, 5-17
  - SCKTB16I bit, 5-17
  - SCKTB16M bit, 5-17
  - Scratch Pad Register (Ports 2FFh & 3FFh), 4-13
  - Screen Adjust Lower Byte Register (Video Index 1Ah)
    - description, 3-32–3-34
    - page boundaries in 80x25 text mode (table), 3-33
  - Screen Adjust Upper Byte Register (Video Index 1Bh)
    - description, 3-32–3-34
    - page boundaries in 80x25 text mode (table), 3-33
  - screen-background color-bit logic (table), 3-25
  - screen blanking. *See* auto screen blanking (ASB).
  - Screen Control 2 Register (Video Index 19h)
    - auto screen blanking, 3-32
    - timer setting (table), 3-32
    - bit descriptions, 3-31
    - display controller enable, 3-32
  - Screen Control Restore Register (Video Index 18h), 3-30–3-31
  - SD03 bit, 5-42
  - SD8B bit, 5-42
  - SD47 bit, 5-42
  - SDCF bit, 5-42
  - SE03 bit, 5-43
  - SE8B bit, 5-43
  - SE47 bit, 5-43
  - SECF bit, 5-43
  - SET bit, 4-17
  - SF03 bit, 5-43
  - SF8B bit, 5-43
  - SF47 bit, 5-43
  - SFCF bit, 5-43
  - SHADOW bit, 5-40
  - Shadow RAM Enable 1 Register (Index 68h), 5-42
  - Shadow RAM Enable 2 Register (Index 69h), 5-43
  - SHUTD bit, 5-37
  - signal interfaces for video controller, 3-1–3-3
  - SLCT bit
    - Parallel Status Port (Ports 279h & 379h), 4-8
    - Parallel Status Port (Ports 279h, 379h & 3BDh), 4-7

- SLCTIN bit, 4-7
- Sleep mode
  - programming PMU to enter at Battery Level 2, 1-33
  - purpose and use, 1-7
- Sleep to Suspend Timer Register (Index 86h)
  - description, 5-57
  - initializing for suspend/resume operation, 1-37
  - state-transition timing, 1-22
- slow refresh, 1-41, 2-4
- SLP bit, 5-56
- SLREF bit, 5-71
- SMI and NMI control, 1-25–1-31. *See also* NMIs; SMIs.
  - accesses to powered-down device SMI, 1-29–1-31
  - device-powerdown flowchart (figure), 1-30
  - enabling SMIs, 1-27
    - registers for (table), 1-27
  - external SMI
    - with multiple devices, 1-31
    - with single device, 1-31
  - external SMI pin, 1-31
  - processing NMI or SMI source, 1-28–1-29
  - SMI processing flowchart (figure), 1-26
  - sources for generating SMIs, 1-25
  - Temporary-On mode, 1-26–1-27
  - treatment of pending SMIs, 1-31
- SMI Enable Register (Index 41h)
  - bit descriptions, 5-25
  - powered-down device SMIs, 1-30, 1-31
  - using with I/O Timeout Register (Index 40h), 5-24
- SMI I/O Status Register (Index 42h)
  - bit descriptions, 5-26
  - powered-down device SMIs, 1-30
- SMI MMS Page Register (Index AAh)
  - bit descriptions, 5-73
  - enabling SMIs, 1-27
- SMI Status Register (Index 43h)
  - bit descriptions, 5-26
  - determining source of SMIs, 1-28
- SMIA14 bit, 5-73
- SMIA15 bit, 5-73
- SMIA16 bit, 5-73
- SMIA17 bit, 5-73
- SMIA18 bit, 5-73
- SMIA19 bit, 5-73
- SMIA20 bit, 5-73
- SMIA21 bit, 5-73
- SMIA22 bit, 5-73
- SMIA23 bit, 5-73
- SMIs
  - generation by BL2# and  $\overline{BL3}$ , 1-33
  - overview, 1-1, 1-2
  - registers for controlling and determining status (table), 5-11
  - resume input causing SMI, 1-39
  - SMI-generation settings for PIO access example (table), 1-13
  - suspend input causing SMI, 1-37
  - suspend/resume operation, 1-35
    - start of SMI handler, 1-37
  - timer-controlled shutdown using SMI interface, 1-14–1-15
- SMM RAM state-save area
  - resume pseudocode, 1-39
  - suspend pseudocode, 1-38
- Software Mode Control Register (Index 88h)
  - avoiding incoherency in SMI or NMI state-transitions, 1-29
  - bit descriptions, 5-58
  - PMU mode select logic (table), 5-58
- SOUTL bit, 5-41
- SP bit
  - Line Control Register (Ports 2FBh & 3FBh), 4-11
  - NMI/SMI Control Register (Index A5h), 5-70
- SP0 bit, 5-75
- SP1 bit, 5-75
- SP2 bit, 5-56
- SP3 bit, 5-74
- SP4 bit, 5-74
- SPC0 bit, 5-58
- SPC1 bit, 5-58
- SPC2 bit, 5-58
- SPCHRFN bit, 3-34
- SPEED bit, 5-34
- SPKD bit, 4-18
- SQWE bit, 4-17
- SRAM
  - choosing between DRAM and SRAM, 2-2
  - configurations, 2-3–2-4
  - display memory for video controller, 3-3–3-4
    - SRAM address mapping during CPU access (table), 3-4
  - memory configuration (DRAM and SRAM) (table), 5-41
  - memory initialization example (table), 2-3

- registers for setting up (table), 5-7
- SRAM option configuration (table), 2-4
- SRAM wait state select logic (table), 5-38
- video controller font fetches to SRAM
  - font areas (table), 3-14
  - SRAM address mapping during font fetches (table), 3-15
  - storing fonts in video SRAM, 3-15–3-16
- wait states, 2-5
- SRAM display data area partitioning, 3-11–3-13
  - 480x320 LCD panel, 60x40 characters (table), 3-13
  - 640x200 display, 40x25 characters (table), 3-12
  - location of pages in 40x25 text mode, 3-13
  - location of pages in 80x25 text mode, 3-12
- SRAMCS bit, 3-37
- SSMSIZE bit, 5-37
- standard video registers. *See* video index registers, standard.
- Start Address Registers (Video Indexes 0C-0Dh), 3-27
- state machine. *See* PMU state machine.
- State-Transition Timer, 1-22–1-23
  - definition, 1-2
  - events causing reset, 1-23
  - programming with Mode Timer registers, 1-22
  - registers for controlling (table), 5-11
- state-transitions
  - avoiding incoherency in SMI or NMI state-transitions, 1-28–1-29
  - flowchart (figure), 1-20–1-21
- status logic for PCMCIA controller, 2-21
- stepping level, determining. *See* Version Register (Index 64h).
- STOPDIS bit, 3-37
- STP bit, 4-11
- STROBE bit, 4-7
- STSAIRQSEL bit, 5-20–5-21
- SU bit, 5-70
- SU0 bit, 5-75
- SU1 bit, 5-75
- SU2 bit, 5-56
- SU3 bit, 5-74
- SU4 bit, 5-74
- SUS bit, 5-56
- SUS/RES pin. *See also* suspend/resume operation.
  - overview, 1-3
  - PMU operating-mode transitions (figure), 1-4
  - resume input, 1-35
  - suspend input, 1-35
  - wake-up events, 1-25
- Suspend mode
  - automatic entry upon Battery Level 4, 1-33–1-34
  - pseudo-Suspend mode, 1-9
  - purpose and use, 1-7
- suspend/resume operation, 1-34–1-39
  - avoiding problems, 1-39
  - programming considerations, 1-36
  - purpose and use, 1-34
  - required configuration-register initializations, 1-37
  - resume input causing SMI, 1-39
  - resume pseudocode, 1-39
  - start of SMI handler, 1-37
  - suspend input causing SMI, 1-37
  - suspend pseudocode, 1-38
  - using SMIs, 1-35
- Suspend to Off Timer Register (Index 87h)
  - description, 5-58
  - state-transition timing, 1-22
- SYSCLK signal
  - disabled in Sleep mode, 1-7
  - XT-keyboard clock (XTCLK), B-2
- SYSCLK speeds (table), 1-6
- System Management Interrupt (SMI). *See* SMIs.
- system memory, 2-2–2-7. *See also* DRAM; SRAM.
  - bus configurations, 2-3
  - choosing between DRAM and SRAM, 2-2
  - DRAM configurations, 2-3
  - ISA, local bus, and internal LCD configurations, 2-6–2-7
  - overview, 2-2–2-3
  - refresh and wait states, 2-4–2-6
  - SRAM configurations, 2-3–2-4
  - typical AT address space (figure), 2-2
- System Timer Registers (table), 4-3

## T

- T1 bit, 5-66
- T2 bit, 5-66
- T2G bit, 4-18
- T2OUT bit, 4-18
- technical support, iii
- Temporary-On mode
  - purpose and use, 1-26–1-27
  - suspend/resume operation, 1-36
- TEMT bit, 4-12

- TERI bit, 4-13
  - text mode, 3-7–3-13
    - attribute byte, 3-9
    - CGA attribute byte, 3-9–3-10
      - background-color bit values (table), 3-10
      - foreground-color bit values (table), 3-10
    - CGA text mode
      - character fonts 1 area (table), 3-7
      - character fonts 2 area (table), 3-8
      - special character fonts area (table), 3-8
    - character byte, 3-9
    - description, 3-7–3-8
    - HGA attribute byte, 3-11
      - bit values (table), 3-11
    - HGA text mode, 3-8
    - SRAM display data area partitioning, 3-11–3-13
      - 480x320 LCD panel, 60x40 characters (table), 3-13
      - 640x200 display, 40x25 characters (table), 3-12
      - location of pages in 40x25 text mode, 3-13
      - location of pages in 80x25 text mode, 3-12
  - Text Truncation Start Register (Video Index 21h), 3-35
  - Text Truncation Stop Register (Video Index 22h)
    - description, 3-35–3-36
    - truncation-register programming example (table), 3-36
  - TGCON bit, 3-24
  - THRE bit, 4-12
  - time-base divider-chain bits (table), 4-16
  - timer-controlled shutdown using SMI interface, 1-14–1-15
  - timer registers
    - Doze to Sleep Timer Register (Index 85h), 1-22, 5-57
    - High-Speed to Low-Speed Timer Register (Index 83h), 1-22, 5-57
    - Low-Speed to Doze Timer Register (Index 84h), 1-22, 5-57
    - Sleep to Suspend Timer Register (Index 86h), 1-22, 1-37, 5-57
    - Suspend to Off Timer Register (Index 87h), 1-22, 5-58
  - Transmitter Holding Register (Ports 2F8h & 3F8h), 4-9
  - trigger period select logic (table), 5-66
  - TRUNC bit, 3-34
  - truncation registers
    - Graphics Truncation Start Register (Video Index 23h), 3-36
    - Graphics Truncation Stop Register (Video Index 24h), 3-36–3-37
    - Text Truncation Start Register (Video Index 21h), 3-35
  - Text Truncation Stop Register (Video Index 22h), 3-35–3-36
  - truncation-register programming example (table), 3-36
  - TXTBLNK bit
    - CGA Mode Control Register (Port 3D8h), 3-24
    - HGA Mode Control Register (Port 3B8h), 3-22
  - TXTCON bit, 3-24
  - TXTGRP bit, 3-22
- ## U
- UART registers
    - converting Port 3FDh from read only to read/write, 5-30
    - Divisor Latch Lower Byte (Ports 2F8h & 3F8h), 4-10
    - Divisor Latch Upper Byte (Ports 2F8h & 3F9h), 4-10
    - Interrupt Enable Register (Ports 2F9h & 3F9h), 4-10
    - Interrupt Identification Register (Ports 2FAh & 3FAh), 4-10–4-11
    - Line Control Register (Ports 2FBh & 3FBh), 4-11
    - Line Status Register (Ports 2FDh & 3FDh), 4-12
    - Modem Control Register (Ports 2FCh & 3FCh), 4-12
    - Modem Status Register (Ports 2FEh & 3FEh), 4-13
    - Receiver Buffer Register (Ports 2F8h & 3F8h), 4-9
    - registers for setting up UART (table), 5-7
    - Scratch Pad Register (Ports 2FFh & 3FFh), 4-13
    - Transmitter Holding Register (Ports 2F8h & 3F8h), 4-9
    - UART Clock Enable Register (Index 92h), 5-63
  - UART\_EN bit, 5-55
  - UART\_IOP bit, 5-55
  - UART\_IR3 bit, 5-55
  - UART\_IR4 bit, 5-55
  - UF bit, 4-17
  - UIE bit, 4-17
  - UIP bit, 4-15
- ## V
- VD bit
    - Activity Mask 2 Register (Index 76h), 5-54
    - Activity Status 2 Register (Index A1h), 5-67
  - Version Register (Index 64h)
    - bit descriptions, 5-38
    - read functions (table), 5-38
    - read version stepping level decode (table), 5-38
    - refresh interval select logic (table), 5-39
    - write functions (table), 5-39

- Vertical Registers (Video Indexes 04-07h), 3-27
- VERTRET bit
  - CGA Status Register (Port 3DAh), 3-22
  - HGA Status Register (Port 3BAh), 3-22
  - LCD Special Register (Index 25h), 3-37
- VIDCON bit
  - CGA Mode Control Register (Port 3D8h), 3-24
  - HGA Mode Control Register (Port 3B8h), 3-22
- video clock speeds (table), 1-6
- video controller
  - 480x320 LCD panel operation in 1-bpp mode, 3-7
  - disabling internal video controller, 3-32
  - display interface
    - display-data bits, 3-2
    - signals (table), 3-2
  - display memory
    - description, 3-3
    - SRAM address mapping during CPU access (table), 3-4
  - fonts, 3-13–3-19
    - font areas, 3-13–3-14
    - font example 1: the letter A, 3-16–3-17
    - font example 2: the letter M, 3-18–3-19
    - storing fonts in video SRAM, 3-15–3-16
    - video controller font fetches to SRAM, 3-14–3-15
      - CA7-CA0, 3-15
      - DISPMOD, 3-15
      - font areas (table), 3-14
      - RA7-RA0, 3-15
      - SPCHRFNT, 3-15
      - SRAM address mapping during font fetches (table), 3-15
  - graphics mode, 3-4–3-7
    - 640x200 and 320x200 video-memory data (table), 3-5
    - 640x200 video-memory data for 480x320 LCD panel (table), 3-6
    - description, 3-4
    - memory requirements
      - 480x320 LCD panel, 3-5
      - 640x200 LCD panel, 3-5
  - memory interface signals, 3-1–3-2
  - miscellaneous signals interface
    - description, 3-3
    - signals (table), 3-3
  - overview, 3-1
  - signal interfaces, 3-1–3-3
  - text mode, 3-7–3-13
    - attribute byte, 3-9
    - CGA attribute byte
      - background-color bit values (table), 3-10
      - bit descriptions, 3-9–3-10
      - foreground-color bit values (table), 3-10
    - CGA text mode
      - character fonts 1 area (table), 3-7
      - character fonts 2 area (table), 3-8
      - special character fonts area (table), 3-8
    - character byte, 3-9
    - description, 3-7–3-8
    - HGA attribute byte
      - bit descriptions, 3-11
      - bit values (table), 3-11
    - HGA text mode, 3-8
    - SRAM display data area partitioning, 3-11–3-13
      - 480x320 LCD panel, 60x40 characters (table), 3-13
      - 640x200 display, 40x25 characters (table), 3-12
      - location of pages in 40x25 text mode (table), 3-13
- video index registers, extended
  - Color Mapping Registers (Video Index 14-17h & 1C-1Fh), 3-29–3-30
  - Control 1 Register (Video Index 20h), 3-34
  - Disable Software Switch Register (Video Index 13h), 3-28
  - Enable Software Switch Register (Video Index 12h), 3-28
  - Graphics Truncation Start Register (Video Index 23h), 3-36
  - Graphics Truncation Stop Register (Video Index 24h), 3-36–3-37
  - LCD Special Register (Video Index 25h), 3-37
  - light-pen functionality unsupported (note), 3-20
  - list of registers (table), 3-21
  - nonstandard, superset of CGA/HGA (note), 3-20
  - programming examples, 3-37–3-42
    - 320x240 LCD panel configuration (table)
      - CGA (Video Index 18h, Bit 0 = 0), 3-40
      - HGA (Video Index 18h, Bit 0 = 1), 3-41
    - 480x320 LCD panel configuration (table), 3-39
    - 640x200 LCD panel configuration (table), 3-38
    - 720x348 LCD panel configuration (table), 3-42
  - Screen Adjust Lower Byte Register (Video Index 1Ah), 3-32–3-34
  - Screen Adjust Upper Byte Register (Video Index 1Ah), 3-32–3-34
  - Screen Control 2 Register (Video Index 19h), 3-31–3-32
  - Screen Control Restore Register (Video Index 18h), 3-30–3-31
  - Text Truncation Start Register (Video Index 21h), 3-35
  - Text Truncation Stop Register (Video Index 22h), 3-35–3-36
- video index registers, standard
  - Cursor Address Registers (Video Indexes 0E-0Fh), 3-27

- Cursor Start and End Registers (Video Indexes 0A-0Bh), 3-27
- Horizontal Registers (Video Indexes 00-03h), 3-26–3-27
- Interlace Mode Register (Video Index 08h), 3-27
- light-pen functionality unsupported (note), 3-20
- list of registers (table), 3-21
- Max Scan Line Register (Video Index 09h), 3-14, 3-27
- Reserved Registers (Video Indexes 10-11h), 3-27
- Start Address Registers (Video Indexes 0C-0Dh), 3-27
- Vertical Registers (Video Indexes 04-07h), 3-27
- Video PLL, 1-8
- video port registers
  - CGA Color Select register (Port 3D9h), 3-25
  - CGA Index Address register (Port 3D4h), 3-23
  - CGA Index Data register (Port 3D5h), 3-23
  - CGA Mode Control register (Port 3D8h), 3-24
  - CGA Status register (Port 3DAh), 3-26
  - HGA Configuration register (Port 3BFh), 3-23
  - HGA Index Address register (Port 3B4h), 3-21
  - HGA Index Data register (Port 3B5h), 3-22
  - HGA Mode Control register (Port 3B8h), 3-22
  - HGA Status register (Port 3BAh), 3-22–3-23
  - light-pen functionality unsupported (note), 3-20
  - video port definitions (table), 3-20
- VPP pins
  - programmable logic for PCMCIA controller, 2-21
  - VPPA pin, 5-15
  - VPPB pin, 5-24
- VRT bit, 4-17
- Wait State Control Register (Index 63h)
  - bit descriptions, 5-37
  - DRAM bank miss wait state select logic (table), 5-37
  - DRAM first cycle wait state select logic (table), 5-37
  - setting up Page-mode DRAM accesses, 2-5
  - SRAM wait state select logic (table), 5-38
  - SRAM wait states, 2-5
- wait states and command delays
  - 33-MHz wait states (table), 2-6
  - command delay duration for various cycles (table), 2-24
  - Enhanced Page mode, 2-5
  - memory-speed initialization example (table), 2-5
  - Page-mode DRAMs, 2-5
  - ROM-BIOS, ROM-DOS, PCMCIA, and ISA cycles, 2-23
  - wait states for various cycles (table), 2-24
- wake-up logic, 1-23–1-25
  - definition, 1-2
  - events allowed, 1-23–1-24
  - false wake-ups, 1-24–1-25
  - IRQ or DRQ pins, 1-24
  - ring-in wake-ups, 1-25
  - Temporary-ON SMI or NMI routine requirements, 1-24
  - wake-up signal descriptions (table), 1-23
- WLB0 bit, 4-11
- WLB1 bit, 4-11
- WP bit, 5-68
- WPB bit, 5-19

## W

- wait state registers
  - I/O Wait State Register (Index 61h)
    - bit descriptions, 5-34
    - floppy disk drive wait states (table), 5-34
    - general bus I/O wait states (table), 5-35
    - hard disk drive wait states (table), 5-35
  - MMS Memory Wait State 1 Register (Index 62h)
    - 8-bit ISA memory-cycle wait states (table), 5-36
    - 16-bit ISA memory-cycle wait states (table), 5-36
    - bit descriptions, 5-35
    - setting up Page-mode DRAM accesses, 2-5
  - MMS Memory Wait State 2 Register (Index 50h)
    - bit descriptions, 5-30–5-31
    - PCMCIA command delay select logic (table), 5-32
    - PCMCIA wait state select logic (table), 5-31
    - ROM DOS command delay select logic (table), 5-31
    - ROM DOS wait state select logic (table), 5-31

## X

- X1SEL bit, 5-77
- XDMSTAT bit, 3-22
- XT Keyboard Data Register (Port 060h), 4-18
- XT-keyboard interface
  - block diagram, B-1
  - controlling, B-2
  - enabled by bit 3 of Index 0ADh, B-2
  - I/O map summary, B-2
  - keyboard data port, B-2
  - overview, B-1
  - pins used, B-2
  - timing, B-3
- XTALUSE bit, 5-78
- XTCLK, B-2
- XTKBDEN bit, 5-76